कृषि ज्ञान प्रबंधन के लिए वर्गीकरण पाठ से ऑन्टोलॉजी लर्निंग

Ontology Learning from Taxonomic Text for Agricultural Knowledge Management

CHANDAN KUMAR DEB

ICAR-INDIAN AGRICULTURAL STATISTICS RESEARCH INSTITUTE

ICAR-INDIAN AGRICULTURAL RESEARCH INSTITUTE

NEW DELHI – 110012

2020

Ontology Learning from Taxonomic Text for Agricultural Knowledge Management

BY

CHANDAN KUMAR DEB

A Thesis Submitted to the Faculty of Post-Graduate School,
ICAR-Indian Agricultural Research Institute, New Delhi
In partial fulfillment of the requirements
For the degree of

DOCTOR OF PHILOSOPHY
IN
COMPUTER APPLICATION
2020

Approved by:

Chairperson: ---------- 3/11/2020

(Dr. Sudeep Marwaha)

Co- Chairperson ----------

(Dr. Rajni Jain)

Members

(Dr. Alka Arora)

(Dr. A. R. Rao)

(Dr. Monendra Grover)

फैक्स/FAX : 011-25841564
दूरभाष/Phones: संस्थान/Off: 011-25841074
Mobile: 9711707437
ईमेल/Email : sudeep.marwaha@icar.gov.in

भारतीय कृषि सांख्यिकी अनुसंधान संस्थान
(भा.कृ.अ.प.)
लाइब्रेरी एवेन्यु, पुसा, नई दिल्ली - 110012 (भारत)
Indian Agricultural Statistics Research Institute
(ICAR)
Library Avenue, Pusa, New Delhi-110012 (India)

Dr. Sudeep Marwaha
Professor & Head (A),
Division of Computer Applications

CERTIFICATE

This is to certify that the thesis entitled "***Ontology Learning from Taxonomic Text for Agricultural Knowledge Management***" submitted to the Faculty of the Post-Graduate School, Indian Agricultural Research Institute, New Delhi, in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Applications**, embodies the results of bona fide research work carried out by **Mr. Chandan Kumar Deb** under my guidance and supervision, and that no part of this thesis has been submitted for any other degree or diploma.

It is further certified that any assistance and help availed during the course of investigation as well as source of information have been duly acknowledged by him.

(Dr. Sudeep Marwaha)
Chairperson
Advisory committee

Place: New Delhi

Date: 3/1/2020

ACKNOWLEDGEMENTS

I owe my deepest sense of gratitude and unforgettable indebtness to Dr. Sudeep Marwaha, Principal Scientist, ICAR-IASRI and chairperson of my advisory committee for his constant encouragement, valuable suggestions, affectionate behaviour, useful discussions and constructive criticisms during the course of investigation and preparation of the manuscript. His caring behaviour was not only limited to being a chairperson but also gave all kind of moral support to me. I find no words to express my heart-felt obligation to him.

I express my sincere thanks to Dr. Rajni Jain, Principal Scientist, ICAR-NIAP, New Delhi and co-chairman of my advisory committee, for her encouragement, valuable and generous guidance during the entire course of work.

I extend my sincere thanks to Dr. Alka Arora, Principal Scientist, Division of Computer Applications and member of my advisory committee for her wise counsel and help during this investigation.

I feel immense pleasure to thank Dr. A. R. Rao, Principal Scientist, Division of Bioinformatics for his valuable suggestions and help throughout this endevour.

I implore my feeling of profound gratitude to Dr. M. Grover, Principal Scientist, Division of

Bioinformatics, for his cordial attitude and generous help during the course of this investigation.

I avail myself of this opportunity to convey my heartfelt thanks to Director, ICAR-IASRI and Dr. Sudeep Marwaha, Professor, Computer Applications for providing necessary facilities and valuable help and encouragement during the entire span of my work.

I also express my sincere thanks to my teachers, seniors, juniors and friends for their priceless co-operation specially Pradip and Murari.

I owe a lot to the staff of CAS lab and TAC of IASRI for their help and co-operation.

No word of mine would be adequate to express my indebtedness to my life partner Madhurima for her painstaking and untiring efforts to make me what I am today. My vocabulary utterly fails in expressing my love to her for her affection and unstinted support during all these years of study.

I extend my thanks to the Dean and Director, ICAR-IARI, New Delhi and staff of PG School for their helpful attitude and cooperation throughout the period of study. I wish to thank ICAR-ICAR-IASRI and DST-INSPIRE for providing financial assistance in the form of fellowship during the tenure of my study.

Above all, I wish to bend my head to the Almighty who blessed me to retain good health and peace of mind.

DATE: 03/01/2020

PLACE: New Delhi

Chandan Kumar Deb

(CHANDAN KUMAR DEB)

CONTENTS

CHAPTER I

INTRODUCTION

The present era is the era of Artificial Intelligence (AI). Every aspect of life feels the presence of AI. Knowledge representation is one of the prominent fields of AI. Like other fields of AI, contemporary knowledge representation techniques are far sophisticated than the previous knowledge representation techniques. But the souls of the today's knowledge representation techniques are present in the base of old knowledge representation techniques. The representation of natural knowledge into machine understandable form has a step by step history of progression. Initially, the knowledge was represented in simple statements with the associated meaning of TRUE and FALSE by using Propositional Logic (Lewis and Leibniz, 1918; De Morgan A., 1847; Boole G., 1848). Next, the Predicate Logic (Kowalski, 1974) was used for long time to represent complex statements. Instead of the Propositional Logic the Predicate Logic gave some more expressiveness to the knowledge representation.

After the era of Propositional and Predicate Logic the graph based representation was used for a long period. The Semantic Net (Richens, R. H., 1956) is a famous example of graph based knowledge representation technique. Semantic Net represents the knowledge with the help of nodes and links. The nodes usually describe the objects and labelled links describes the relationship among the objects. To represent the real world knowledge, Semantic Net can be used but links has no standard definition. Semantic Net lacks intelligence. The intelligence or the expressiveness of the network depends on the creator of the Semantic Net. Frame (Hayes, 1981) Slot and scripts techniques represent the knowledge in natural groupings of their existence and distribute it in hierarchy. All these representation techniques have their own advantages and limitations. These techniques are not suitable for internet based applications. They lack standardization and universal acceptability.

The above described knowledge representation techniques are primitive in nature, also both the expressiveness and reasoning ability has some limitation compared to latest XML based knowledge representation technique. XML based knowledge representation techniques are RDF (Klyne and Caroll, 2003; Brickley and Guha, 1999, 2004), RDFS (Brickley and Guha, 2000; Baader *et al.,* 2002; Gennari *et al.,* 2003), Topic MAP, OIL

(Ontology Inference Layer) and OWL (Web Ontology Language) (Decker *et al*., 2000; Smith *et al.,* 2003, 2004).

Reliable Taxonomy is the basis for any meaningful research in biology. Taxonomy is important for knowledge on Biodiversity. Taxonomic knowledge can also deal with an invasive alien species. Lyas e*t al.,* 2007 showed that how Taxonomic information's are important for agriculture as well as for Biodiversity. Additionally they proposed that the taxonomy based tools of Invasive Alien Species detection, identification, and research on ecological interactions between the pest, host, and ecosystem are all indispensable in planning defensive strategies and integrated control measures. Smith *et al.,* 2011 showed approximately fifty case studies of a long period of time as to how taxonomic knowledge help agriculturist to solve the real world challenges. Unfortunately, systematic knowledge of taxonomy is much unstructured; most of them are available in the form of books. So, information retrieval is also very difficult from this unstructured knowledge. To make the unstructured knowledge into structured one, there are many knowledge organization techniques. Ontology is a very powerful tool for knowledge representation. The Taxonomic Knowledge has a great correspondence to the ontology. Bedi and Marwaha, 2004 proposed a methodology for the conversion of Taxonomies into Ontologies. But manual ontology building is a tremendous labour intensive task. For the motivational purpose, we can refer two manually build ontology e.g. Soil Taxonomy Ontology (Das *et al.,* 2012; Deb *et al.,* 2015) and Microbial Ontology (Biswas *et al.,* 2013) developed at IASRI. These two ontologies have taken large no of person months to build. Although we have unstructured data, we can make it structured, but through a very lengthy process so, the automated ontology learning approach is developed to face this knowledge acquisition bottleneck. But this approach has some serious limitation in text understanding, knowledge extraction, structured labelling and filtering (Zouaq *et al.,* 2011).

1.1 Motivation of the Study

Agriculture domain is one of the biggest sources of knowledge but most of the available knowledge is in unstructured form. Ontology building is a way to convert the unstructured knowledge to a structured one for human as well as machine usage. Multiple authors have attempted to automatic ontology development by learning through plain text. In such a situation the learning is governed by the quality of text provided and the ontology. So developed ontology from plain text is prone to the spurious concepts to concepts extraction and also may place the concepts in the wrong hierarchy in the ontology.

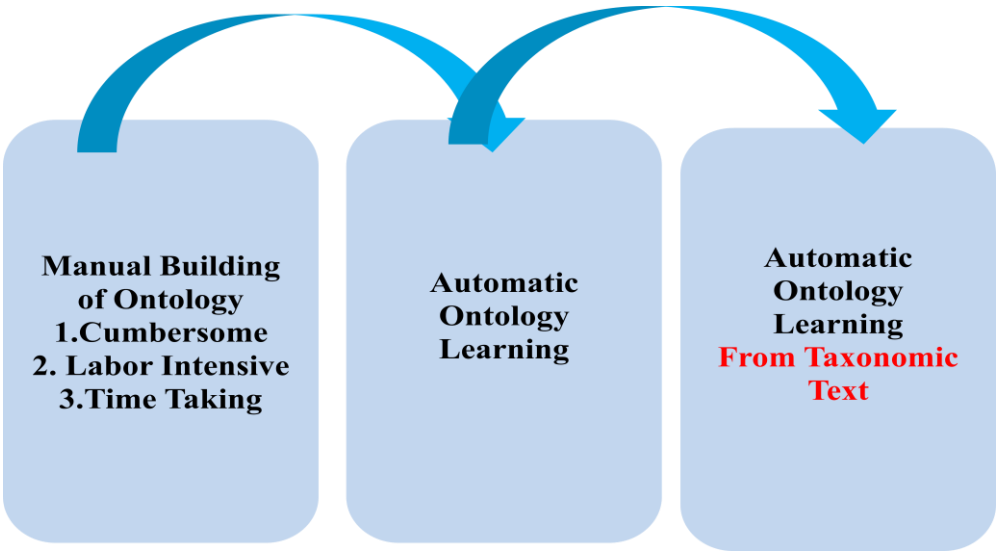


Figure 1.1: Problem definition of the research

On the other hand, the taxonomic text in a domain is a standardized text and the all the concepts of the domain are explicitly defined. The presented research work targets to developed ontology from taxonomic text. In agriculture this kind of ontology learning is not yet attempted. Keeping in view above, the present research work entitled **"Ontology Learning from Taxonomic Text for Agricultural Knowledge Management**" is proposed with the following objectives.

Objectives:

The objectives of the study are-

1. To study the ontology learning algorithms.
2. To develop the ontology learning algorithms for taxonomic text.
3. To validate the developed algorithms in agricultural domain.

1.2 Scope of the Thesis

The present chapter introduces the significance of the automated ontology learning from taxonomy in the agriculture. The chapter outlines the advantages of the taxonomic text over the normal text. Based on the objectives of the study a detailed review of literature has been done and it is depicted in chapter 2. Chapter 3 discusses the methodology and tool used for the study. Chapter 4 discusses the results obtained from the study. Chapter 5 summarizes the work.

CHAPTER II

BACKGROUND

Existence of a domain Ontology is the primary requirement for building any knowledge based application in any knowledge domain. Nowadays, it is a well-established fact that the Ontology learning is becoming an unavoidable step for facilitating the development of Ontology; as the manual process is not suitable for large scale ontologies. Maedche, A. and Staab, S. (2001) formally introduced Ontology learning in the field of knowledge management. Ontology leaning is based on the study embedded in the previous study of Linguistic, Statistics as well as the Machine learning. The Ontology learning can be subdivided into three major categories. After the introduction of Ontology learning, many attempts have been made to formalize the study of Ontology learning. Buitelaar *et al.,* (2005) first introduced the concept of "**Ontology learning layer cake**". In this chapter we tried to capture some of most prominent works on Ontology learning. Most of the literature suggested that the Ontology leaning methodology can be broadly classified into two categories- firstly, the Linguistic and secondly, the Statistics. Some of the studies are based on the hybrid of the both category. This chapter discusses both the category of Ontology learning methodology. In the last section of this chapter we have discussed some of the prominent Ontology learning systems.

2.1 Linguistic Methodology of Ontology Learning

Linguistic is the study of languages. The linguistic discipline mostly deals with the human language or the natural language. In this study, a scientific study is done to identify the relationship among the entities. In the Ontology learning from the natural text, it is compulsory to study and exploit the characteristics of the natural language.

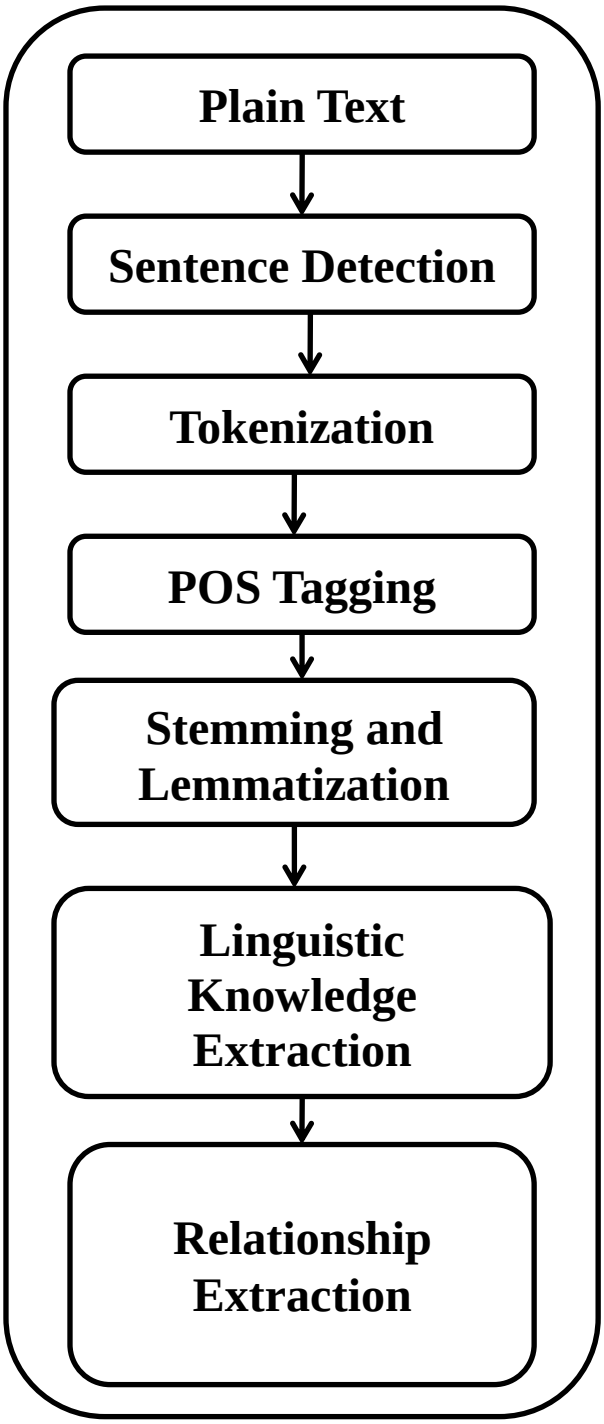


Figure 2.1 Linguistic methodology of Ontology leaning from plain text

2.1.1 Linguistic pre-processing

Linguistic preprocessing is a necessary task before the actual analysis of the text. The sentence detection, tokenization are the initial essential task for any linguistic analysis. After sentence detection and the tokenization, the Parts of Speech Tagging and Lemmatization done for further linguistic analysis.

- **POS Tagging**

Parts of speech tagging is a very important preprocessing task in the Ontology learning process. Following are some of the prominent tools that have been used to develop and make the POS tagging easier.

Brill, E. (1992) developed a rule based system for POS. The tagger shows better result than statistical techniques for identifying the Parts of Speech in the text. The developed tagger is tested on 5% of the Brown corpus.

Schmid, H. (1994) developed a probabilistic model for part-of speech tagging. He used the decision tree to get the reliable estimate of the transition probabilities. He used modified version of ID3 algorithm for building the decision tree and got 96.36% accuracy on Treebank data.

Lin, D. (1994) developed a parser based on the principle grammar called as Principar. It was implemented in C++. It is based on the message passing framework proposed by Dor *et al.*, (1995).

Minipar is a shallow NLP parser. It can identify the dependency relationship of the words in a sentence. It is enabled to find the lemma, POS name of the dependency path.

Oliveira *et al.,* (2001) developed two systems, namely TextStorm and Clouds for concept map induction. They have used augmented grammar for POS tagging and co reference resolution.

Drymonas *et al.,* (2010) used GATE architecture for the POS tagging and other natural language task.

Standford Parser (2012) is natural language parser is capable of parsing the natural language like English. The parser was developed by the Standford University. The package is Java implemented probabilistic natural language parser. The parser is also capable parsing several languages like Chinese, Arabic, French, German and Spanish. Before parsing the parse tag, the parts of speech of the words are needed to be analyzed.

Sleator and Temperley (1995) published a series of paper that describes the Link Grammar Parser. They describe a new form of grammar that is an encoded form of English grammar.

Petit *et al.,* (2017) used Standford core NLP API to support common natural language preprocessing task and it includes tokenization, POS tagging etc.

- **Lemmatization**

Lemmatization is one of the important Natural Language Processing task that help to bring the term into their root form (e.g.‘Running’ and ‘Ran’ should be run).

Petit *et al.*, (2017) developed an Ontology learning system that used the Cornel API of Standford University to lemmatize the textual data.

Drymonas *et al., (2010)* used the JNWL API to lemmatize the text. The JNWL is WordNet java library.

2.1.2 Linguistic Knowledge Extraction

Caraballo, S.A. (2001) developed a method, where nouns are clustered. They have used the hierarchical clustering and have produced a tree with unlabeled internal node. The unlabeled internal node are replaced the hypernym present in the WordNet.

Cederberg, S. and Widdows, D. (2003) demonstrated the mathematical model of hypernym identification by Latent Semantic Analysis (LSA). They got significant improvement in the result for identifying the relationship of the concepts. They have used several supervised classification techniques (e.g. C4.5).

Pantel, P. & Ravichandran, D. (2006) developed a system that is capable of automated labeling of the semantic class. They have used the top downclusteringapproach. The developed system also abled to identify the infrequent words that should be labeled.

Riloff, E. and Shepherd, J. (1999) developed a semi-automated system which identifies the context window of the seed words as provided by the users. It also calculates the score on the basis of the window and the whole corpus and subsequently removes the stop words. Thereafter, it identifies the nouns and after several iteration the system is able to find the new nouns and increase the seed word list size.

Roark, B. and Charniak, E. (1998) developed an algorithm that is capable of extracting the semantic relationship among the corpus which is populated online. The input of the algorithm is parsed corpusand the noun is scored on the basis of co-occurrence statistics. The calculated score helps in the identification of nouns and semantic relationship.

Sombatsrisomboon, R. *et al.*, (2003) used the WWW as a corpus source. They used the search engine to find the domain related text which analyzed the text and selects the

hypernym/hyponym pattern from the extracted text. As the WWW is an enormous source of text; they restricted the volume through restriction on the search query phrases.

Buitelaar *et al., (*2005) developed a well-accepted frame work of Ontology learning and proposed the "**Ontology learning layer cake**". They divided the Ontology learning activities in some well-defined sub tasks which include the term extraction, synonym extraction, concept and concept hierarchy and lastly the rules generation. This is the base of the modern Ontology learning.

Ciaramita *et al.,* (2005) developed an unsupervised model that is capable of extracting the arbitrary relationship between the concepts present in biomedical corpus, namely GENIA. They have extracted the semantic relationship between the words by inducting the dependency tree.

Hearst, M.A. (1998) developed a system that is capable of identifying the hypernym/hyponym relationship from the text based on lexico-syntactic pattern present in WordNet. The system identifies three kind of situation. First, whether both the hypernym and hyponym are present in WordNet and their relationship existence and Secondly,if the hypernym and hyponym are present but the relationship is non-existent and lastly, all of them are non - existent.

Ismail, R. *et al*., (2015) developed an Ontology learning system from Quran based on the lexico-synatactic pattern. They have succeeded to extract the "part of", "synonym" and "definition".

Panchenko, A. *et al.,* (2016) developed a system which inducts the taxonomy from the natural text. The system is tested over four languages and three domains.

Kaushik, N. and Chatterjee, N. (2018) proposed a system to identify the concept and their semantic relationship on the basis of the lexico - syntactic pattern. The proposed algorithm namely RelExOntisapplied in the agricultural domain. It has got reasonably good result. Term extraction output produces 75.7% precision and 60% recall.

Atapattu, T.*et al.,* (2017) described a system that is capable of extracting knowledge in the form of triples (concept-relation-concept). They used the subject verb object (SVO) to identify the triple extractor and used a heuristic based approach for SVO extraction from sentences. It extracts the hierarchy knowledge from the lecture slides and creates a concept hierarchy.

Snow, R. *et al.,* (2005) is one of the highly cited article in the web that have generalized the lexico - syntactic pattern to identify the hyponym/hypernym relationship. For training purpose, they collected the noun pairs from the source (corpus) by using WordNet. For the pair, they have found the sentences where both the nouns are available thereafter they parsed the sentences and automatically extracted the pattern and trained the classifier based on the features. For testing they gave the noun pair and identification as to whether they follow the pattern they incorporated the machine learning concept to the lexico-synatactic identifier of the hypernym and hyponym. They checked the methods in 6 million sentences parsed by the MINIPAR.They got significantly good result than the lexico- syntactic pattern.

Sen, S. *et al.,* (2015) described how Ontology can be used as a guide for any information extraction. It can also be helpful in the Ontology learning process. They claimed that Ontology can be used in two ways- firstly, the lexicon based approach and secondly, the thesauri based approach. The developed system enabled the tokenization, POS tagger, sentence splitter etc. processes. It also depicts that the Ontology can act as a repository for the domain knowledge and can further be used as a guide in Ontology learning process.

Oliveira *et al*., (2013) developed the TextStorm /Clouds for use in WordNet for POS tagging, and it also have the ability to parse the sentences by the use of augmented grammar; thereafter the co reference resolution and raw conceptual mapping is done. The resulting clouds are produced which are used in the creation of the formal conceptual map.

2.1.3 Relationship Extraction

Relation extraction is one of the important tasks for Ontology learning process. Following are the important relationship extraction approaches:

Turcato, D.*et al.,* (2000) developed a methodology for identification of the synonymous relationship in the aviation domain. They worked on the extraction of the domain and specific relation extraction through the manual and automated pruning methods. They have used the general purpose lexical database WordNet.

Navigli, R. *et al.,* (2003) developed a system known as OntoloLearn. The developed system has two phases - term extraction and semantic relationship extraction. To extract the semantic relation, they have used a domain appropriate semantic relation inventory. They have presented three classification rules that worked as a model to extract the semantic relationship among the concepts. For association of the appropriate relation of the

domain concepts, the inductive machine learning has been used. C4.5 is used to produce a decision tree which can select a particular rule for a particular situation.

Girju, R. *et al.,* (2003) developed a method to identify the part-whole relationship or meronymy. They have used thelexico -syntantic pattern to identify the meronymy relationship. The phrase level and sentence level pattern has been identified. This model also learned the semantic constraints from the text.

Kambhatla, (2004) developed a methodology that enabled to extract the semantic relationship among the entities. He has used the Maximum Entropy model to combine lexical, semantic and syntactic relationship.

Bunescu and Mooney (2005) proposed a kernel based novel approach to identify the relationship between entities. The approach can capture the relationship between the name entities. They generated a dependency graph and identified the shortest path among the entities.

Ciaramita, M *et al.,* (2005) developed a system in the field of molecular biology. Unsupervised learning has been used to extract the dependency and relationship among the concepts.

Banko *et al.,* (2007) developed a highly scalable system namely TEXTRUNNER and Open information Extraction Paradigm (OIE). They experimented over a wide range of domain by extracting the information from the web. The whole process is an unsupervised method of information extraction and relation extraction.

Fundel (2007) developed a system called RelExand itprovided an approach for relation extraction from the free text. It is based on natural language processing and produced the dependency parse tree. They applied their method to 1 million abstract from MEDLINE and got good performance of 80% precision and recall.

Angeli *et al.,* (2015) developed a system that is capable of extracting the triple from the free text and identified the clauses of the entities.

Kang, S. *et al.,* (2015) developed a system takes plain text as an input and converted it to Semantic Application Design Language (SADL). The converted language helped in finding out the entity present in the language.

Sordo, M. *et al.,* (2015) developed a system that does the recommendation of music on the basis of information available of the albums in a textual form. The system pre-processed the natural text and thereafter the dependency parsing and NER is done. It also

combined the result of the dependency parsing and the NER for relationship extraction from the text. On the basis of the analysis, it recommended the music.

Hearst, M.A. (1998) suggested an algorithm that is capable of extracting the relationships between the concepts. The algorithm can capture the hypernym/hyponym as well as the meronym among the concepts.

Kaushik, N. and Chatterjee,N in (2018) succeeded to extract the relation among the concept and it produced 86.89% precision.

Ismail,R. *et al.,* (2015) in their series of work developed some algorithm for parsing and extracting Ontology from the Quran. They identified some lexico - syntactic pattern which identified the semantic relationship between an object extracted from English translated Quran.

Panchenko, A. *et al.,* (2016) developed a system which identifies the relationship between the concepts present in the natural text by lexico-syntactic pattern, substrings identification and crawling which are in the focused area of the domain. These outperform the entire similar model in that area.

Atapattu, T., *et al.,* (2017) worked on concept relation extraction from lecture slide in the discipline of computer science. They have taken 11 courses, 15 slides sets with 40 slides per slide set. The found the 1838 triples extracted from 1731 sentences.

Sen, S. *et al.,* (2015) presented the way of Ontology in the relation extraction for Ontology learning.

2.2 Statistical techniques

Statistical Techniques are based on the frequency of the terms. This technique does not consider the meaning of the term. The statistical techniques involves - term extraction, concept extraction, taxonomic relation extraction and co-occurrence analysis. All the method considers the frequency distribution of the term.

2.2.1 Contrastive Analysis

Navigli, R. (2003) performed contrastive analysis, by using two type of corpus- one is domain relevant corpus and another is domain non relevant corpus.

Frantzi, K. *et al.,* (2000) described the importance of the C/NC value. The C/NC value is not purely based on the frequency of the terms present in the corpus. It gives better result than the pure frequency distribution of the terms and concepts.

Karoui, L.*et al.,* developed a system that is capable of ontological concept extraction from the HTML document. They used modified K means algorithm for Contextual Concept Discovery (CCD).

2.2.2 Co-Occurrence Measures

Resnik, P. (1999) identified the semantic similarity between words using the branch count methods. He has considered the concepts of multiple inheritances. The relatedness of the objects identified on the basis of the co-occurrence of the words.

Fortuna, B. *et al.,* (2008) proposed anovel approach of Ontology learning based on term extraction, known as-OntoTermExtraction. They have used clustering method that cluster the document. It extracted the term from the cluster and thereafter identified the keyword from the cluster.

Frikh, B. *et al.,*(2011) developed a hybrid model which considered both statistical and semantic relationships between the objects present in the text and it also developed Ontology from that text. They proposed hybrid model which is known as HCHIRSIM (Hybrid chir-statistics and similarity). The model emphasized on the coocurrence of the words to obtain the concepts.

Paiva, L. *et al.,* (2014) developed an automated Ontologylearning system using java technology and it has five subtasks. The subtasks are - document analysis, FP Growth, Association rule and Frequent Item set mapping. They mainly focused on the co-occurrence of the concepts from greater than 4-gram to 50-gram.

Xiao, L. *et al.,* (2016) suggested a method to utilize the instance relationship from DBpedia. The instances from DBpedia enforce supervised learning in any domain.

Suresu, S. and Elamparithi, M. (2016) developed the Probabilistic Relational Of Concept Extraction in Ontology Learning (PROCEOL). They find the term and concepts on the basis of co-occurrence of the pair - word. They have used markov logic network for the Ontology learning.

Idoudi, R. *et al.,* (2016) has developed an algorithm named on the Association Rule Mining in the domain of medical science.

2.3 Some prominent system of Ontology learning

Faure *et al.,* (1998); Faure and Poibeau (2000) developed a system namely ASIUM. ASIUM learns sub categorization frames of verbs and ontologies from syntactic parsing of

technical texts in natural language. It is developed in French language. The ASIUM method is based on conceptual clustering.

De Chalendar and Grau (2000) developed a system to classify nouns in context. It is able to learn categories of nouns from texts, whatever their domain is. Words are learned considering the contextual use of them to avoid mixing their meanings. This system was a pre-processor of Ontology learning.

Maedche and Staab (2000); Maedche and Volz (2001) developed a system of Ontology learning named TEXT TO ONTO.TEXT TO ONTO learns concepts and relations from unstructured, semi-structured, and structured data, using a multi-strategy method which is a combination of association rules, formal concept analysis and clustering. But this is based on the shallow natural language processing. This system fails to address complex levels of understanding. Mostly it identified concepts through regular expression.

Yamaguchi *et al.*, (2001) developed a system namely DODDLE II. DODDLE II is a Domain Ontology Rapid Development Environment. It can construct the hierarchical and non-hierarchical relationship of the domain concepts. For the hierarchical relationship it uses WordNet.

Hahn and Schnattinger (1998); Hahn and Romacker (2001); **Hahn** and Markó (2002) developed a system namely SYNDIKATE. SYNDIKATE is a system for automatically acquiring knowledge from real-world texts. It is available in German language. It has the problem of co-reference resolution.

Shamsfard &Barforoush (2002, 2003, 2004) developed a system namely HASTI. HASTI is an automatic Ontology building system, which builds dynamic ontologies from scratch. HASTI learns the lexical and ontological knowledge from natural language texts. This is available in the Persian language.

Velardi *et al.*, (2005) developed a system namely Ontolearn based on the algorithms like extractions of term, extractions of definition from natural language and also based on the parsing the definitions. Work only on plain text so it suffers from inherent Ontology learning problem.

Fortuna *et al.*, (2007) developed a system that integrates machine learning and text mining algorithms into an efficient user interface; lowering the entry barrier for users who are not professional Ontology engineers. The main features of the systems include unsupervised and supervised methods for concept suggestion and concept naming, as well as Ontology and concept visualization.

Drymonas *et al.,* 2009 developed a system namely OntoGain based on the multi word concept extraction. It gives better result than the TEXT TO ONTO in terms of multiword of multiword concept extraction.

Weichselbraun *et al.,* (2010) developed a system that integrates the external source knowledge like DBPedia and OpenCyc for getting the automatic suggestions for labelling the concepts.

Tamagawa *et al.,* (2010) discussed how to learn large scale Ontology from Japanese Wikipedia. The large Ontology includes IS-A relationship; Class-Instance Relationship; synonym; object and data type properties of domain However, a big problem of weakness in upper Ontology arose against building up higher-quality general Ontology from Wikipedia.

Xing Jiang and Ah-Hwee Tan (2010) developed a system CRCTOL, a semantic based domain Ontology learning system which is based on statistical as well as lexico-syntactic pattern. The CRCTOL is based on Concept-Relation-Concept Tuple based Ontology learning.

Gil and Martin-Bautista (2012) proposed a novel model of an *Ontology-Learning Knowledge Support System (OLeKSS)* to keep the Knowledge Support System updated. The proposal applies concepts and methodologies of system modeling as well as a wide selection of Ontology Learning processes from heterogeneous knowledge sources (ontologies, texts, and databases), in order to improve KSS's semantic product through a process of periodic knowledge updating.

Dong and Hussain (2013) developed a semi supervised Ontology learning based focused (SOF) crawler. This embodies a series of schemas for Ontology generation and web information formatting. In this system the web pages are segregated by Support Vector Machine (SVM).

Zhao & Ichise (2013) proposed Ontology learning is approach has been used for developing the Ontology. They used Linking Open Data (LOD) cloud which is collection of Resource Description Framework (RDF). They used domain Ontology for learning Ontology and called Mid-Ontology Learning. Mid-Ontology learning approach that can automatically construct a simple Ontology, linking related Ontology predicates (class or property) in different data sets.

Kumara *et al.,* (2013) gave an approach of clustering of the web services for efficient clustering. They adopted the Ontology learning to generate ontologies via hidden

semantic pattern. But they also mentioned the chances of failure of the Ontology based discovery of web services.

Dasgupta *et al*., (2013) introduced a system $DLOL_{IS-A}$ based on description logic and analysis the semantic construction if IS-A relationship of the sentence. From the IS-A relationship they derive generated the Ontology in OWL Format.

Gil and Martin-Bautista (2014) used heterogeneous sources like databases, ontologies and plain text for Ontology learning.

Liu *et al.,* (2014) generated Ontology structure called Ontology graph. The Ontology graph defines Ontology and knowledge conceptualization. The Ontology learning process defines the method of semiautomatic learning and generates Ontology graphs from Chinese text of different domains.

Petrucci *et al.,* (2016) have used the deep leaning architecture to generate text from synthetic grammar. They trained Recurrent Neural Network (RNN) based architecture to OWL formulae from text. This is first attempts Ontology learning using the deep architecture. They also claimed that this methodology reduce engineering cost domain, improved the domain independence and can deal with the various language.

El Ghosh *et al.,* (2017) developed an Ontology learning system on the legalisation domain. They proposed the middle out methodology and modularize the system into four subcomponents namely Upper Module, Domain Module, Core Module and Domain specific Module. First three are the top down and last one follows the bottom up approach.

Zhao *et al.,* (2018) developed a system namely ROCP: A Rapid Ontology Construction Platform. The system provided a question answering session with domain expert. It proposed a novel algorithm to extract the term from the domain. The system is capable to derive taxonomy on the basis of hyponymy height.

CHAPTER III

MATERIAL AND METHODS

The Ontology Learning process involves key sub areas of Artificial Intelligence such as Semantic Web, Machine Learning and Natural Language Processing. This chapter starts with the discussion about the characteristics of taxonomic text and the manual process of developing Ontology from taxonomic text followed by the technology stack of Semantic Web. The present state of the art in the area of Ontology Learning is presented along with the proposed methodology of Ontology Learning from Taxonomic Text.

3.1. Taxonomic Text and Manually developed Ontology

This research deals with the development of the Ontology from a specialized kind of text i.e. taxonomic text. The taxonomic text comes under the natural text but it is not as much unstructured like the free text. The taxonomic text has some typical characteristics which makes it more structured. The characteristics of taxonomic text are exploited to generate the Ontology from the taxonomic text.

Ontology creation task from scratch is a complex process. For creation of Ontology, in a domain, it requires in depth knowledge of that domain, as well as it requires techniques of knowledge representation. Development of Ontology requires much iteration to identify the concept to concept relationships and it includes the following major tasks:

Class Creation: Class is heart of the any ontology. The class indicates the concepts available in the domain.

Individual Creation: All instances of the class come under the Individuals of the Ontology.

Property Creation: Property is essential entity in the Ontology. A property may by two types object type property and the data type property. Object type properties are the property that establishes the relationships of Individuals to Individuals of the classes. On the other hand the Object type property establishes the relationships of Individuals with literals.

Restriction Creation: There are value constraints and cardinality constraints are the two minds of restrictions use in ontology building. Value constraints are also known as domain range constraints. The cardinality constraints restrict the no. of Individuals in the Ontology

The Soil Taxonomy Ontology (Based on USDA soil taxonomy) is considered here; which is developed by Das *et al.*, (2010, 2012) and Deb *et al.*, (2015) as a standard Ontology. Comparison of the Ontology by the developed algorithms with the standard Ontology is done.

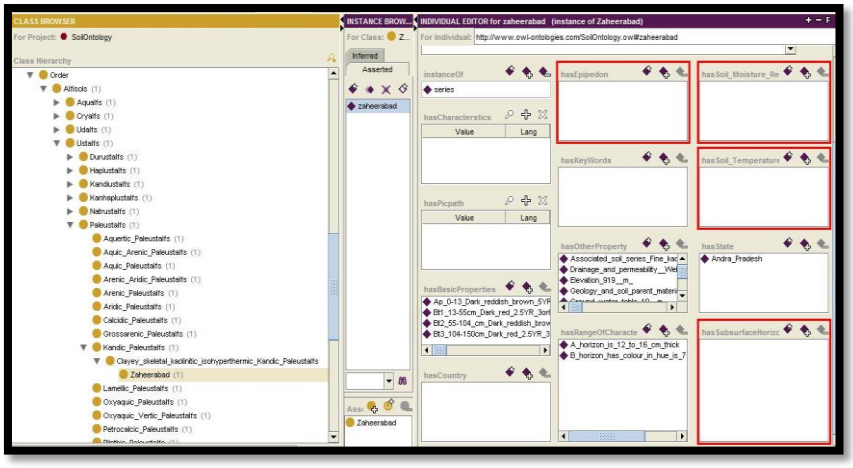


Figure 3.1 Manually developed Ontology of USDA soil taxonomy (Das *et al.*, (2010, 2012) and Deb *et al.*, (2015)**)**

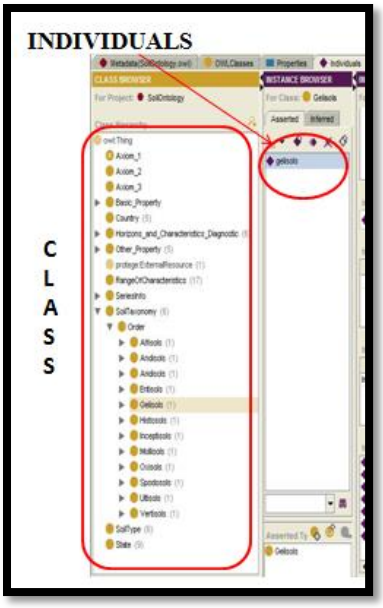


Figure 3.2 The class and Individuals of the manually developed Ontology (Das *et al.*, (2010, 2012) and Deb *et al.*, (2015)**)**

Figure 3.3 Restrictions of the manually developed Ontology (Das *et al.*, (2010, 2012) and Deb *et al.*, (2015)**)**

Another manually developed Ontology on bacteria and archaea has been used as a base standard Ontology for experimentation of the Ontology Learning process.

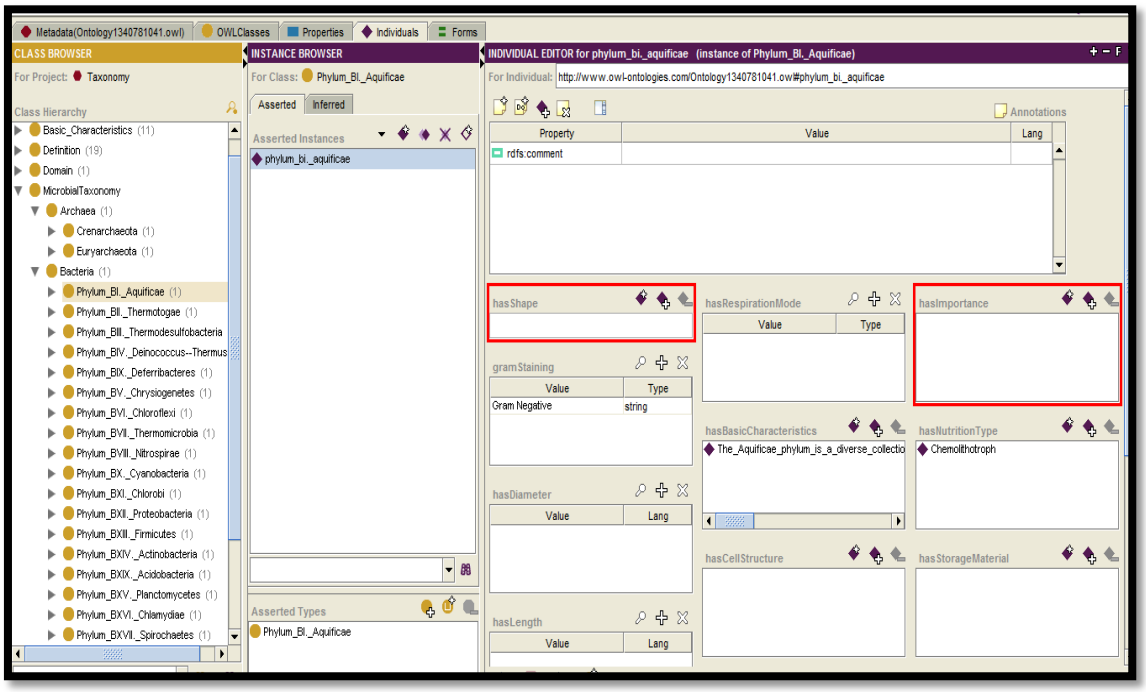


Figure 3.4 Manually developed Ontology of Bacteria and Archea

3.2. **Semantic Web**

The term "Semantic Web" was coined by Tim Berners-Lee, the inventor of World Wide Web (WWW) and the director of Word Wide Web Consortium (W3C). It can be defined as "a web of data that can be processed directly and indirectly by machines".

3.2.1 **Semantic Web Stack**

Semantic Web Stack has several layers; the functions and the relationship among the layers can be summarized as follows:

- **URI (Universal Resource Identifier) and UNICODE:** URI is a string that is in a standardized form that uniquely identifies resources. UNICODE is a standard of encoding international character set which allows all natural language for use in web.
- **XML (eXtendible Markup Language):** It's schema and names space make sure that there is a common syntax in semantic web architecture. XML is a general purpose markup language for structured documents.
- **RDF (Resource Description Framework):**Itis a data model, that represents all data in a semantic web. Primarily it was made for acting as a metadata of the web documents, but it has the capability to store and transfer the data in triple format. The representation of data is in form of *Subject-Predicate-Object*format.
- **RDFS (Resource Description Framework Schema):** It is the extended RDF vocabulary which describes the taxonomic class and properties.
- **Ontology Layer (OWL):** Web Ontology Language extends the RDF and RDFS. The OWL has it three kinds, first the OWL Lite, OWL DL and OWL Full. OWL Lite describes the taxonomies and the simple constraints. OWL DL is intended to build for the support of the description logic power. OWL Full has no expressiveness constraint.
- **Simple Protocol and RDF Query Language (SPARQL):**SPARQL is a similar SQL (Structured Query Language) for querying RDF data. SPARQL provides four forms of query results e.g. SELECT-returns list of the variables, CONSTRUCT-returns RDF graph, DESCRIBE-returns RDF graph describing the resources and ASK- return Boolean value whether query pattern matches or not. Some key words like ORDER BY, DISTINCT, OFFSET and LIMIT acts like the SQL query.

3.3. **Ontology Learning Layer Cake**

Buitelaar P. *et al*., (2005) proposed the methodology which decomposes the Ontology Learning tasks into several subtasks. The work beautifully summarizes the whole Ontology task into independent module, which gives the Ontology researcher a prescribed path to do Ontology Learning tasks.

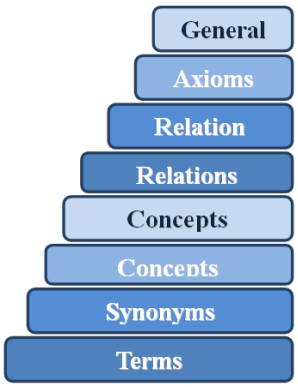


Figure 3.5 Schematic representation of the 'Ontology Learning Layer Cake' (Figure reproduced from Buitelaar P. *et al.*, 2005)

The first layer describes the extraction of the "Terms"from the domain documents. The "Terms" act as fundamental semantic unit of the domain. It is also used as a potential source of ontological class. Second layer normalizes the extracted Terms and replaces the "Terms" with "Synonyms". The first two layers are known as the **lexical layer**. The third layer is known for the identification of the Terms which can be used as the "Concepts". The Terms can be considered as Concepts, if it follows some special criteria although it is somewhat a controversial issue. Some literature suggests the Concepts are a cluster of terms in a particular domain. The Terms can act as a concept, and it is dependent on the domain context. The concept can be intentional as well as the extensional point of view. Sometimes, the concept may be extracted from the lexical pattern of the extracted Terms. Next layer is devoted to find the taxonomic and non- taxonomic relations. It can be done in three ways firstly, the lexico syntactic pattern; secondly, the hierarchical clustering and lastly, the term sub sumption method of taxonomic relation extraction. Next two layers are used for relationship extraction and relation hierarchy extraction. Last two layers are about the General axioms and Axioms schemata.

3.4. **Working pipeline of the of Ontology Learning Process from taxonomic text**

The Ontology Learning from the taxonomic text has been summarized in the diagram (Figure 3.6). First task of the whole Ontology Learning process starts with the development of the corpus. Next, the usage of the hierarchical clustering is there which helps to find out the parent child relationship among the object using the WordNet API. Simultaneously, the extraction of the taxonomic relation is done by the lexico-syntactic pattern.

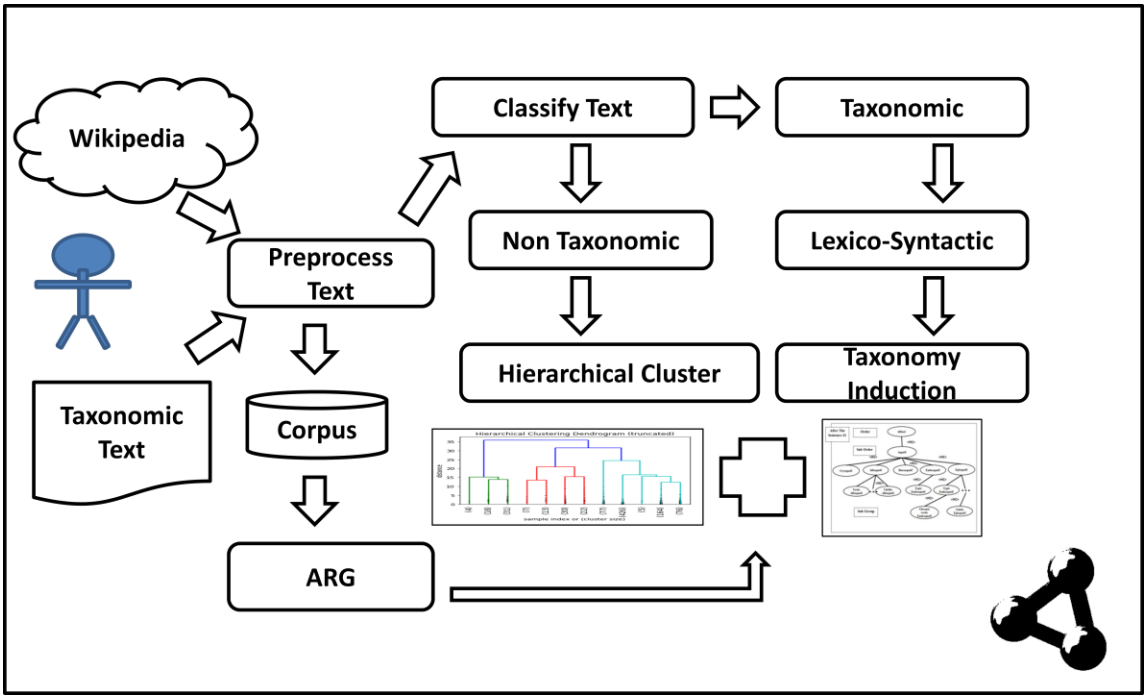


Figure 3.6 Working pipeline of the Ontology Learning process from the taxonomic text

3.4.1. **Corpus Development**

Any natural language task starts with suitable corpus development. In this case, the development of our Corpus is from two sources – firstly, the automated scrapping from Wikipedia and secondly, the standard taxonomy book from a particular domain. For the development of the corpus, a couple of algorithms have been developed and are described in **`Algorithm 3.1.`** Figure 3.7 describes the process flow of Corpus development.

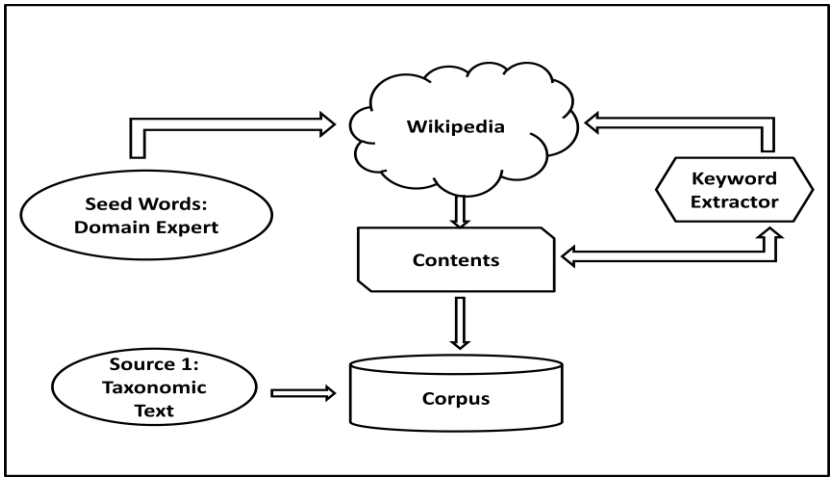


Figure 3.7 Methodologies of Corpus development

An algorithm has been developed for the enhancement of the corpus by using the resource of Wikipedia. The first set of keyword is supplied by the domain expert. Based on

the supplied keyword, a second level of searching is done on the basis of the extraction of the keywords from the first level.

Algorithm 3.1 Corpus Enhancement by Wikipedia Scraping

Step 1: Keywords from domain expert

Step 2: for k in Keywords:

Step 2.1: Download Content from Wikipedia

Step 2.2: Append to Corpus

Step 2.3: Extract Keywords from downloaded content [using rake library]

Step 2.3.1: For k in New-Keywords:

Step 2.3.1.1 Download Content from Wikipedia

The corpus development process involved many of the packages which are available in the python. Some of the prominent packages involved in the corpus development are described below:

nltk: NLTK or Natural Language Tool kit is a very powerful tool, facilitates working with human language or the natural languages. It also provides the facility to access more than 50 corpora and lexical resources. Word Net is one of the prominent resources that can be accessed through the **nltk** library.

From **nltk.tokenize** package**word_tokenize, sent_tokenize** etc. functionality has been used for developing the corpus.

nltk.corpus: Another essential package in **nltk** library is **nltk.corpus**may be used for the corpus management. It provides the access facility to a wide range of corpus (e.g. brown corpus).

wikipedia: Wikipedia is a specialized library in python which facilitates the access and manipulation of the content of the Wikipedia. The method namely **summary()** has been used to scrap the contents of the Wikipedia.

rake: For initial enhancement of the corpus , the classic **Rapid Automatic Keyword Extraction(rake)** library has been used . It is well suited to find the single word and multiword keywords from a single document.

Algorithmia:Algorithmia is a library for co reference resolution. The **nltk.PorterStemmer** and **nltk.WordNetLemmatizer**are used for normalization of the text. Normalization also helps in the increase the density of the important words

3.4.2. Data Preparation

Ontology Learning is a lengthy task which goes through different pipeline to extract different useful information for structuring the unstructured knowledge available in the natural text. Following data preparation or text preparation techniques are used in the different stages of Ontology Learning:

- **Sentence Detection:** This is the first step of any text preparation technique in Natural Language Processing task. As the name suggests, this technique is used to segregate the sentence from the given taxonomic text.

As in the earlier section 3.4.1., the **`nltk.tokenize`**from **`nltk`**library has been used for detection of the sentence from the corpus.

Alfisols that have a thermic or warmer soil temperature regime tend to form a belt between the Aridisols of arid regionsand the Inceptisols ,Ultisols , and Oxisols in areas of warm ,humid climates . Where the soil temperature regime is mesic orcooler , the Alfisols in the United States tend to form a belt between the Mollisols of the grasslands and the Spodosols andInceptisols in areas of very humid climates .In regions of mesic and frigid soil temperature regimes, Alfisols are mostly on late Pleistocene deposits or surfaces. Inwarmer regions, they are on late Pleistocene or older surfaces if there are only infrequent years when the soils lose bases by leaching or if there is an external source of bases, such ascalcareous dust from a desert. Most Alfisols have a udic ,ustic , or xeric moisture regime ,and many have aquic conditions .
Source: Key to Soil Taxonomy[2014]

Figure 3.8 Example of some text snippet of the taxonomic text

Alfisols that have a thermic or warmer soil temperatureregime tend to form a belt between the Aridisols of arid regionsand the Inceptisols ,Ultisols , and Oxisols in areas of warm ,humid climates .
Where the soil temperature regime is mesic orcooler , the Alfisols in the United States tend to form a belt between the Mollisols of the grasslands and the Spodosols andInceptisols in areas of very humid climates .
In regions of mesic and frigid soil temperature regimes, Alfisols are mostly on late Pleistocene deposits or surfaces. Inwarmer regions, they are on late Pleistocene or older surfaces if there are only infrequent years when the soils lose bases byleaching or if there is an external source of bases, such ascalcareous dust from a desert.
Most Alfisols have audic, ustic, or xeric moisture regime, and many have aquic conditions.

Figure 3.9 Detection of the sentences available of the given text

- **Tokenization:** After sentence detection, the second task of data preparation is tokenization of the sentences. Tokenization splits the sentence into list of words.

`nltk.tokenize:` From the **ntlk.tokenize**package**`word_tokenize`** has been used for detection of the words.

```
['Alfisols', 'that', 'have', 'a', 'thermic', 'or', 'warmer', '
soil', 'temperature', 'regime', 'tend', 'to', 'form', 'a', 'be
lt', 'between', 'the', 'Aridisols', 'of', 'arid', 'regions', '
and', 'the', 'Inceptisols', ',', 'Ultisols', ',', 'and', 'Oxis
ols', 'in', 'areas', 'of', 'warm', ',', 'humid', 'climates', '
.']
```

Figure 3.10 Tokenization of first sentence (Figure: 3.9) into words

- **POS Tagging:** Parts of Speech tagging is an essential task for NLP. It annotates each and every word with its corresponding Parts of Speech. POS helps to identify the pattern present in the text.

Nltk POS Tagging: Parts of tagging can be done by the **`nltk.pos_tag()`** method. The `pos_tag()` method takes tokenized sentences as an arguments and returns the set of paired element i.e. actual word and corresponding parts of speech.

```
[('Alfisols', 'NNS'), ('that', 'WDT'), ('have', 'VBP'), ('a', 'DT
'), ('thermic', 'JJ'), ('or', 'CC'), ('warmer', 'JJ'), ('soil', '
NN'), ('temperature', 'NN'), ('regime', 'NN'), ('tend', 'VBP'), (
'to', 'TO'), ('form', 'VB'), ('a', 'DT'), ('belt', 'NN'), ('betwe
en', 'IN'), ('the', 'DT'), ('Aridisols', 'NNP'), ('of', 'IN'), ('
arid', 'JJ'), ('regions', 'NNS'), ('and', 'CC'), ('the', 'DT'), (
'Inceptisols', 'NNP'), (',', ','), ('Ultisols', 'NNP'), (',', ','
), ('and', 'CC'), ('Oxisols', 'NNP'), ('in', 'IN'), ('areas', 'NN
S'), ('of', 'IN'), ('warm', 'NN'), (',', ','), ('humid', 'NN'), (
'climates', 'NNS')]
```

Figure 3.11 Detection of Parts of Speech tagging (POS) of first sentence (Figure: 3.9)

- **Stemming and Lemmatization:** Stemming and lemmatization both the methods are used for the normalization of the text. Stemming of the word means the removal of the beginning or the end of the word to extract the root word. For example, the words ‘studies’, ‘studying’ normalized into ‘studi’ and ‘study’. For studies we get ‘studi’ by removing the end part es and for studying we get study by removing the end part ing. On the other hand, in case of lemmatization both studies and studying are converted into the word ‘study’.

`nltk.PorterStemmer` and **`nltk.WordNetLemmatizer:`** PorterStemmer and WordNetLemmatizer are two prominent packages for stemming and lemmatization in the natural language task. For stemming the method **`stem ()`** and for lemmatization **`lemmatize()`** is used. Both the method takes word as arguments and returns the stemmed or lemmatized word respectively.

- **Feature Engineering**

Natural Language Processing (NLP) is a cumbersome task because direct computation is not possible in the natural language. The language must be converted into some numeric representation so that the computational technique or machine learning technique can easily be implemented into the natural language. This conversion of the text into numeric representation is known as Feature Engineering. For engineering the features, the following three features of engineering technique namely; Count vector, TF-IDF and the Word Embedding using Word2Vec [Mikolov *et al.,* (2013)] have been used. A brief comparative study is also done among the feature engineering techniques.

- **Count Vector:** Count Vector is based on the frequency of the word in a particular document. In this representation, the frequency of the unique word is attached with that word and used as an input of machine Learning as a vector. Equation 3.1 depicts the formula of count vector.

$$\boldsymbol{lef_{l.d} = lexical\ entry\ frequency\ l\ in\ document\ d} \quad \text{.....Equation 3.1}$$

This is the simplest representation of the terms of the text but it is unable to capture the inherent relationship present in the text.

- **TF-IDF (Term Frequency and Inverse Document Frequency):** This method of text representation is an advanced form than the count vector representation. In this method of text representation, two types of frequency of a lexical entry are considered. It considers single document frequency and also considers frequencies in an overall document. Equation 3.2 depicts the formula for TF-IDF.

$$\boldsymbol{tfidf_{l,d} = lef_{l,d} * \log(\frac{D}{df_l})} \quad \text{.....Equation 3.2}$$

$$lef_{l.d} = lexical\ entry\ frequency\ l\ in\ document\ d$$

$$df_l = overall\ document\ frequency\ of\ lexical\ entry\ l$$

$$D = lexical\ entry\ frequency$$

The TFIDF model is advantageous over count because it consider a word on the basis of the frequency of in a particular document as well as the entire documents.

- **Word2Vec: Embedding techniques for word**

Word2Vec is a word embedding technique that represents an individual word as a real valued vector in a predefined vector space. This model of word representation is proposed by Mikolov *et al*. (2013). They proposed 2 models to produce the vector representation of the words; namely, CBOW or Continuous Bag of Words and SKIP Gram model.

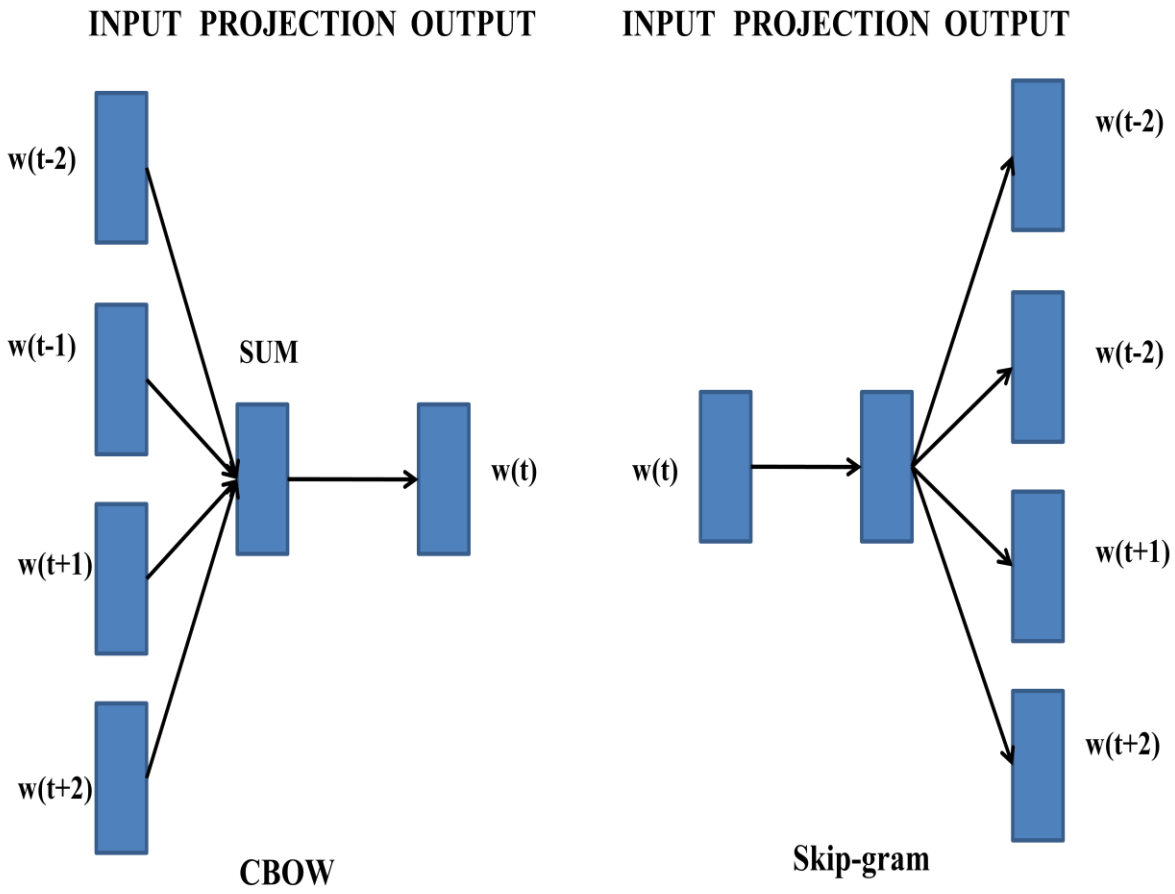


Figure 3.12 CBOW and Skip-gram architecture in Word2Vec (Figure reproduced from Mikolov *et al.*, 2013)

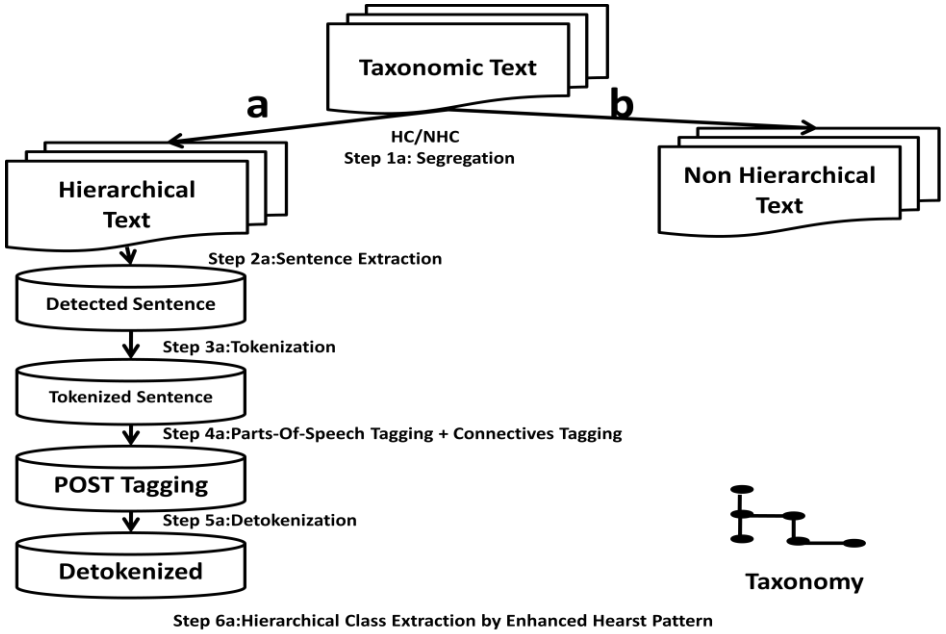


Figure 3.13 Process of Taxonomy Induction

For feature extraction python has several tools to deal with. Following feature extraction tools has been used in this study:

`sklearn:sklearn` is a robust machine Learning library originated from Google summer code project 2017. **`sklearn`**build on the stack of packages like **`numpy, scipy, matplotlib, IPython, sympy`** and **`pandas.`** The major activities that can be achieved by the **`sklearn`** are clustering, cross validation, Datasets tests and generation, dimensionality reduction, ensemble methods, feature extraction, feature selection, parameter tuning, manifold Learning and supervised models.

Feature extraction is done for broad three categories firstly the Count Vectorize, secondly the TFIDF and lastly Word2Vec. First and second category used**`sklearn.feature_extraction.textto`** extract the features. **`CountVectorizer()`** and **`TfidfVectorizer()` `method`** has been used respectively for extracting count vector and TFIDF vector. Count vector extraction is done on the basis of word level and TFIDF vector has been extracted on the basis of word, character and n-gram level.

The Word2Vec conversion **`keras.preprocessing`**package has been used to create two shallow networks e.g. CBOW and Skip-Gram for extracting the numerical representation of the word as COUNT VECTOR and TFIDF.

3.4.3 Classification of the text into Hierarchical and Non hierarchical

In this study the taxonomic text available in agricultural domain has been used. One of the essential features of the taxonomic text is the distinguished taxonomic hierarchy present in the text. The taxonomic text is naturally divided into two parts- first, the text which contains the taxonomic hierarchy and second, the text which does not contains the taxonomic hierarchy. In this research, the various methods to classify the text have been studied. The following classification techniques has been used and studied to develop the model and the developed model is used to classify the text into hierarchical and non hierarchical text:

Algorithm 3.3 Classify text into Hierarchical and Non hierarchical

Step 1: Input Text

Step 2: Vectorize

Step 3: Train using ML Technique

Step 4: Test

- **Classification Techniques used for segregation**

In this research one study has been conducted on the taxonomic text to identify whether the encoding techniques have significant influence on the classification techniques. For classification , some standard classification techniques are used - **Naïve Bayes Classifier, Linear Classifier, Support Vector Machine, Random Forest, eXtreme Gradient Boosting and Multilayer perceptron.**

Naïve Bayes Classifier is Machine Learning classification techniques based on Probability. **Linear Classifier** classifies the objects on the basis of linear combination of characteristics of the object. **Support Vector Machine** is a supervised hyper plane discriminatory classifier. It classifies objects by some hyper plane. For two dimensional spaces, it classifies object by the line and for three dimensional spaces it classifies objects by the plane; generally these planes are called as hyper plane. **Random Forest** is an ensemble learning process for regression and classification. It creates multiple decision trees to reduce the chance of over fitting like decision tree. **eXtreme Gradient Boosting** regularize version of the of the gradient; boosting helps to avoid the over fitting.

In the section 3.4.2 there is an introductory discussion on **`sklearn`** and found that the **`sklearn`** is a very large Machine Learning library which is robustly capable of many Machine Learning tasks. Following packages are used to classify the taxonomic text into hierarchical and non hierarchical categories:

a) **sklearnmodel_selection,**

b) **preprocessing,**

c) **linear_model,**

d) **naive_bayes,**

e) **metrics,**

f) **svm,**

g) **decomposition,**

h) **ensemble**

3.4.4. Heuristic Keyword Extraction Methodology

Literature suggested some methods that have use in general Ontology Learning methods for identification of the term and the concepts in the natural text. Those are Count Based term extraction, TF-IDF and Dictionary based term extraction.

In this research work, one method has been developed for extracting the keyword from the Rapid Automatic Keyword Extraction (RAKE). In the classical method of the RAKE, sentences are splitted into arrays of word and then it is splitted into sequence of words. Calculations of the metrics of particular keywords are done. After that, the selections of the keywords on the basis of the calculated metrics score are done. RAKE tool which is made available in the python are used.

The **rake_nltk** package has been used for extracting the keywords. From **rake_nltk**, **Rake()** have been used for initialization of the model. **extract_keywords_from_text()** and **get_ranked_phrases ()** has been used for the extraction of the keywords and getting descending order score wise keyword list respectively.

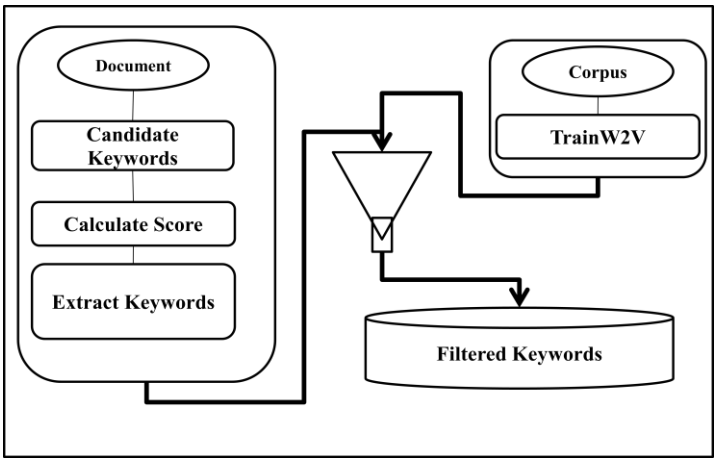


Figure 3.14 Hybridization RAKE and W2V for heuristics Keyword Extraction Methods

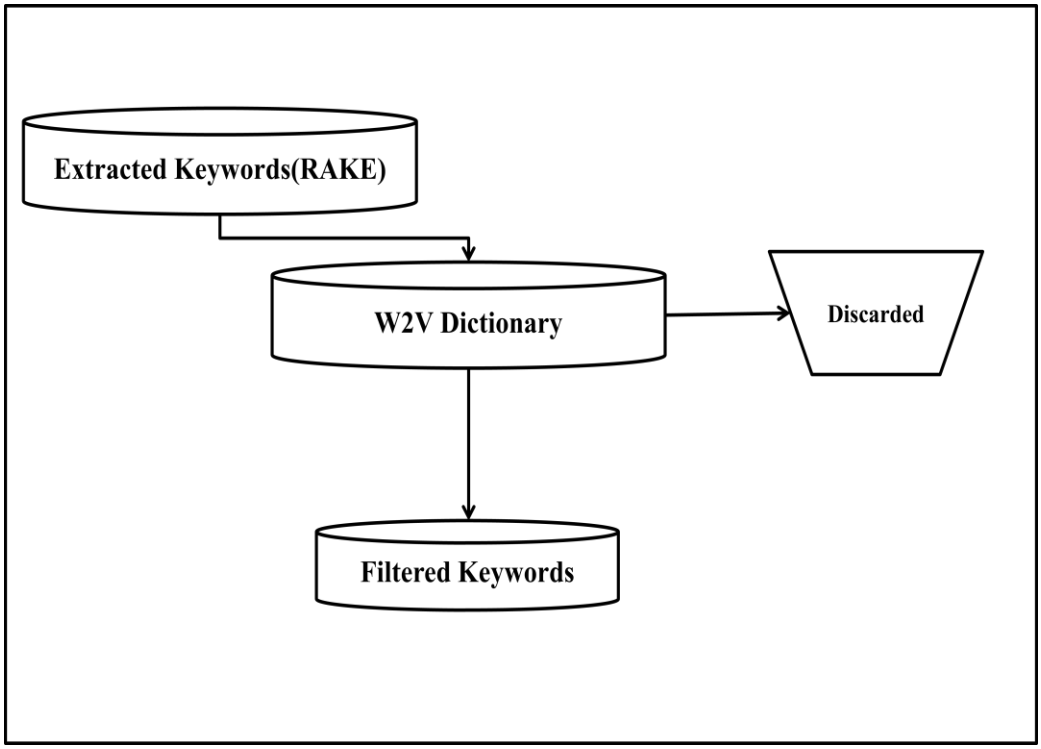


Figure 3.15 Simplified flow chart of Hybridization of Keyword Extraction Methods

Classical RAKE does not consider the semantics of the word. A hybrid heuristic method that combined the RAKE with W2V is used here. This new method considers the principle of RAKE. W2V works as a guide to identify which identified keyword is semantically correct.

3.4.4. Hierarchy Induction from Taxonomic Text

The study of the Taxonomic text suggests that this type of text has two distinct portion of text. The core taxonomic part of the text has uniform pattern for hierarchical relationships and the other part of the text is more like the plain text. The section following two sub sections describes the induction of the hierarchy from both type of text.

- **Taxonomy induction From Hierarchical Part of the Text**

Taxonomy Induction is the core work for Ontology Learning task. The taxonomy induction can be extracted using the Hearst Pattern. A tree based methodology has been suggested here with enhanced Hearst Pattern to induct the Ontology from the taxonomic text.

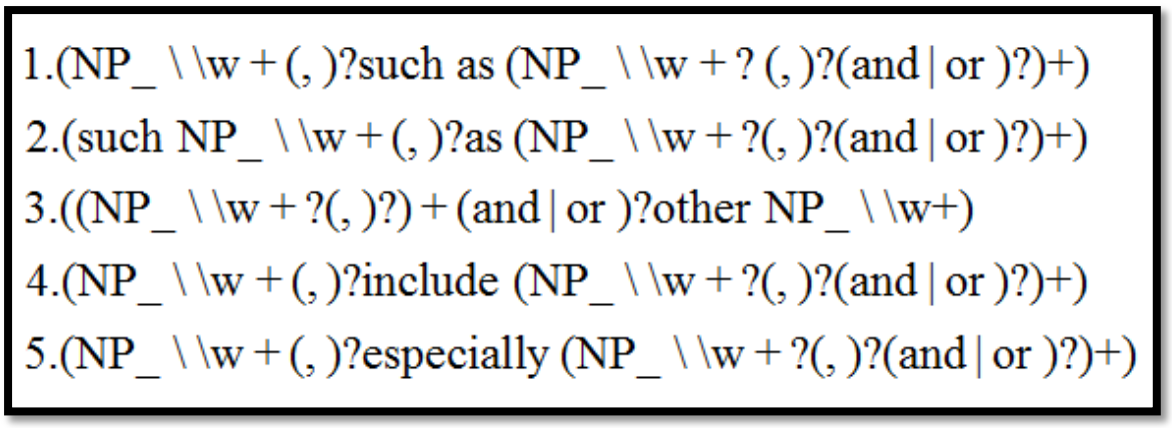


Figure 3.16 Pattern to Identify the Hyponyms and Hypernym (Figure reproduced from M.A Hearst *et al.*, 1998)

```
Algorithm 3.4 Taxonomy Induction by Enhanced Hearst Pattern

Step 1: Input Hierarchical Text

Step 2: H<- Hearst Pattern ()

Step 3: If (H is True)

      Step 3.1: Find (Parent, Child)

         Else

      Step 3.1: Return Null

Step 4: End
```

```
Algorithm 3.5 Recursive Taxonomy Induction from Hierarchical
text

Step 1: Input Text

Step 2: findSubclass(level,line)

            If (NextLineLevel==Level+1)

                  Parent<-line

                  Child<-line+1

                  findSubclass (level)

Step 3: End of the function
```

- **Taxonomy induction From Non hierarchical Part of the Text**

The Vectorize data are used to make hierarchical cluster that can help in identification of the parent child relationship from the non taxonomic data. Taxonomic class has been used for identification of the number of cluster in the non taxonomic class.

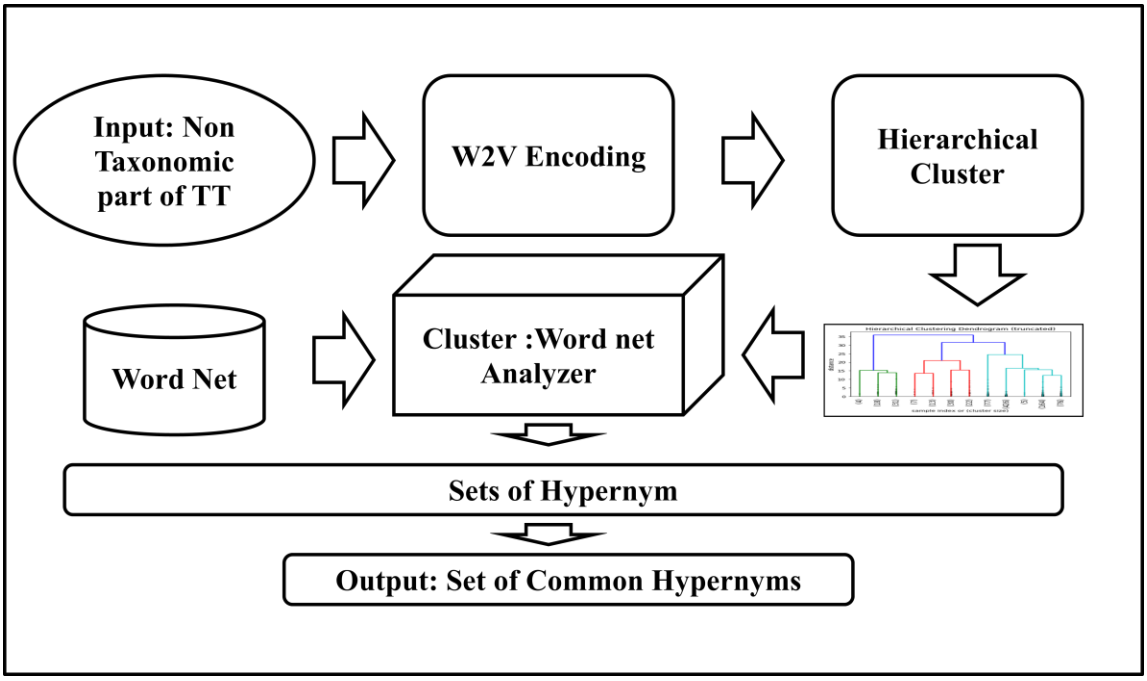


Figure 3.17 Extraction of generalized class available in the taxonomic text

Algorithm 3.6 Hierarchical Clustering for property identification

Step 1: Input Nonhierarchical Text

Step 2: Encode in W2V

Step 3: AglomerativeClustering(No.of Taxonomic Class)

3.4.5 Association Rule Generation: The hierarchy of the taxonomic and the non taxonomic text. This is important for extraction of the relationship among the taxonomic class and non taxonomic class. Association rule (Agrawal *et al.*, 1993) mining is a methodology for extraction of association between objects on the basis of two measures - support and confidence.

The support means how frequently an item set occur in the data and the confidence means often the rule generated are found to be true.

This well established theory of association has been used for identification of the association of taxonomic class and the property of the taxonomic class automatically.

3.5. Tools

The whole Ontology Learning process involved the following tools:

3.5.1 Protégé

Protégé is an environment for Ontology development. It provides the functionality for editing classes, slots (properties) and instances. One of its strengths is that it can automatically generate a user interface from class definitions, and thus can support rapid knowledge acquisition. At its core is a frame-based knowledge model (Noy *et al.,* 2001) with support for meta classes. Protégé can be extended with backend for alternative file formats like XML, RDF, and OWL. Protégé not only allows developers to extend the internal model representation, but it also allows customizing the user interface freely. Protégé's user interface consists of several screens called tabs, which display different aspects of the Ontology in different views. Each of the tabs can be filled with arbitrary components. Most of the existing tabs provide a tree-browser view of the model, with a tree on the left and details of the selected node on the right hand side. The details of the selected object are typically displayed by means of forms. The forms consist of configurable components called widgets. Typically, each widget displays one property of the selected object. There are standard widgets for the most common property types, but Ontology developers are free to replace the default widgets with specialized components. Widgets, tabs and back-ends are called plug-ins. Protégé's architecture allows developers to add and activate plug-in arbitrarily.

3.5.2 OWLSyntax and ProtégéOwl

These two frameworks provide the facility to interact with the Protégé software. Protégé is used for the development of the Ontology. By using the API which is provided by the OWLSyntax and ProtégéOWL the extracted text can be used for the automated Ontology building.

3.5.3 WordNet

WordNet (Miller *et al*., 1998) is a large lexical database of English. In this database, the nouns, verbs, adjectives and adverbs are grouped together. It resembles to the thesaurus that makes the grouping of the words on the basis of their meaning. They define language as a pair of words, in which the first word describes the finite alphabet and the second one describes the sense of an alphabet. A word uses a particular supporting set of words that act as a context of the word. WordNet contains the following semantic relations:

Synonym: Perhaps the most important resource of WordNet is the synset or the synonyms.

Antonym: Opposite name of the words.

Hyponym: Hyperonymy, hyponymy or the ISA relationship is one of the most important kinds of relationship. This relationship is a transitive relationship and it helps to find out the parent child relationship of given texts.

Troponymy: it captures the parent child relationship between the verbs.

Entailments: Capture the relation between verbs.

According to WordNet website (https://wordnet.princeton.edu/) WordNet 2.1 contains 117097 Nouns with 81426 synset and 145104 word-sense pairs; 11488 Verbs with 13650 sunset and 24890 word-sense pairs; 22141 Adjectives with 18877 synsets and 31302 word-sense pairs; 4601 Adverbs 3644 synset and total 5720 word-sense pairs. In total, the WordNet 2.1 contains 155327 strings; 117597 synsets and 207016 word sense-pairs.

3.5.4 JavaWordNet Library (JWNL)

JWNL is a Java API to access the WordNet relational databases. It allows a developer the access to the use of WordNet resources in their programming environment. It contains 10 packages with approx 68 classes and interfaces. The JWNL provides a crisp list of classes and interfaces that enable the developer for easy familiarization with whole API and use it efficiently.

3.5.5 DL4J (Deep Learning For JAVA)

It is an Open source and distributed java framework, released under Apache License 2.0 for Deep Learning facility. It is an extremely huge framework that captures almost all the aspects of deep Learning in Java. Presently, DL4J has approximately 900 packages and 5850 classes and it is constantly increasing. This framework is compatible with Java and other programming languages like Scala, Closure etc. Training with DL4J occurs in cluster. It is compatible with Hadoop-YARN and Spark. It also integrated with CUDA kernel to conduct pure GPU operation and works with distributed GPU. DL4J supports many of the Deep Learning tasks and is divided into different modules. The NLP module for training and classification purpose is used.

3.6. Ontology Evaluation

The evaluation of Ontology is a very cumbersome and challenging process. Unavailability of standard measures of Ontology evaluation is one of the biggest challenges in Ontology Learning. In this research work, a methodology for induction of the Ontology from the taxonomic text is developed. Fortunately, the developed algorithms were evaluated with reference of manually developed USDA Soil Ontology. It is assumed

that the manually developed Ontology is a standard one and the output of the algorithms with the standard Ontology is compared. Division of the evaluation process into three broad categories is done. Firstly, evaluation of the lexical entry and secondly, the evaluation of the hierarchical induction from the taxonomic text is done. Lastly, the evaluation of the non taxonomic relation which do not comes directly under the hierarchical relationships is done.

The following evaluation measures are adopted for evaluation of the result of the developed algorithms:

3.6.1 Precision Recall and F-Measures

The Lexical Entry is the important word that has the significant meaning in the domain. For evaluation of the lexical entry we have used some

Precision and Recall are two popular measure of performance of the information retrieval system.

$$precision = \frac{tp}{tp + fp} \dots\dots\dots\dots\dots\dots\dots (1)$$

$$recal = \frac{tp}{tp + tn} \dots\dots\dots\dots\dots\dots\dots\dots . (2)$$

$$tp: True\ positive$$

$$tn: True\ negative$$

$$fp: False\ positive$$

The hierarchy extraction from the taxonomic text has two aspects the core hierarchy of the taxonomic text and the hierarchy extracted from the non taxonomic text. The taxonomic hierarchy is evaluated with the reference ontology. For evaluating the non taxonomic hierarchical relations the human evaluation (Muhammad *et al.,* 2018)

3.6.2 Support, Confidence and Lift

The generated rules by using the principle association rule generation are described in the section 3.4.5 can be evaluated by precision recall and human evaluation. Prominent measurement of rule to be included in the rule list is support and confidence. Support Confidence scatter graph is used for visualization of the rules generated. Also measurement of the Lift for the visualization of the relevance of the rule that is generated from the text is done.

CHAPTER IV

RESULTS AND DISCUSSION

This chapter is dedicated to depict and discuss the results of Ontology Learning from Taxonomic Text. The objective wise results of this research work are included in the following sections and the critical discussion of the result has also been done in this chapter. The chapter is mainly divided into three sections. Section 4.1, 4.2 and 4.3 discusses the results of the Objective 1, 2 and 3 respectively.

4.1 Ontology Learning Algorithms

Under Objective 1, the three major categories of algorithms have been studied. The literature survey of the Ontology Learning consists of mainly three categories of algorithms i.e. Lexical Entry Extraction, Taxonomy Induction and Non Taxonomy Extraction.

Table 4.1: Activity list under study of Ontology Learning Algorithms

- **Lexical Entry Extraction**
  - **Count Vector**
  - **TFIDF(Term Frequency Inverse Document Frequency)**
  - **Dictionary**
- **Taxonomy Induction**
  - **Top down hierarchy**
  - **Bottom up hierarchy**
- **Non Taxonomy Relation Extraction**
  - **Association rule mining**

4.1.1 Lexical Entry Extraction

The starting point of Ontology Learning algorithms pipeline is lexical entry extraction activities. The Lexical entry of a document suggests that important words, that have the potential to be a part of the Ontology. A traditional algorithm uses mainly three

categories of the formulas - the Count Vector, TFIDF and the Dictionary. The theoretical aspects of this category of lexical entry extraction techniques are discussed in the sub section **Feature Engineering** of the section 3.4.2 .In this chapter, the results of the first two categories i.e. the Count Vector and TFIDF particularly on taxonomic text is discussed.

- **Count Vector**

Figure 4.1, Figure 4.2, Figure 4.3 shows the results of the Count Vector for finding the unigram, bigram and trigram from the text. The figure shows that the count vector is able to capture a frequently occurring single word keyword as well as the multi word keyword. The following figures are generated from the Corpus which is developed from the USDA soil taxonomy and Wikipedia.

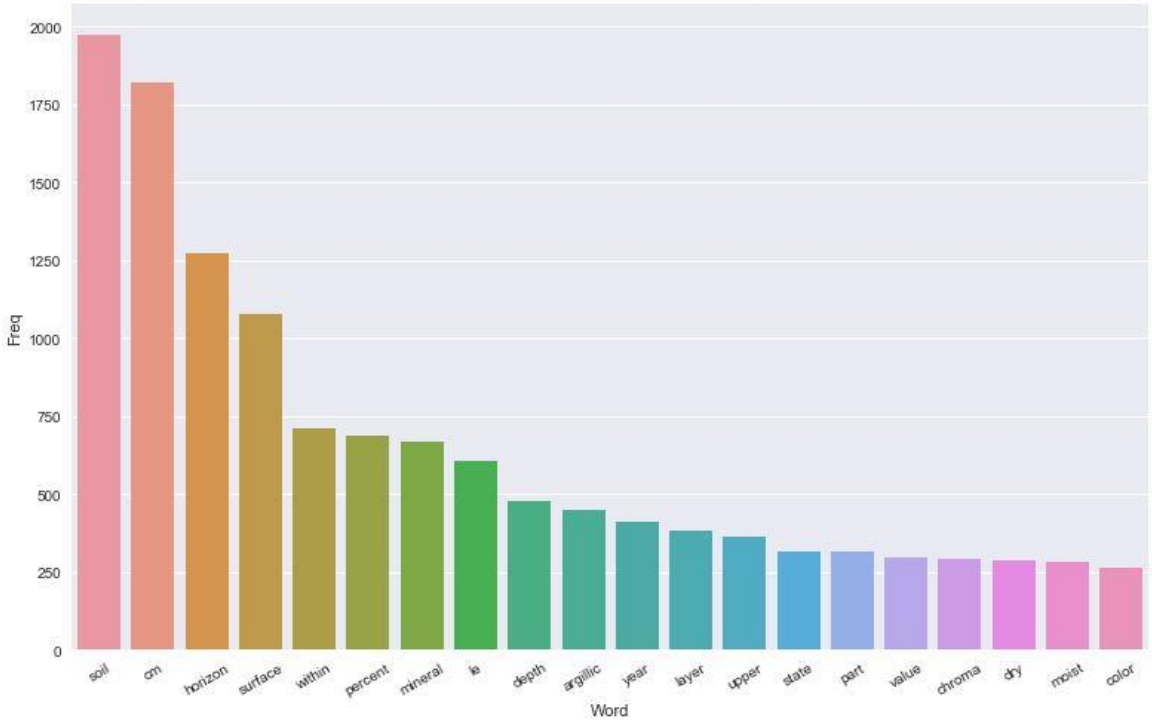


Figure 4.1 Unigram Higher Frequency Value in Corpus: USDA Soil Taxonomy (Order-Alfisols)

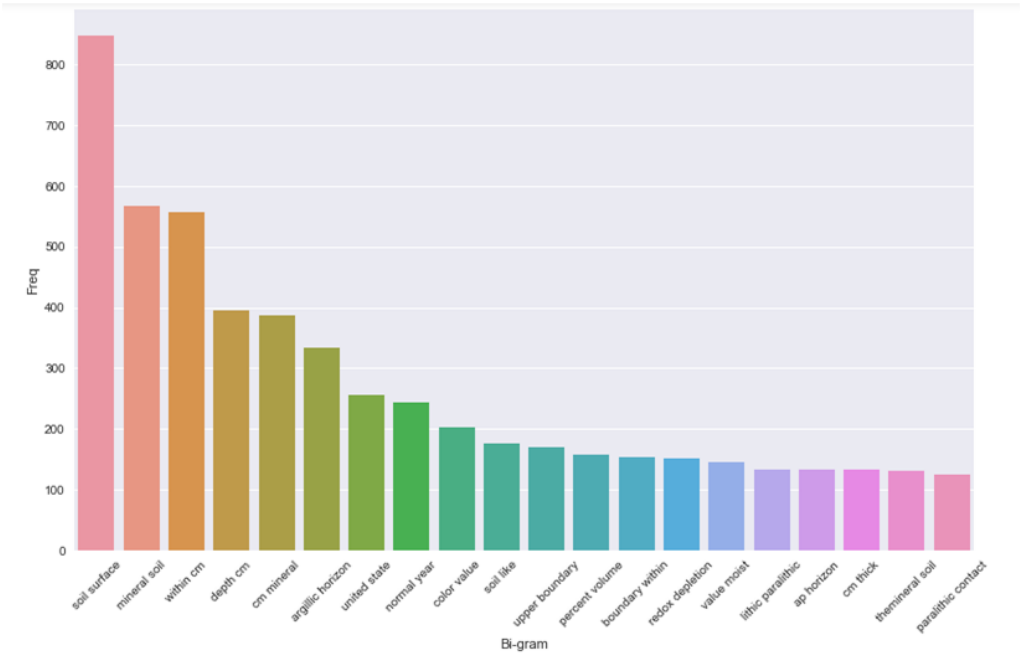


Figure 4.2 Bigram with Higher Frequency Value in Corpus: USDA Soil Taxonomy (Order- Alfisols)

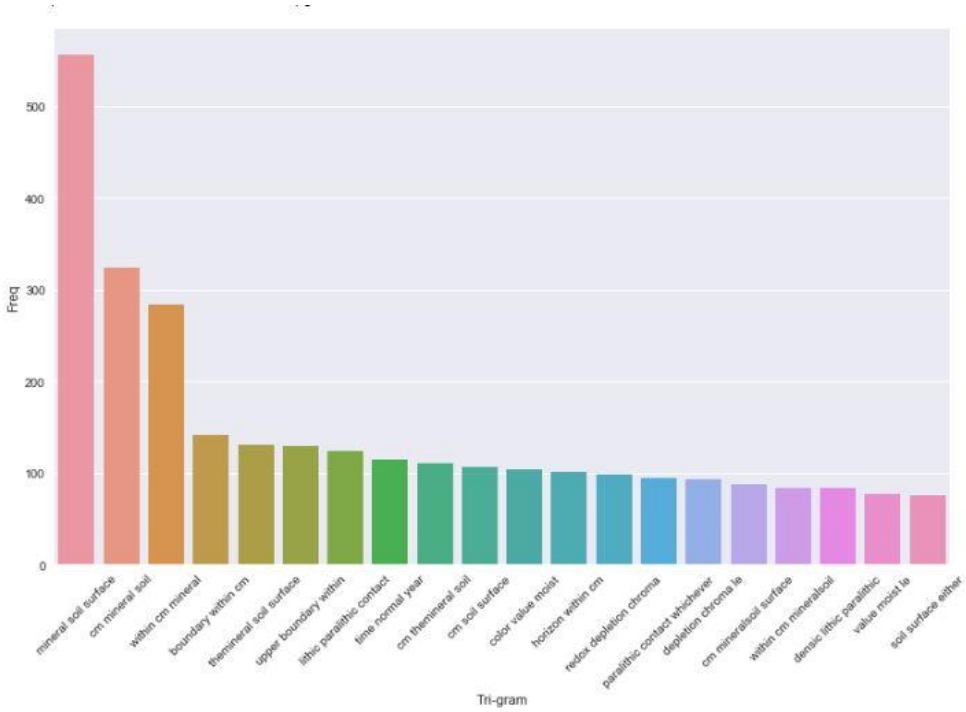


Figure 4.3 Trigram with Higher Frequency Value Corpus: USDA Soil Taxonomy (Order-Alfisols)

The size of the developed Corpus is of approximately 1 million words. The Unigram count vector focused mainly on the frequently occurring single word. Twenty frequent words have been shown in the Figure 4.1, 4.2 and 4.3.

Similar to Unigram, Bigram and Trigram all are used to identify the important multiword term in the taxonomy. For taxonomic text, the n-gram term is very important because most of the taxonomic terms are multiword.

- **TFIDF (Term Frequency Inverse Document Frequency)**

The formula and concept of the TFIDF is already been discussed in the sub section **TF-IDF (Term Frequency and Inverse Document Frequency)** of the section 3.4.2 The following figure depicts the results of TFIDF in the developed Corpus:

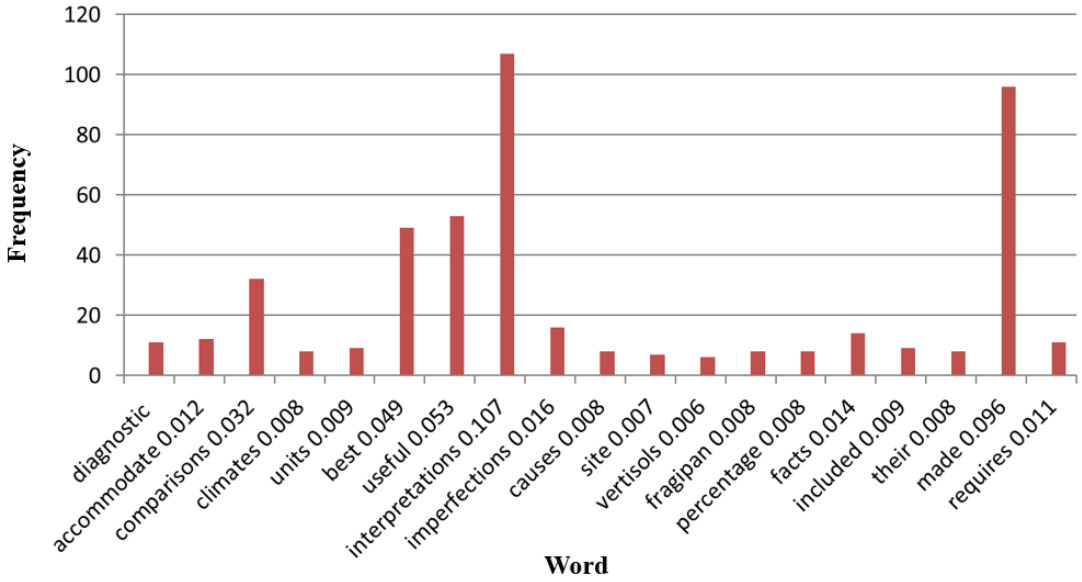


Figure 4.4 Lexical entry extractions from the taxonomic text using TFIDF

The above described two methods of lexical entry extraction are based on the frequency of the particular lexicon. But in case of taxonomic text, lexical entry extraction from the taxonomic text is not sufficient because many of the extremely important terms in the taxonomic text has very less frequency sometimes it is as less as once. From both the methods, we have seen that it can identify the higher frequently occurred term like ‘soil’ for unigram, ‘soil surface’ for bigram and ‘mineral soil surface’ for trigram. But it is unable to identify the terms like ‘udalfs’, ‘typic udalfs’ etc. which is more important for USDA soil taxonomy. The above discussed method of lexical entry extraction has miserably failed in the extraction of the lexical entry in the taxonomic text. In the sub section **Heuristic Methods of identification of multiword keyword (lexical entry) by using hybrid method comprising of RAKE and W2V** of the section 4.2.5**,** the proposed methodology to deal with this matter is discussed.

4.1.2 Taxonomy Induction

Taxonomy induction is the core of any Ontology Learning task. The traditional Ontology Learning has used two kinds of techniques to induct the taxonomy. The top down hierarchical clustering and bottom up hierarchical clustering is being traditionally used to identify the taxonomy of a particular text.

- **Top down hierarchy**

Top down hierarchical clustering algorithms assumes that the whole data set is a single cluster. On the basis of the distance of the object, it splits the whole cluster into two clusters. This is an iterative process and the cluster is identified step by step.

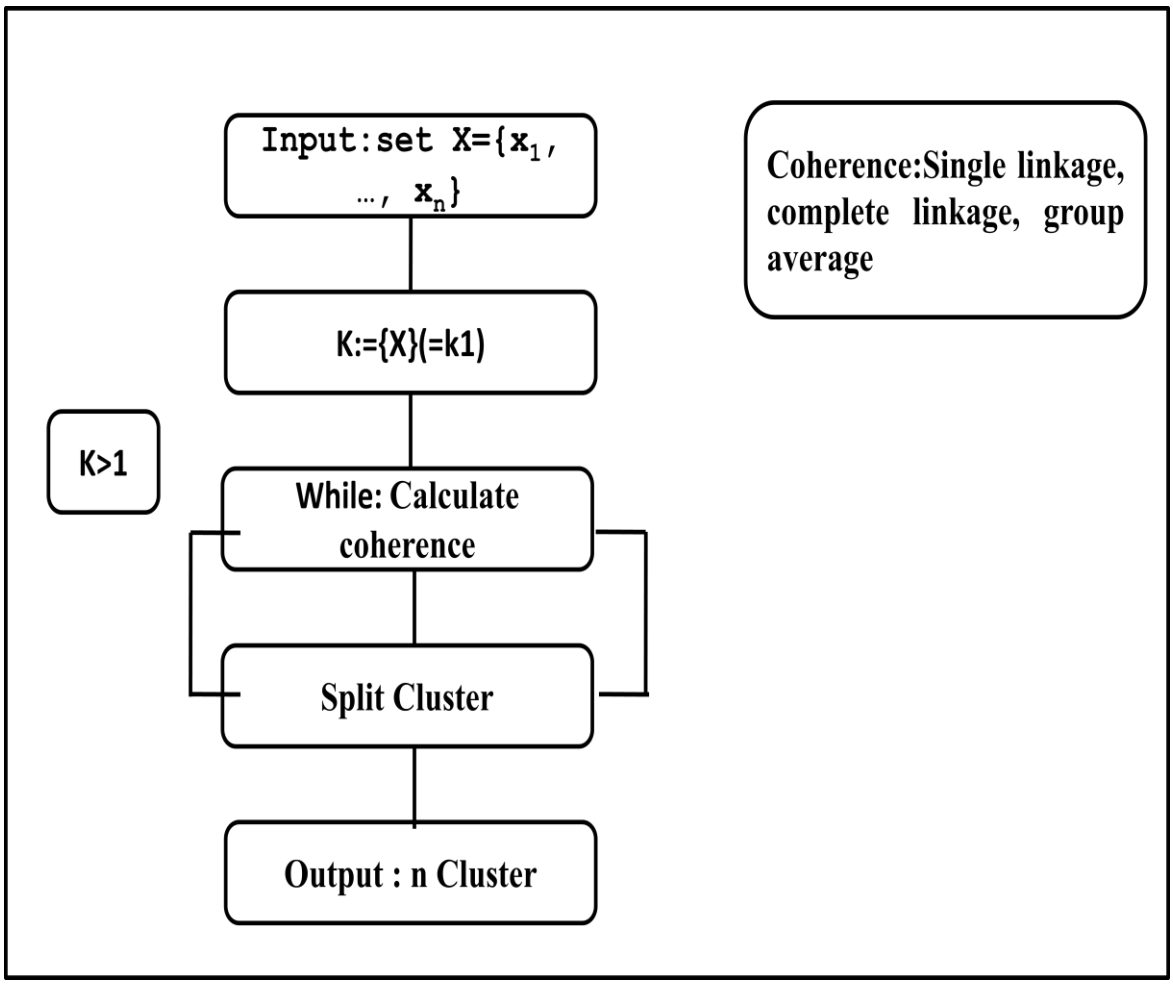


Figure 4.5 Algorithm of hierarchy induction of plain text by top down approach

- **Bottom up hierarchy**

The bottom up hierarchy follows the opposite principle of the top down approach. It assumes that each and every object is a cluster. The less distant clusters are merged to produce a single cluster.

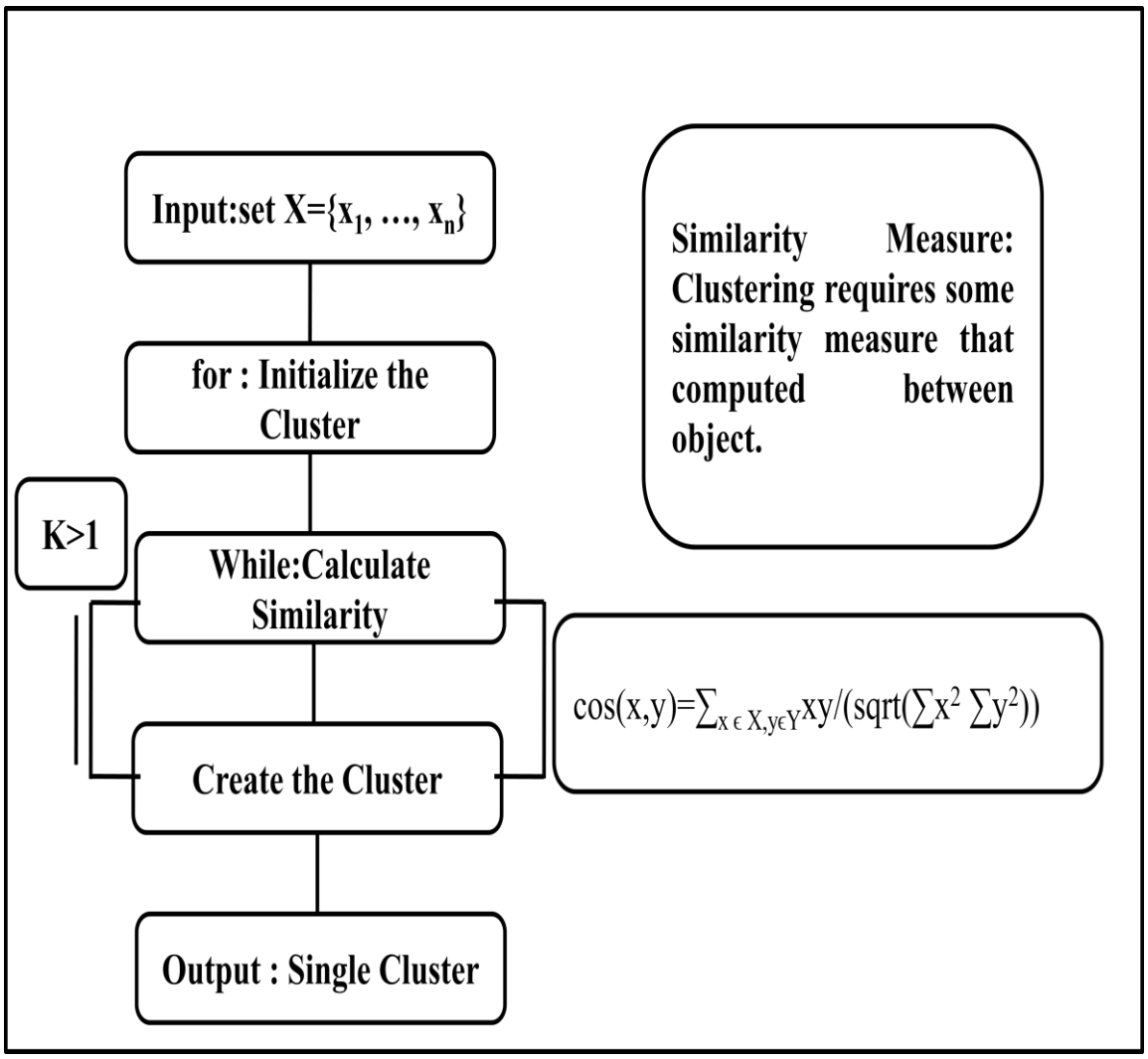


Figure 4.6 Algorithm of hierarchy induction of plain text by bottom up approach

4.1.3 Non Taxonomic Relation Extraction

The second important relationship of the Ontology Learning process is the non taxonomic relationship. The non taxonomic relationship describes the relations apart from the hierarchical relation. In Ontology, it can be said that non taxonomic relationship is the property of a class.

4.1.4 Association rule mining

To establish the relationship between taxonomic and non taxonomic text, this technique is used on the basis of support and confidence. The following figure depicts the algorithm which is studied and the use of the results of this methodology is described in the section 3.4.5.

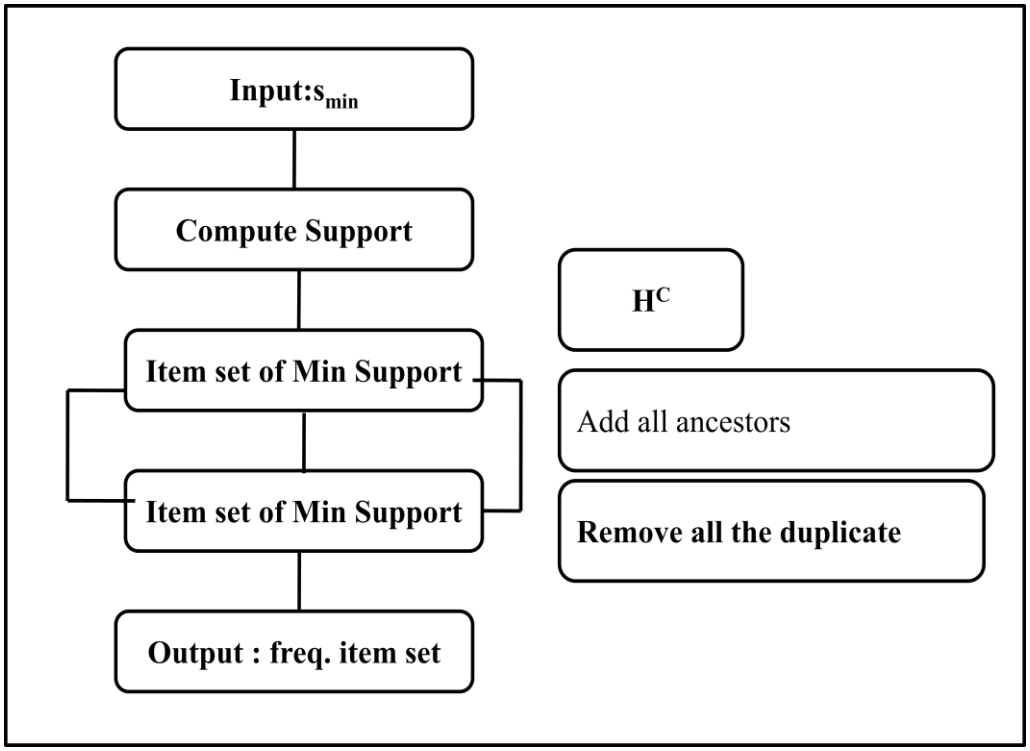


Figure 4.7 Extraction of the non taxonomic relationship by association rule mining (Figure reproduced from Alexander Maedche, 2001)

4.2 Enhancement of Ontology Learning approach

This objective extends the existing algorithms for taxonomic text and developed a working frame work for the Ontology Learning from the taxonomic text. It also includes the pre-processing task for natural language.

Table 4.2: Activity list under the objective of development of Ontology Learning algorithms from taxonomic text

- **Extend the algorithms for taxonomic text**
  - **Classify the Taxonomic and Non Taxonomic Text**
  - **Heuristic Methods of Keyword Extraction Using RAKE and W2V**
  - **Enhanced the Hearst Pattern for Taxonomy Extraction**
  - **Connective Based Taxonomic Tree Induction**
- **Develop a framework for Ontology Learning for the taxonomic text**
- **Natural Language Processing for extracting the concepts, instance, properties etc. from the taxonomic text**

4.2.1 Study of Taxonomic Text

A thorough study of the taxonomic text has been done. As a result of the study it was observed that the entity in the taxonomic text follows a particular pattern. For example, the USDA soil taxonomy used the “are the” connectives to describe the parent child relationship.

Table 4.3: Sentences present in the USDA soil taxonomy and the occurrence of the connectives to describe the parent child relationship

Sentence	Parent child Relationship
Aqualfs are the Alfisols	Aqualfs is a child of Alfisols
Other Vermaqualf, TypicVermaqualfs	Vermaqualf is the parent of TypicVermaqualfs

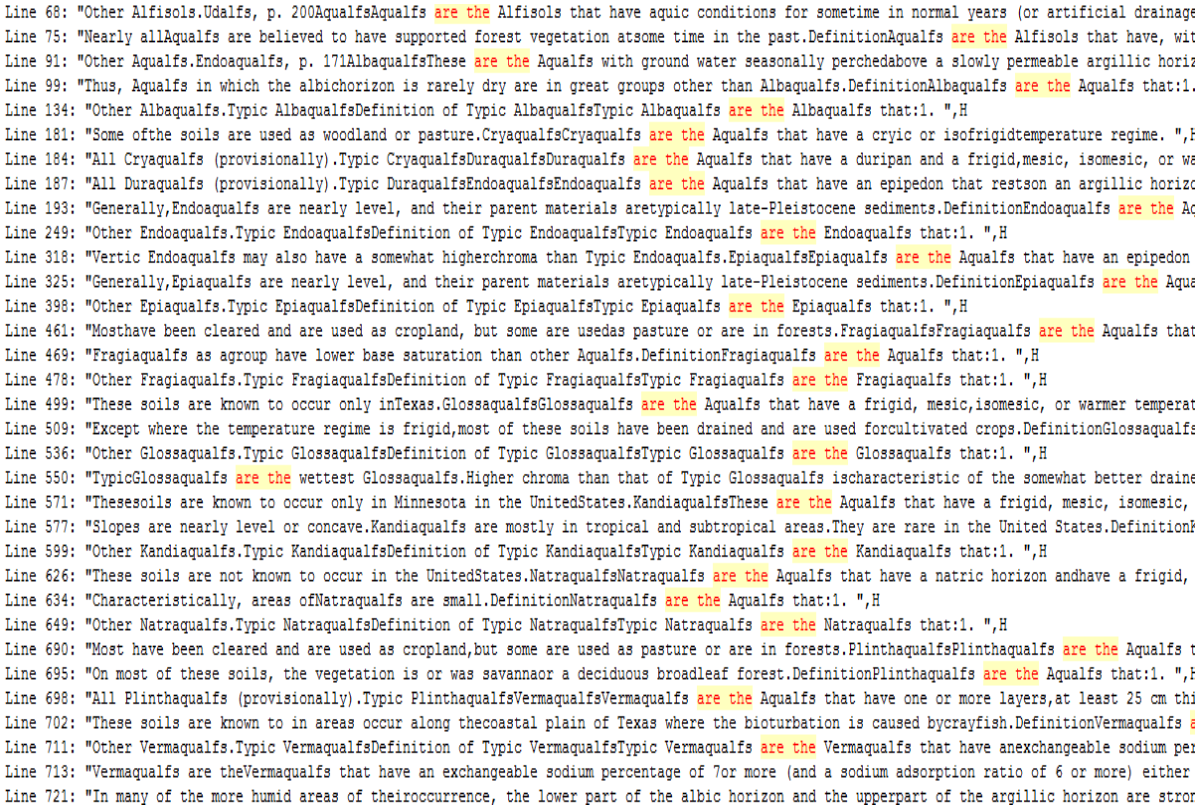


Figure 4.8 Snippet example of USDA soil taxonomy where the “are the” connectives are highlighted

This pattern can also be observed in microbial taxonomy (A.Parte, 2012)

Phylum XIII. *Firmicutes*
Class I. *"Bacilli"*
Order I. *Bacillales*
Family I. *Bacillaceae*
Genus I. *Bacillus*
Genus II. *Alkalibacillus*
Genus III. *Amphibacillus*
Genus IV. *Anoxybacillus*
Genus V. *Cerasibacillus*
Genus VI. *Filobacillus*
Genus VII. *Geobacillus*
Genus VIII. *Gracilibacillus*
Genus IX. *Halobacillus*
Genus X. *Halolactibacillus*
Genus XI. *Lentibacillus*
Genus XII. *Marinococcus*
Genus XIII. *Oceanobacillus*
Genus XIV. *Paraliobacillus*
Genus XV. *Pontibacillus*
Genus XVI. *Saccharococcus*
Genus XVII. *Tenuibacillus*
Genus XVIII. *Thalassobacillus*
Genus XIX. *Virgibacillus*
Family II. *"Alicyclobacillaceae"*
Genus I. *Alicyclobacillus*
Family III. *"Listeriaceae"*
Genus I. *Listeria*
Genus II. *Brochothrix*
Family IV. *"Paenibacillaceae"*
Genus I. *Paenibacillus*

Figure 4.9 Text snippet 1 from Microbial Taxonomy

Family *"Carnobacteriaceae"*

The members of the family "*Carnobacteriaceae*" are found in two paraphyletic clusters. *Carnobacterium* together with *Alkalibacterium*, *Allofustis*, *Alloiococcus*, *Atopococcus* (new; Collins et al., 2005), *Atopostipes*, *Desemzia*, *Dolosigranulum Isobaculum*, *Marinilactibacillus*, and *Trichococcus* represent the most comprehensive group. *Granulicatella* and *Atopobacter* (formerly in the "*Enterococcaceae*") are in the second group. However, the phylogenetic position of these genera remains ambiguous, and reassignment may be warranted as more information becomes available.

Family *"Enterococcaceae"*

Four genera remain within the family "*Enterococcaceae*": *Enterococcus*, *Melissococcus*, *Tetragenococcus*, and *Vagococcus*. *Atopobacter* was transferred to the "*Carnobacteriaceae*" (see above). The recently described genus *Catellicoccus*, which is not described in this volume, phylogenetically represents a sister group to the "*Enterococcaceae*".

Family *"Leuconostocaceae"*

No changes of the taxonomic organization are made for the "*Leuconostocaceae*", which unifies three phylogenetically related genera: *Leuconostoc*, *Oenococcus*, and *Weissella*.

Figure 4.10 Text snippet 2 from Microbial Taxonomy

After a thorough study, it is seen that the taxonomic texts are different in two ways. Firstly, they have the well-structured hierarchy description in terms of sentence and secondly, it has multilevel hierarchy.

4.2.2 Morphological Characteristics of the Taxonomic Text

Morphologically taxonomic text is quite different from the normal text. Taxonomic text follows a uniform indentation throughout the text and same morphological indentation is maintained for equivalent hierarchical class. From the Figure 4.9 and 4.10 it can be observed that Family I, II, III and IV has same indentation, in the same way the entire Genus has maintained the same convention - thus indicating a definite pattern.

In the chapter 3, **`Algorithm 3.5`** described the usage of recursive taxonomy induction from hierarchical text. This can be utilized to identify the existing pattern and use it for hierarchical relationship extraction from taxonomic text.

4.2.3 Key characteristic features of Taxonomic Text

a) The Taxonomic Text is relatively more structured than the normal text.
b) Taxonomic Texts are domain specific so the terminological density of that particular domain is higher than the normal text.
c) Normal text contains multiple hierarchies but taxonomic text mainly contains multilevel hierarchy.
d) Extraction of the parent child relationship is not uniform in the normal text but in the case of taxonomic text, it always exists.

4.2.4 Corpus Development of Taxonomic text

The Ontology Learning Process involves building of Ontology automatically from the text. In this case the Ontology building is done automatically from the taxonomic text. The taxonomic text is very crisp in terms of concept density, for that reason the statistical relationship extraction among the concepts is difficult. In order to increase the domain related terms and the easy extraction of the relationships among the objects; we have selected two sources for development of the Corpus. The primary source is the taxonomy book for a domain and the secondary source is the Wikipedia.

In chapter 3, **`Algorithm 3.1`**discusses the methodology for extracting the domain term on the basis of given keywords by the domain expert and tried to increase the size of the Corpus. Limitation of the iteration for 2 is there, because after the second iteration it would increase the non domain term and it is difficult to extract the exact relation from the text and development of the Ontology.

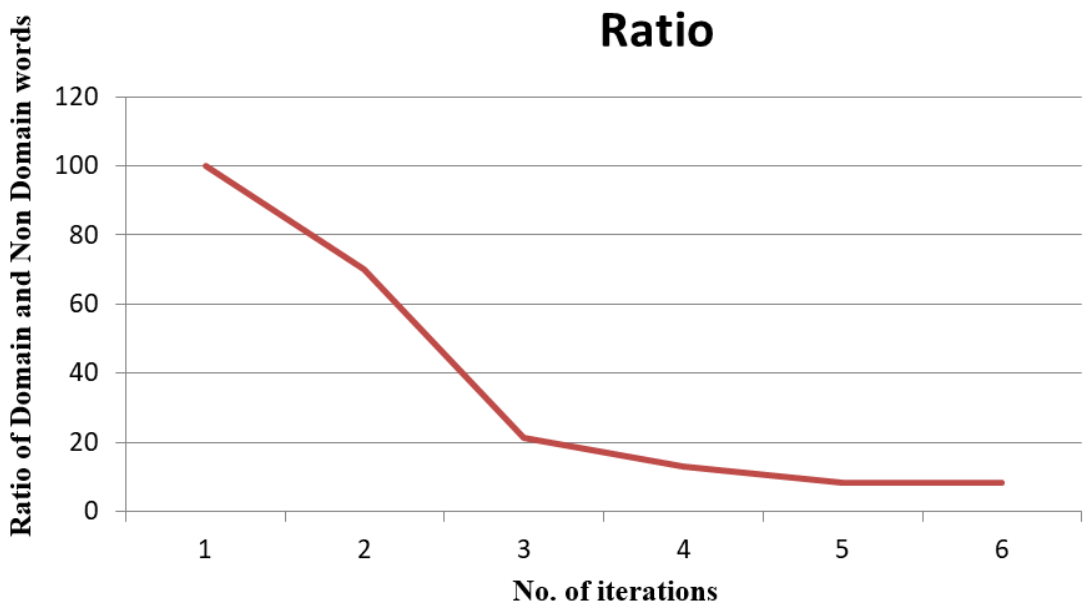


Figure 4.11 the graph showing the decreasing ratio of domain and non domain terms with increasing iterations

A graph (Figure 4.11) is plotted where x axis is the number of scrapping iterations in the Wikipedia and y axis is the ratio between the domain word and non domain word. The result shows that upon increasing the number of iterations of scraping decreases the relative frequency of domain term. Thus, it indicates that the number of iterations should be limited to 2 else there is a chance of degradation in the quality of the Corpus which can produce an inconsistency in the results expected.

Below is a table (Table 4.4), which shows the increase in the domain term upon increment of seed words from 2 to 5.

Table 4.4 Word count after the automated extraction of the domain term

Particulars	Taxonomic Text	Seed Word=2	Seed Word=3	Seed Word=4	Seed Word=5
Word Count	668957	676493	683246	703412	724355
Domain Term	75564	76223	79508	84698	89925

It can be seen that increment of seed word from 2 to 5 resulted in the increment of the domain term from 76223 to 89925. This resulted in the enhancement of the developed Corpus by increasing the domain term density.

In the module of Corpus development, the Ontology engineer or domain expert can give the seed word for Wikipedia scraping. The snapshot of the console result of Wikipedia scrapping is shown below:

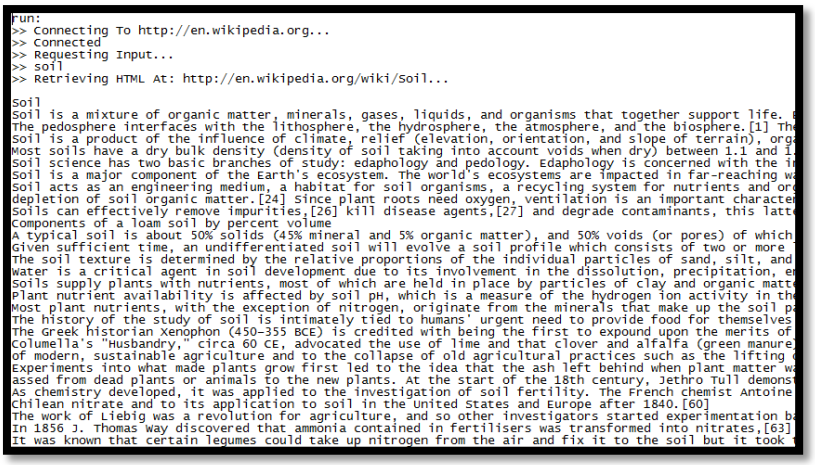


Figure 4.12 Results of scraping of Wikipedia on the basis of given keyword 'soil'

From the Figure 4.12, it can be seen that there is a successful scrapping of the content of the given domain specific seed word from Wikipedia and dumping into local system- resulting in the enhancement of the volume of the Corpus. It will ultimately benefit for NLP tasks of better extraction of concept of a particular or a given domain.

- **Output of the resulting Corpus**

Collection of the text of a particular domain is the prime aim of a Corpus development. In this research the Corpus development is based on the two source of domain text. Firstly, the chapters of the taxonomy book (**Key to Soil Taxonomy, 2009**) taken as an input in pdf format input for Corpus development. Secondly, the pdf format is converted into several other format i.e. text, .arff by using Python packages for easy machine readability.

Below is a snapshot of the developed Corpus

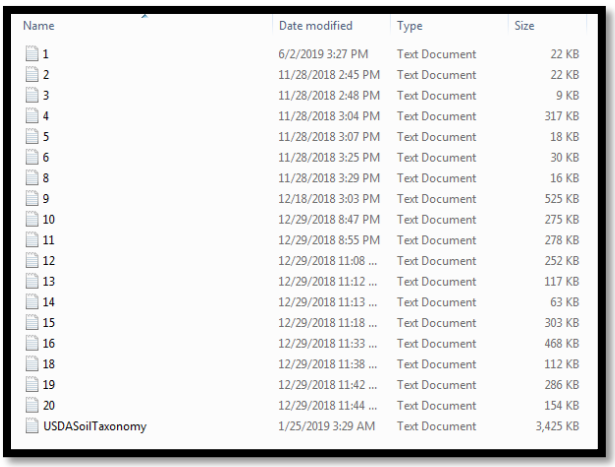


Figure 4.13 Developed Corpus in a standard text file format

The electronic copy of the developed Corpus is shown below.

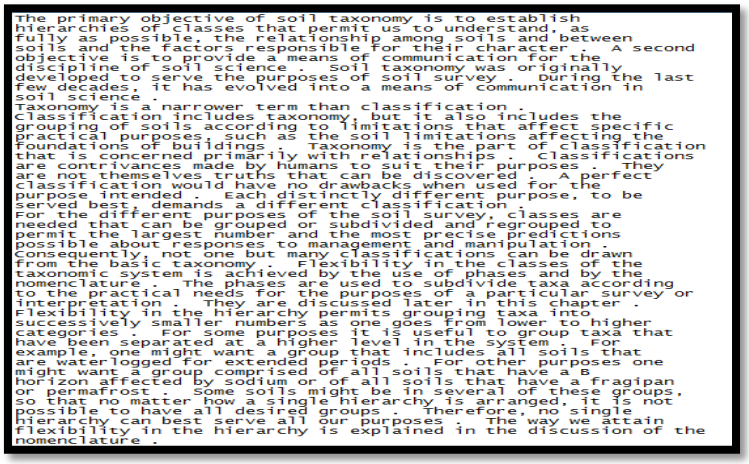

The primary objective of soil taxonomy is to establish
hierarchies of classes that permit us to understand, as
fully as possible, the relationship among soils and between
soils and the factors responsible for their character . A second
objective is to provide a means of communication for the
discipline of soil science . Soil taxonomy was originally
developed to serve the purposes of soil survey . During the last
few decades, it has evolved into a means of communication in
soil science .
Taxonomy is a narrower term than classification .
Classification includes taxonomy, but it also includes the
grouping of soils according to limitations that affect specific
practical purposes, such as the soil limitations affecting the
foundations of buildings . Taxonomy is the part of classification
that is concerned primarily with relationships . Classifications
are contrivances made by humans to suit their purposes . They
are not themselves truths that can be discovered . A perfect
classification would have no drawbacks when used for the
purpose intended . Each distinctly different purpose, to be
served best, demands a different classification .
For the different purposes of the soil survey, classes are
needed that can be grouped or subdivided and regrouped to
permit the largest number and the most precise predictions
possible about responses to management and manipulation .
Consequently, not one but many classifications can be drawn
from the basic taxonomy . Flexibility in the classes of the
taxonomic system is achieved by the use of phases and by the
nomenclature . The phases are used to subdivide taxa according
to the practical needs for the purposes of a particular survey or
interpretation . They are discussed later in this chapter .
Flexibility in the hierarchy permits grouping taxa into
successively smaller numbers as one goes from lower to higher
categories . For some purposes it is useful to group taxa that
have been separated at a higher level in the system . For
example, one might want a group that includes all soils that
are waterlogged for extended periods . For other purposes one
might want a group comprised of all soils that have a B
horizon affected by sodium or of all soils that have a fragipan
or permafrost . Some soils might be in several of these groups,
so that no matter how a single hierarchy is arranged, it is not
possible to have all desired groups . Therefore, no single
hierarchy can best serve all our purposes . The way we attain
flexibility in the hierarchy is explained in the discussion of the
nomenclature .

Figure 4.14 Text snippet of the electronic copy of the developed Corpus

- **Key observation from Corpus development**

a) The taxonomic text is a great source of domain knowledge but to impose natural language processing technique the Corpus size should be sufficiently large.

b) To make the Corpus size automatically large we have used the available API's.

Table 4.5 Enhancement of the Corpus specification of the packages

Packages	Functions used
`Wikipedia`	`summary()`
`RAKE`	`RAKE()` `extract_keywords_from_text()`
`nltk.tokenize`	`word_tokenize()` `sent_tokenize()`

The packages like Wikipedia, RAKE, nltk.tokenize are used for providing domain content, identifying keywords for further iteration process and helping both the packages Wikipedia and RAKE respectively.

The soil taxonomy Corpus is now well developed wherein the algorithms described below is applied for automated Ontology development.

4.2.5 Enhancement of the Ontology Learning algorithms for taxonomic text in the developed Corpus

Taxonomic text is used for imposing the enhanced algorithms. Firstly, the proposal of the initial segregation of the text is done. For segregating the text, several classification

techniques as mentioned in the sub section of **Classification Techniques used for segregation of** section 3.4.3 have been used. Then, a heuristic method for identification of the important multi word that has the potential to be a class in the Ontology is developed. Next, an enhancement of the Hearst pattern suitable for the taxonomic text is done as per the pattern developed by Hearst *et al*., (1992) and Seitner *et al.*, (2016).

- **Segregation of the taxonomic text**

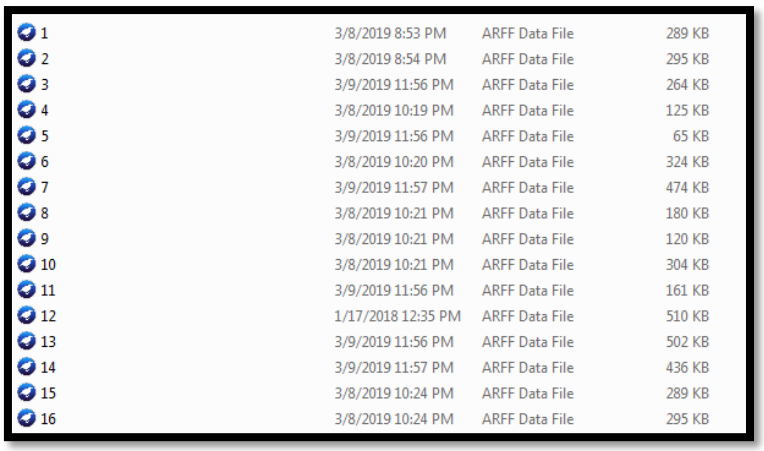
A consideration of the taxonomic text as mixer of two types of text - the hierarchical text and non hierarchical text is counted here. Hand crafting of the data into two classes- "H" for hierarchical i.e. core taxonomic and "NH" for non hierarchical i.e. non-taxonomic text is done on the basis of the availability of the core taxonomy (existence of parent-child relationship). It is shown below.

	text	label
0	@relation	Alfisols
1	@attribute sentence	string
2	@attribute	class{H,NH}
3		@data
4	"nullThe central concept of Alfisols is that o...	",NH
5	"Alfisols may alsohave a fragipan, a duripan, ...	",NH
6	"Where the soil temperature regime is mesic or...	",NH
7	"Inwarmer regions, they are on late-Pleistocen...	",NH
8	"Alfisols are not known tohave a perudic moist...	",NH
.		
3333	"The subgroup isprovided for use in other coun...	",NH
3334	"soils are like TypicRhodoxeralfs in defined p...	",H
3335	"They are not knownto occur in the United States.	",NH
3336	"The subgroup is provided for usein other coun...	",NH
3337	"soils are like TypicRhodoxeralfs, but they ha...	",H
3338	"They are not known to occur in theUnited States.	",NH
3339	"The subgroup is provided for use in othercoun...	",NH
3340	"soils are like TypicRhodoxeralfs, but they ha...	",H
3341	"Thesubgroup is provided for use in other coun...	",NH

Figure 4.15 USDA soil taxonomy training data for taxonomic Hierarchical ("H") and Non Hierarchical ("NH") text based on handcrafting

The results of the handcrafted training data will be helpful for further automated segregation of the text into Hierarchical and non-hierarchical data in the whole Corpus.

The handcrafted NH and H classified data set is converted into .arff file format for easy machine readability and usage in the application of further classification techniques. It is shown below.



1	3/8/2019 8:53 PM	ARFF Data File	289 KB
2	3/8/2019 8:54 PM	ARFF Data File	295 KB
3	3/9/2019 11:56 PM	ARFF Data File	264 KB
4	3/8/2019 10:19 PM	ARFF Data File	125 KB
5	3/9/2019 11:56 PM	ARFF Data File	65 KB
6	3/8/2019 10:20 PM	ARFF Data File	324 KB
7	3/9/2019 11:57 PM	ARFF Data File	474 KB
8	3/8/2019 10:21 PM	ARFF Data File	180 KB
9	3/8/2019 10:21 PM	ARFF Data File	120 KB
10	3/8/2019 10:21 PM	ARFF Data File	304 KB
11	3/9/2019 11:56 PM	ARFF Data File	161 KB
12	1/17/2018 12:35 PM	ARFF Data File	510 KB
13	3/9/2019 11:56 PM	ARFF Data File	502 KB
14	3/9/2019 11:57 PM	ARFF Data File	436 KB
15	3/8/2019 10:24 PM	ARFF Data File	289 KB
16	3/8/2019 10:24 PM	ARFF Data File	295 KB

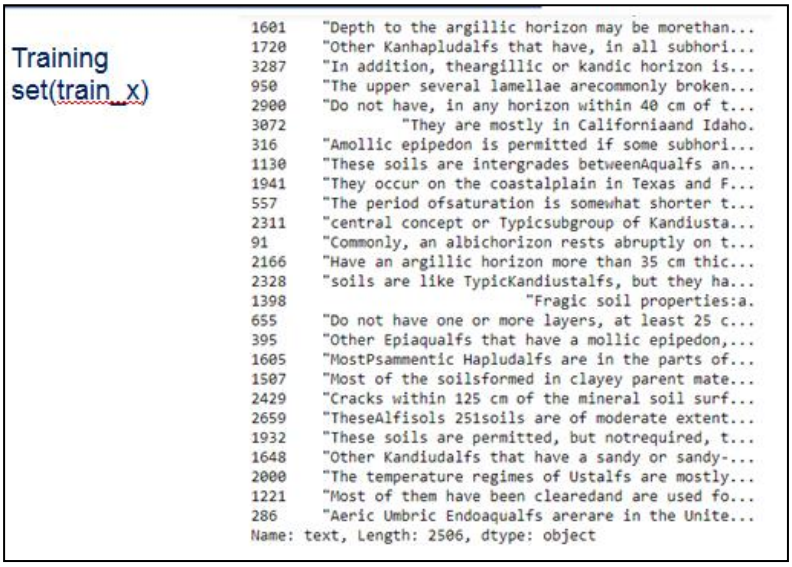
Figure 4.16 Pre processed .arff text data file of taxonomic text

Experimental Setup for classification - the Taxonomic and Non Taxonomic Text

Encoding Techniques: Encoding techniques are the essential techniques for making the text data suitable for machine learning. All the theoretical aspects of the encoding techniques are described in the **Feature Engineering** sub section of the section 3.4.2. The count vector, TFIDF and W2V technique has been used for encoding. For TFIDF, usage of the three variants of TFIDF i.e. the word level, n-gram and character level is there.

Input: In this experiment, the input is the taxonomic text

Below is the snapshot of the training dataset used for classification before conversion into encoded format.

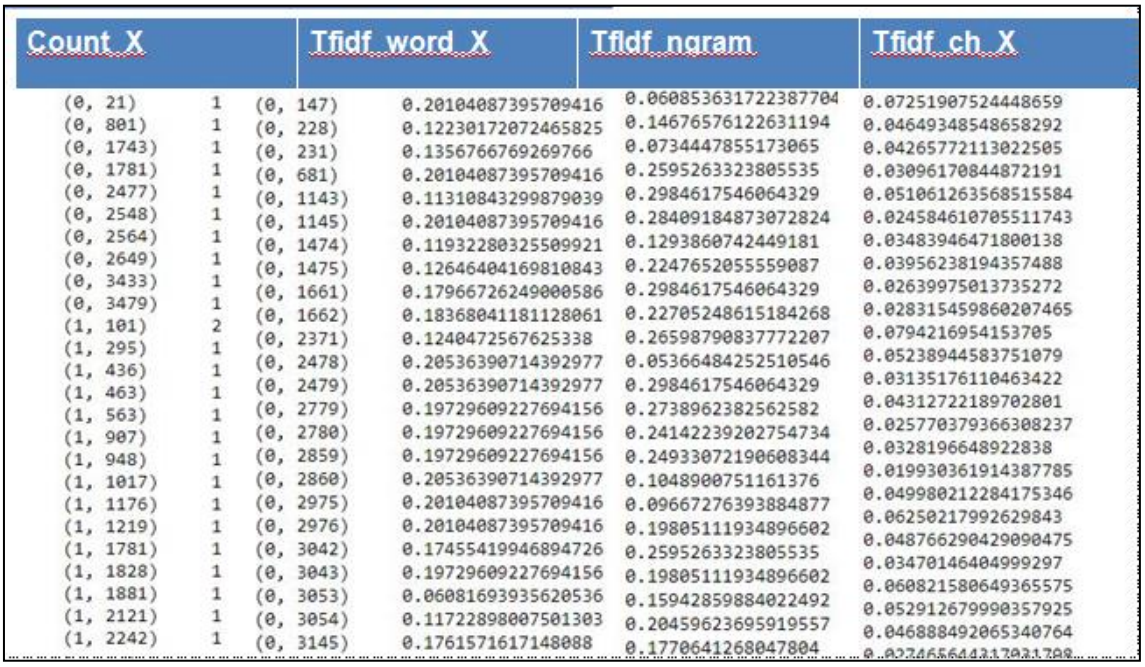


Training set(train_x)

```
1601    "Depth to the argillic horizon may be morethan...
1720    "Other Kanhapludalfs that have, in all subhori...
3287    "In addition, theargillic or kandic horizon is...
950     "The upper several lamellae arecommonly broken...
2900    "Do not have, in any horizon within 40 cm of t...
3072              "They are mostly in Californiaand Idaho.
316     "Amollic epipedon is permitted if some subhori...
1130    "These soils are intergrades betweenAqualfs an...
1941    "They occur on the coastalplain in Texas and F...
557     "The period ofsaturation is somewhat shorter t...
2311    "central concept or Typicsubgroup of Kandiusta...
91      "Commonly, an albichorizon rests abruptly on t...
2166    "Have an argillic horizon more than 35 cm thic...
2328    "soils are like TypicKandiustalfs, but they ha...
1398                          "Fragic soil properties:a.
655     "Do not have one or more layers, at least 25 c...
395     "Other Epiaqualfs that have a mollic epipedon,...
1605    "MostPsammentic Hapludalfs are in the parts of...
1507    "Most of the soilsformed in clayey parent mate...
2429    "Cracks within 125 cm of the mineral soil surf...
2659    "TheseAlfisols 251soils are of moderate extent...
1932    "These soils are permitted, but notrequired, t...
1648    "Other Kandiudalfs that have a sandy or sandy-...
2000    "The temperature regimes of Ustalfs are mostly...
1221    "Most of them have been clearedand are used fo...
286     "Aeric Umbric Endoaqualfs arerare in the Unite...
Name: text, Length: 2506, dtype: object
```

Figure 4.15 Snapshot of the training data for use in classification before encoding

The encoded format is shown in the figures (4.16 and 4.17) below:



Count_X		Tfidf_word_X		Tfidf_ngram	Tfidf_ch_X
(0, 21)	1	(0, 147)	0.20104087395709416	0.060853631722387704	0.07251907524448659
(0, 801)	1	(0, 228)	0.12230172072465825	0.14676576122631194	0.04649348548658292
(0, 1743)	1	(0, 231)	0.1356766769269766	0.0734447855173065	0.04265772113022505
(0, 1781)	1	(0, 681)	0.20104087395709416	0.2595263323805535	0.03096170844872191
(0, 2477)	1	(0, 1143)	0.11310843299879039	0.2984617546064329	0.051061263568515584
(0, 2548)	1	(0, 1145)	0.20104087395709416	0.28409184873072824	0.024584610705511743
(0, 2564)	1	(0, 1474)	0.11932280325509921	0.1293860742449181	0.03483946471800138
(0, 2649)	1	(0, 1475)	0.12646404169810843	0.2247652055559087	0.03956238194357488
(0, 3433)	1	(0, 1661)	0.17966726249000586	0.2984617546064329	0.02639975013735272
(0, 3479)	1	(0, 1662)	0.18368041181128061	0.22705248615184268	0.028315459860207465
(1, 101)	2	(0, 2371)	0.1240472567625338	0.26598790837772207	0.0794216954153705
(1, 295)	1	(0, 2478)	0.20536390714392977	0.05366484252510546	0.05238944583751079
(1, 436)	1	(0, 2479)	0.20536390714392977	0.2984617546064329	0.03135176110463422
(1, 463)	1	(0, 2779)	0.19729609227694156	0.2738962382562582	0.04312722189702801
(1, 563)	1	(0, 2780)	0.19729609227694156	0.24142239202754734	0.025770379366308237
(1, 907)	1	(0, 2859)	0.19729609227694156	0.24933072190608344	0.0328196648922838
(1, 948)	1	(0, 2860)	0.20536390714392977	0.1048900751161376	0.019930361914387785
(1, 1017)	1	(0, 2975)	0.20104087395709416	0.09667276393884877	0.049980212284175346
(1, 1176)	1	(0, 2976)	0.20104087395709416	0.19805111934896602	0.06250217992629843
(1, 1219)	1	(0, 3042)	0.17455419946894726	0.2595263323805535	0.048766290429090475
(1, 1781)	1	(0, 3043)	0.19729609227694156	0.19805111934896602	0.03470146404999297
(1, 1828)	1	(0, 3053)	0.06081693935620536	0.15942859884022492	0.060821580649365575
(1, 1881)	1	(0, 3054)	0.11722898007501303	0.20459623695919557	0.052916279990357925
(1, 2121)	1	(0, 3145)	0.1761571617148088	0.1770641268047804	0.046888492065340764
(1, 2242)	1				0.027465644317031708

Figure 4.16 Snapshot of the encoded data for Count Vector, TFIDF (Word level, n gram and character level)

```
[ 8.550e-02  5.310e-02  4.000e-03  4.440e-02 -8.110e-02  3.430e-02
  6.300e-02 -9.280e-02 -7.520e-02  6.250e-02  8.760e-02 -1.470e-02
 -2.510e-02 -7.780e-02  1.078e-01  9.000e-04  1.820e-02  5.790e-02
  1.519e-01 -5.310e-02 -3.100e-03 -4.020e-02 -7.410e-02 -4.970e-02
  1.320e-02 -2.130e-02 -5.020e-02 -2.410e-02  4.090e-02 -6.100e-02
  3.730e-02  5.700e-03 -1.400e-02  1.890e-02  2.960e-02 -3.100e-02
  1.302e-01 -1.294e-01 -4.280e-02 -1.777e-01  2.550e-02  9.100e-02
 -1.279e-01  9.200e-02  1.151e-01  4.250e-02  2.730e-02  7.000e-03
  2.410e-02  2.300e-02 -7.310e-02 -8.000e-02 -6.862e-01 -3.400e-02
 -1.530e-02 -1.121e-01 -3.000e-03  5.590e-02  1.950e-02  4.520e-02
 -2.290e-02  2.780e-02 -1.860e-02 -3.000e-04  7.830e-02 -7.850e-02
 -5.640e-02  3.170e-02 -5.510e-02  8.780e-02 -3.440e-02  3.000e-02
  7.800e-02 -3.960e-02 -5.280e-02 -5.270e-02  9.380e-02  1.760e-02
 -7.270e-02 -4.690e-02  2.590e-02  3.400e-02  1.040e-02 -1.580e-02
  3.550e-02  7.810e-02 -3.920e-02 -2.600e-03 -8.900e-03 -7.240e-02
  4.550e-02 -4.180e-02 -3.310e-02 -8.800e-02 -1.074e-01  7.800e-03
  6.330e-02  1.600e-03  1.690e-02  5.060e-02  3.850e-02  5.200e-03
 -5.540e-02  1.129e-01 -9.200e-03  1.120e-01 -4.010e-02 -4.980e-02
 -7.390e-02 -1.061e-01  2.100e-03  4.920e-02 -1.180e-02 -6.150e-02
  3.970e-02  3.230e-02 -3.560e-02  3.360e-02 -2.210e-02  3.740e-02
```

Figure 4.17 Snapshot of the encoded data for W2V describing a matrix representation of a single word

The following tables give the details of the parameters used in encoding API's.

Table 4.6: Parameters used in the `CountVectorizer` API

Parameters	Value
analyzer	word
binary	False
decode_error	strict
lowercase	True
max_features	None
min_df	1.0
ngram_range	(1, 1)
preprocessor	None
strip_accents	None
token_pattern	None
tokenizer	\\w{1,}
vocabulary	None

Table 4.7: Parameters used in the `TfidfVectorizer`API for word level

Parameters	Value
analyzer	word
token_pattern	r'\w{1,}'
max_features	5000

Table 4.8: Parameters used in the `TfidfVectorizer`API for character level

Parameters	Value
analyzer	char
token_pattern	r'\w{1,}'
ngram_range	(2,3)
max_features	5000

Table 4.9: Parameters used in the `TfidfVectorizer`API for ngram

Parameters	Value
`analyzer`	char
`token_pattern`	r'\w{1,}'
`ngram_range`	(2,3)
`max_features`	5000

The overall classification module starting from input of the taxonomic text to encoding next training thereafter testing it and further the use of the developed model for text segregation is depicted below:

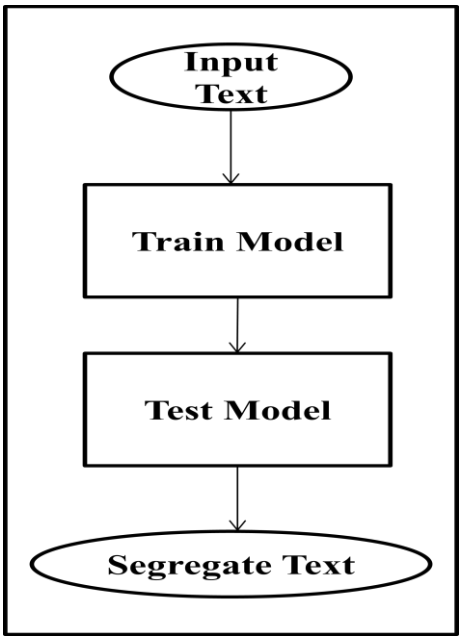


Figure 4.18 Flow chart of the process of classification and segregation of taxonomic text into hierarchical and non hierarchical categories

- **Comparison of the encoding and classification techniques used for segregating the taxonomic text**

The classification techniques which are used in the study are described in the sub section **Classification Techniques used for segregation** of the section 3.4.3. Similarly the encoding techniques used for converting the textual data into a numeric one or Machine learning framework readable format is discussed in the **Feature Engineering** sub section of the section 3.4.2.

A graph is plotted (Figure 4.19) to compare the encoding and classification techniques used for segregation of the taxonomic text for identifying the best possible combination of both the classification and encoding techniques.

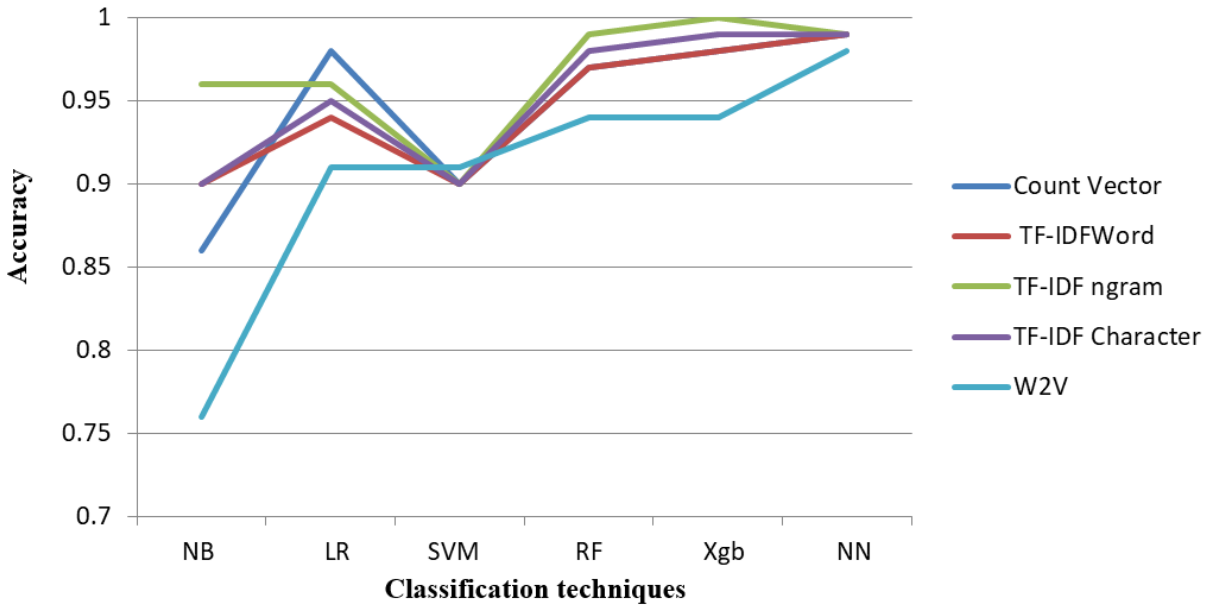


Figure 4.19 Graphical representations of the comparisons among different classification techniques on taxonomic text where Y-axis depicts the accuracy and X-axis depicts the different classification techniques used

It can be seen from the graph and by using factorial CRD on accuracy data set that- there is no significant interaction between the encoding techniques and the classification techniques used at 5% level of significance. So, it may be concluded that- the selection of the encoding and classification techniques do not influence the segregation process of separating the taxonomic part from the non taxonomic one.

The result depicting before and after the segregation process of the text into hierarchical and non hierarchical category is shown in the snapshot below:

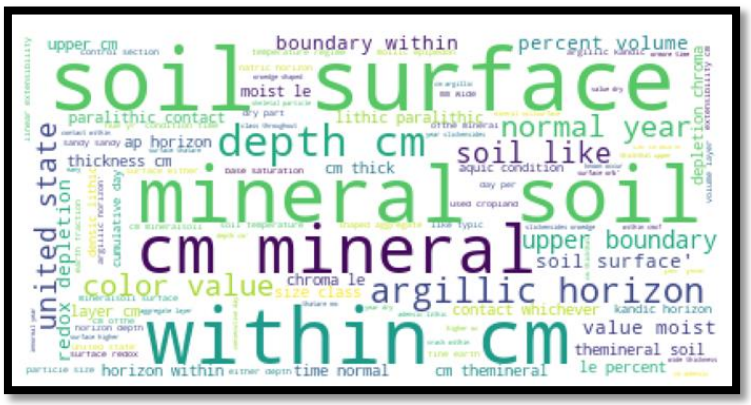


Figure 4.20 Snapshot of the Word Cloud before segregation of the text into hierarchical and non hierarchical category

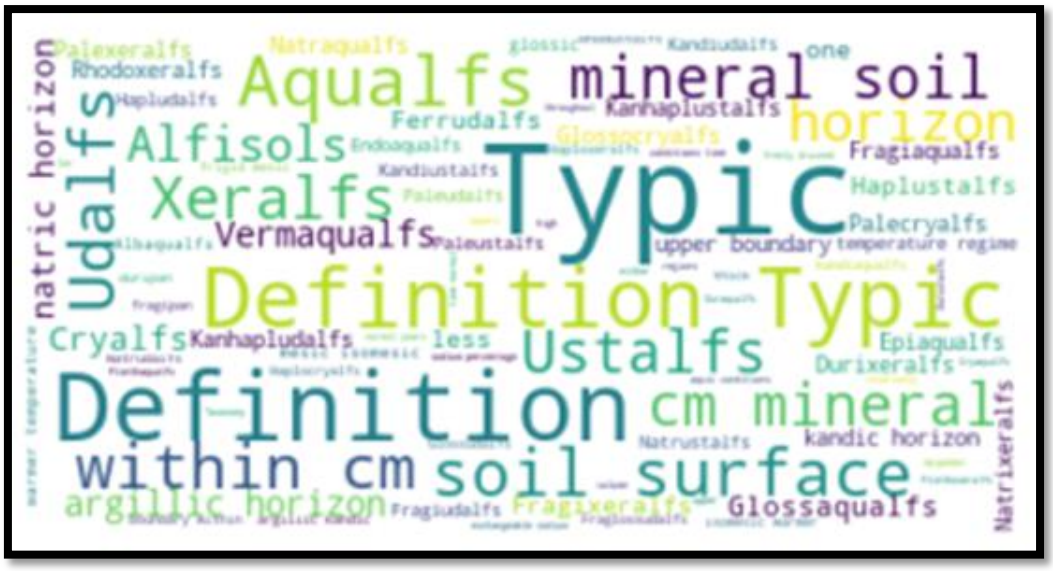


Figure 4.21 Snapshot of the Word Cloud after the segregation of the text into hierarchical and non hierarchical category

The above snapshot (Figure 4.20) depicted that the density of the taxonomic text is lesser than the generalized or non taxonomic texts. In a similar way , the snapshot (Figure 4.21) depicted just the reverse i.e. increase in the density of the taxonomic text than the non taxonomic one- this is the result of the segregation process which thus overcomes the problem of low density but important key words helping in the process of easy identification of the class in Ontology Learning.

- **Heuristic Methods of identification of multiword keyword (lexical entry) by using hybrid method comprising of RAKE and W2V**

A hybrid method of RAKE and W2V is developed and used for multiword keyword extraction from taxonomic text. It will be helpful for identifying the components of the Ontology (class, property and individuals) thus a way towards automated Ontology learning.

Two snapshots (Figure 4.22a, Figure 4.22b) below depicted the results of keyword extraction using RAKE.

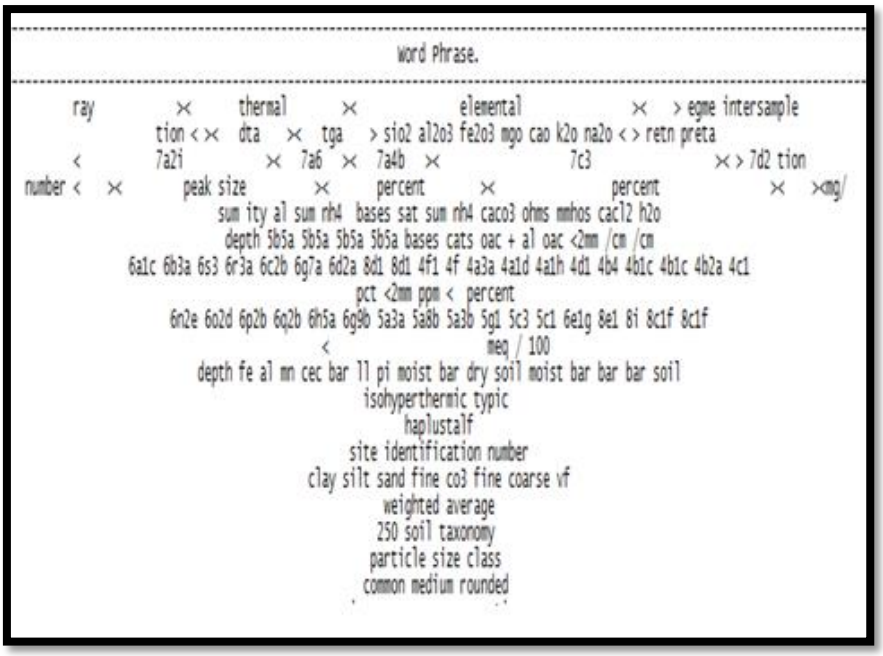


Figure 4.22 a Snapshot of Keyword extraction from taxonomic text using RAKE

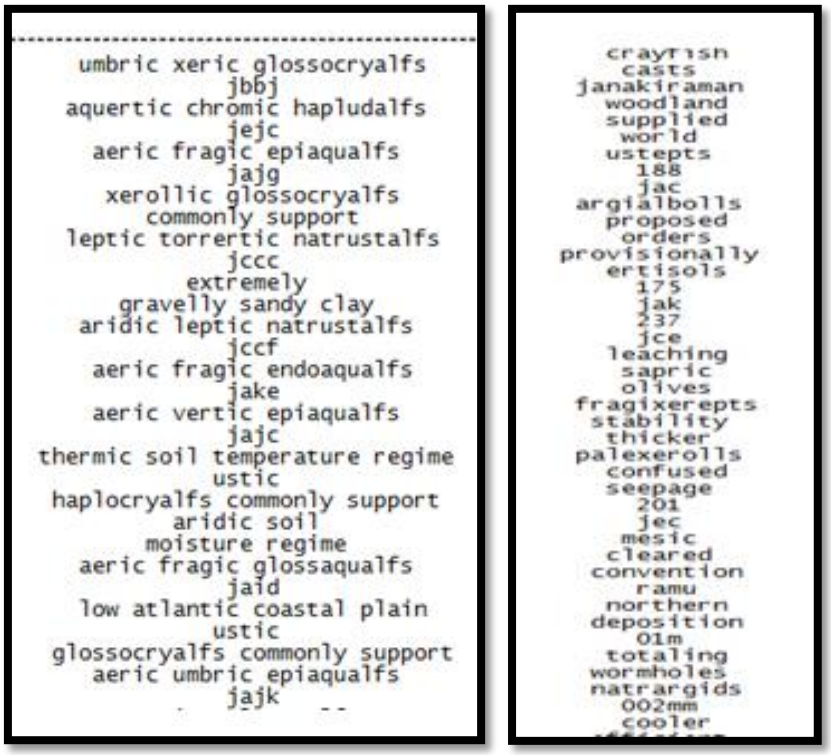


Figure 4.22b Snapshot of Keyword extraction from taxonomic text using RAKE

The result shows that the extractions of the keywords are not consistent and is not identifying domain related multiword keywords. For taxonomic text the multiword keywords are important but all the multiword extracted by RAKE are not be useful for the taxonomic domain.

The result thus shows that RAKE is an efficient tool for multiword keyword extraction, but it did not work well in case of taxonomic text. RAKE did not work well

because for a particular taxonomic text, the Corpus size is not very large; which resulted in the less frequency of the multiword keywords.

After seeing the failure of the RAKE technique alone in identification of multiword keyword, a hybrid method by taking W2V along with RAKE is developed. It simultaneously harnesses the encoding potential inherent in the W2V, and is also capable of capturing of the multiword keyword mediated by RAKE, thus ultimately a meaningful multiword keyword extraction can be done for identifying components of a particular domain of Ontology.

The snapshot of the hybrid method of RAKE and W2V is shown:

fragiaquic paleudults	True
andic endoaquods	True
russian school led.	False
sodic humicryerts	True
acrustoxic kandiustults	True
hartland town line.	False
kandiustalfic eutrustox	True
halic haploxererts	True
monthly air temperature.	False
humic fragiaquepts	True
leptic salitorrerts	True
fluvaquentic haploxerolls	True
torripsammentic haploxerolls	True
duridic haploxerolls	False
ieft.	False
torrertic haploxerolls	True
iefd.	False
drained	False
permeability class.	False
vertic dystrudepts	True

Figure 4.23 Snapshot of the multiword keyword extraction by using the hybrid methods of RAKE and W2V

W2V acted as a guide to the RAKE. Figure 4.23 depicted the results of meaningful multiword keyword extraction from the taxonomic text. It shows the improvement in the extraction of the domain term.

- **Extraction of hierarchical relationship using enhanced Hearst Pattern from Taxonomic text**

Hearst pattern is one of the popular techniques that have been used in several experiments to identify the parent child relationship, which is a characteristic of taxonomic text. Here an enhanced the Hearst pattern is used, according to the Hearst *et al*., (1992) and Seitner *et al.*, (2016).

The results of the usage of the enhanced Hearst pattern and extraction of parent child relationship are showed in the Figure 4.24 and 4.25 respectively. It thus helps in automated removal of the irrelevant terms in a hierarchical taxonomic relationship – thus further strengthening the automation of Ontology Learning.

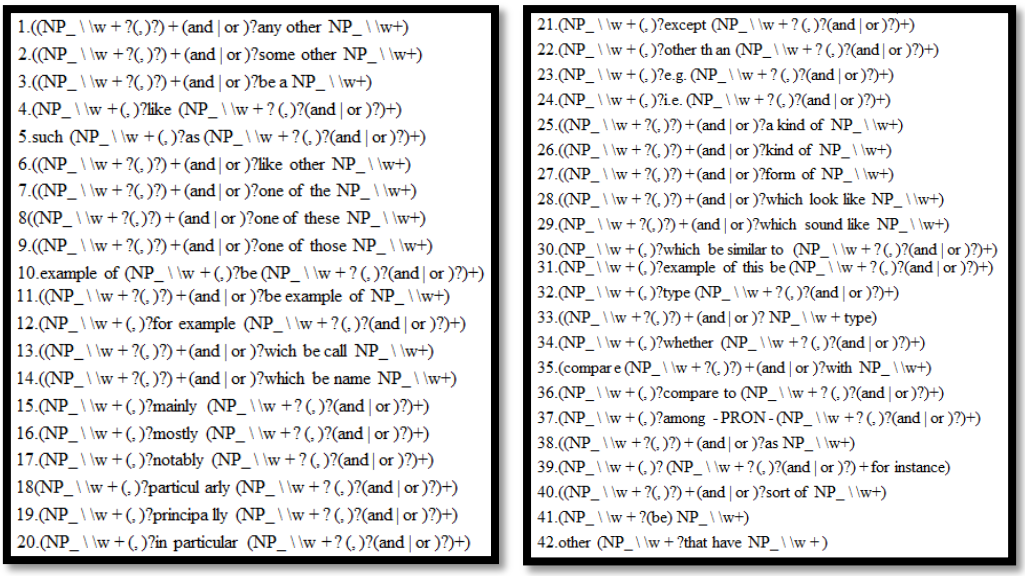


Figure 4.24 List of the Enhanced Hearst Patterns to extract the hierarchical relationship from taxonomic text

Aqualfs Aqualfs are the Alfisols that have aquic conditions for some time in normal years (or artificial drainage) at or near the soil surface .	['NP_aqualfs', 'NP_the_alfisols'] [('aqualfs', 'the alfisols')]
Albaqualfs These are the Aqualfs with ground water seasonally perched above a slowly permeable argillic horizon .	['NP_albaqualfs', 'NP_the_aqualfs'] [('albaqualfs', 'the aqualfs')]
Typic Albaqualfs Definition of Typic Albaqualfs Typic Albaqualfs are the Albaqualfs	['NP_typic_albaqualfs', 'NP_the_albaqualfs'] [('typic albaqualfs', 'the albaqualfs')]
Endoaqualfs are the Aqualfs	['NP_endoaqualfs', 'NP_the_aqualfs'] [('endoaqualfs', 'the aqualfs')]
Typic Endoaqualfs Definition of Typic Endoaqualfs Typic Endoaqualfs are the Endoaqualfs	['NP_typic_endoaqualfs', 'NP_the_endoaqualfs'] [('typic endoaqualfs', 'the endoaqualfs')]

Figure 4.25 Snapshot of the extraction of the semantic relationship i.e. parent child relationship which has been extracted from the text

- **Connective Based Taxonomic Tree Induction for extracting hierarchical relationship from Taxonomic text**

Hierarchical relationship extraction through enhanced Hearst Pattern is a much generalized way of extraction. Connective based taxonomic tree extraction method is more specific if knowledge on the pattern of the relationships existing among the entities in a domain text is known. A connective is a word or a collection of words that establishes the relationship between the classes identified in the taxonomic text. The connectives preserve the pattern throughout the text. A connective varies from taxonomy to taxonomy. For this experimental set up, USDA soil taxonomy have been used. Some of the connectives are

used for describing the classes in the ontology which are in same hierarchy and can be called as equality connectives (EC), while the others are used to identify the parent child relationships in the ontology called as Hierarchical connective (HC).

The snapshot of the hierarchical relationship extraction by HC and EC in individuality and in using both are shown below:

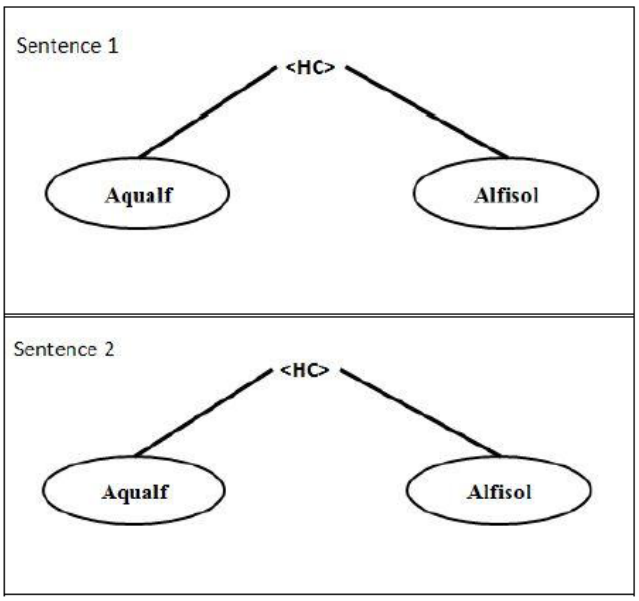


Figure 4.26 Snapshot of Hierarchical relationship extraction - Hierarchical Connectives<HC>: "are the" from input sentence 1 and input sentence 2 of the taxonomic text (Figure reproduced from Deb *et al.*, 2018).

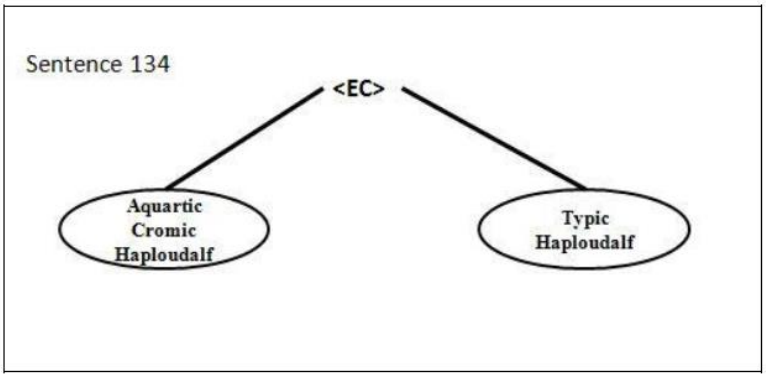


Figure 4.27 Snapshot of Hierarchical relationship extraction Equality <EC>: "other" from input sentence 1 and input sentence 2 of the taxonomic text (Figure reproduced from Deb *et al.*, 2018).

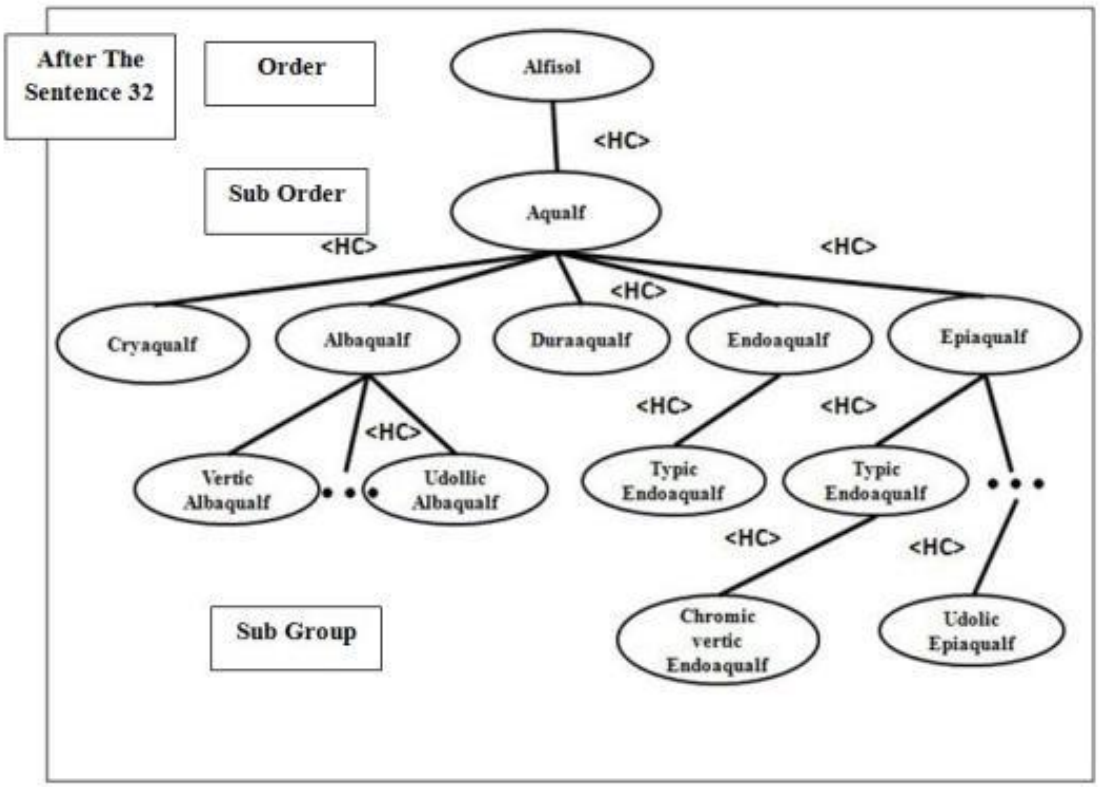


Figure 4.28 Snapshot of Connective based hierarchy extraction using both HC and EC from taxonomic text (Figure reproduced from Deb *et al.*, 2018).

Figure 4.26, 4.27 and 4.28 describes an iterative process of inducting the hierarchy from the taxonomic text using HC and EC.

The result shows that HC and EC connectives are able to identify all the available hierarchical relationships present in the soil taxonomy from order to sub group level.

- **Development of a framework for Ontology Learning from the taxonomic text**

In this research one framework has been developed for learning Ontology from the taxonomic text. The developed framework acted as a guide in the Ontology learning process.

According to the developed framework, the taxonomic text is first segregated into two different classes - the taxonomic text and the non taxonomic text. After the segregation into the respective categories, the taxonomic text is used for generating the taxonomic hierarchy (parent child relationship) whereas; the later text is used for generation of non taxonomic hierarchy. The Framework has led to the development of algorithm library which has ultimately led to its implementations for Ontology Learning.

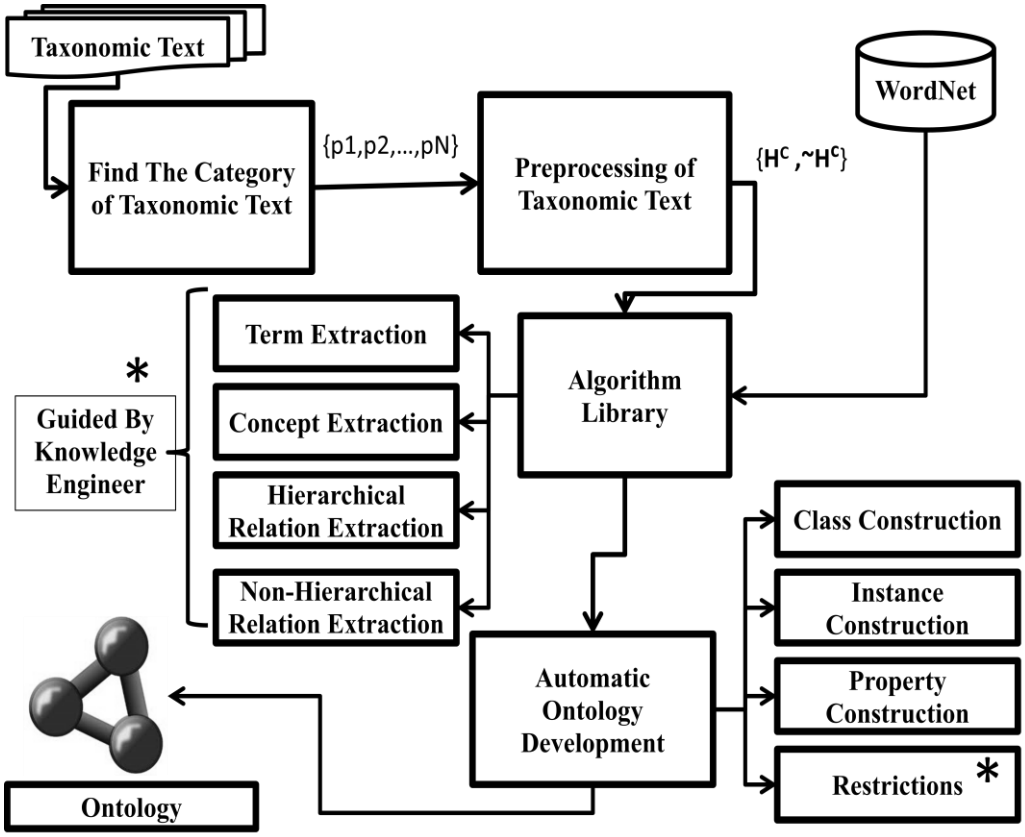


Figure 4.29 Developed Framework for Ontology Learning from Taxonomic Text (Figure reproduced from Deb *et al.*, 2018)

From the Figure 4.29, it can be seen that a particular category of taxonomic text is taken as the input and it is preprocessed (sentence detection, tokenization etc.). It is then used for making algorithm library ultimately leading to automated Ontology development.

- **Natural Language Processing Unit**

Specialized natural language i.e. taxonomic text is used for the purpose of natural language processing. Almost all the module of this research involved natural language processing task. Figure 4.30 and Figure 4.31 depicted the ER Diagram and Class Diagram of the NLP unit respectively.

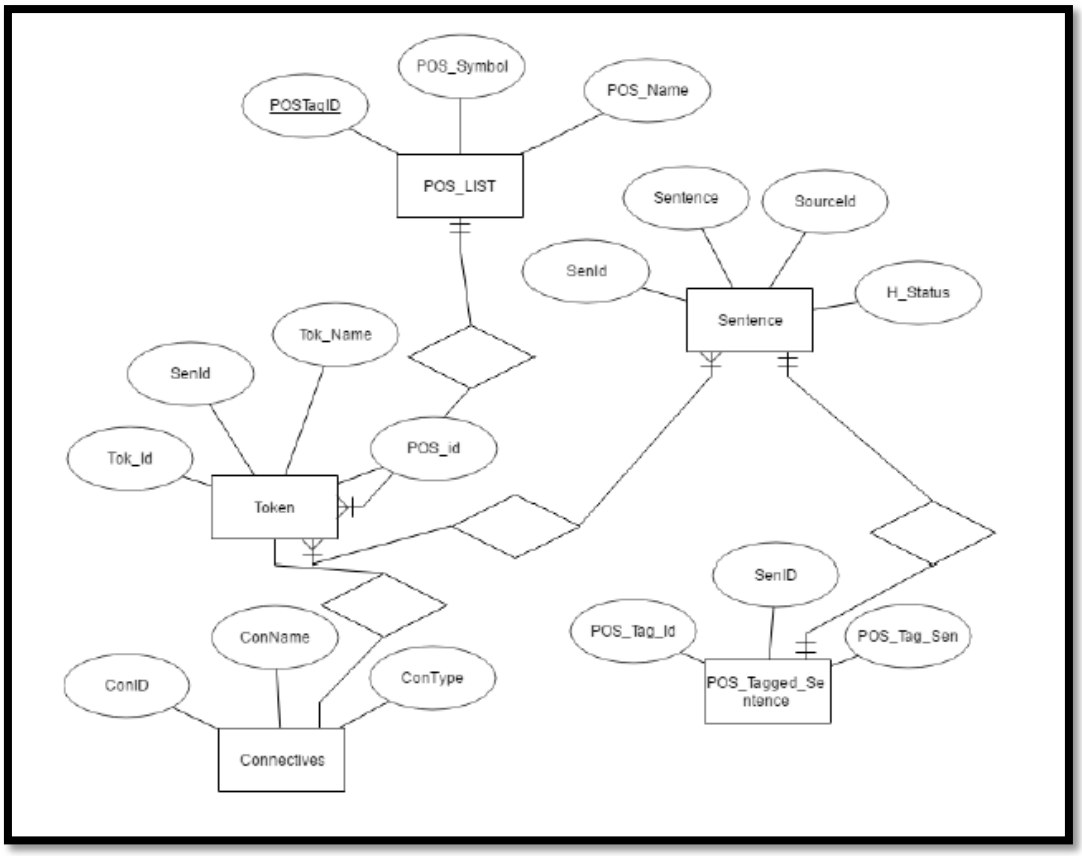


Figure 4.30 Snapshot of an ER diagram for a NLP Module

From the above (Figure 4.30), it can be seen that there are 4 entities (POS_LIST, Sentence, Token, Connectives and POS_Tagged_Sentence). These entities are interconnected and able to store extracted text from Corpus. It is also used to store the pre-processed data temporarily and supply in need.

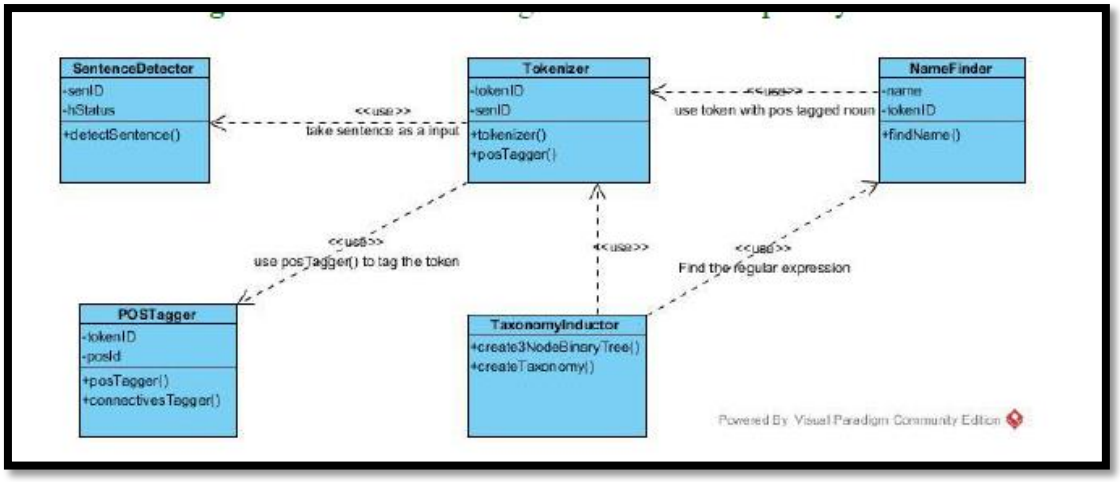


Figure 4.31 Snapshot of a Class diagram for a NLP Module

From the above (Figure 4.31), it can be seen that the classes of this class diagram includes- Sentence Detector, Tokenizer, Name Finder, POSTagger, TaxonomyInductor. All the classes are interconnected for proper working of the NLP module.

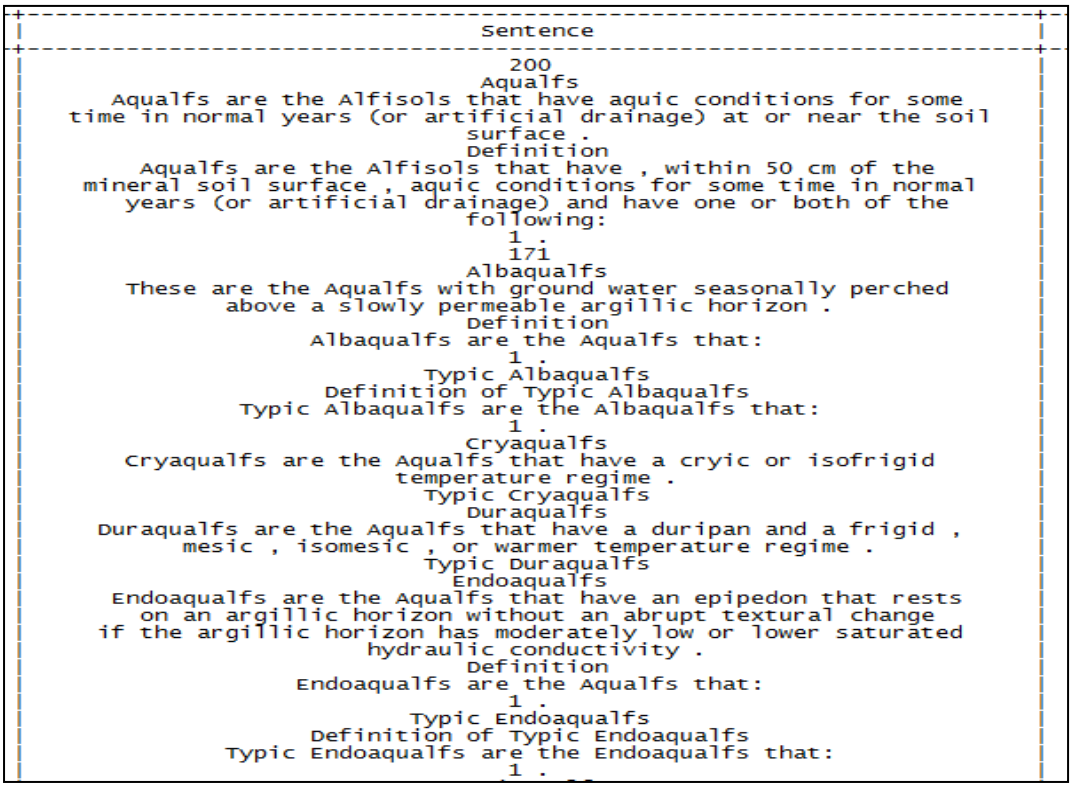


Figure 4.32 Snapshot of a sentence detection which is done by NLP unit from the Corpus

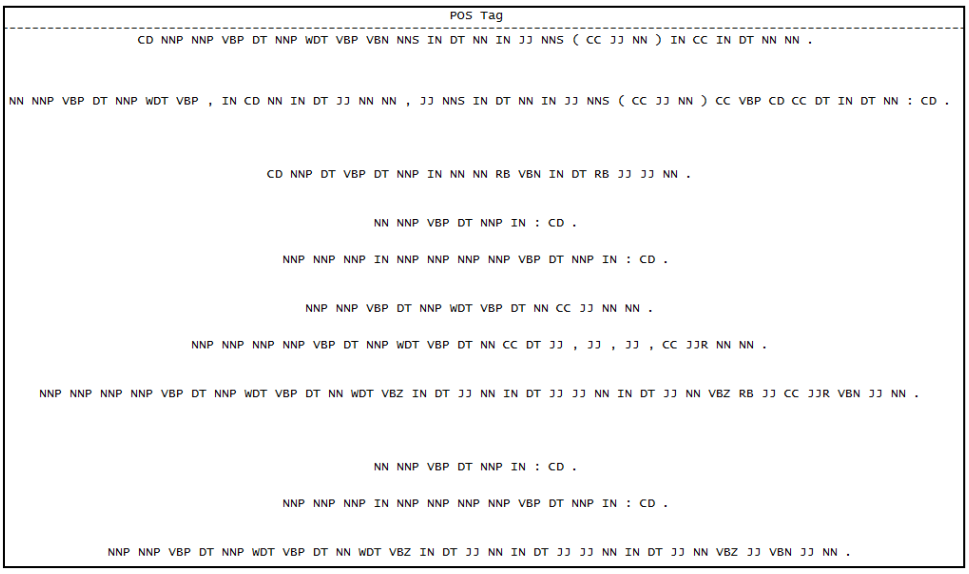


Figure 4.33 Snapshot of POS tagging of the sentences done by using NLP

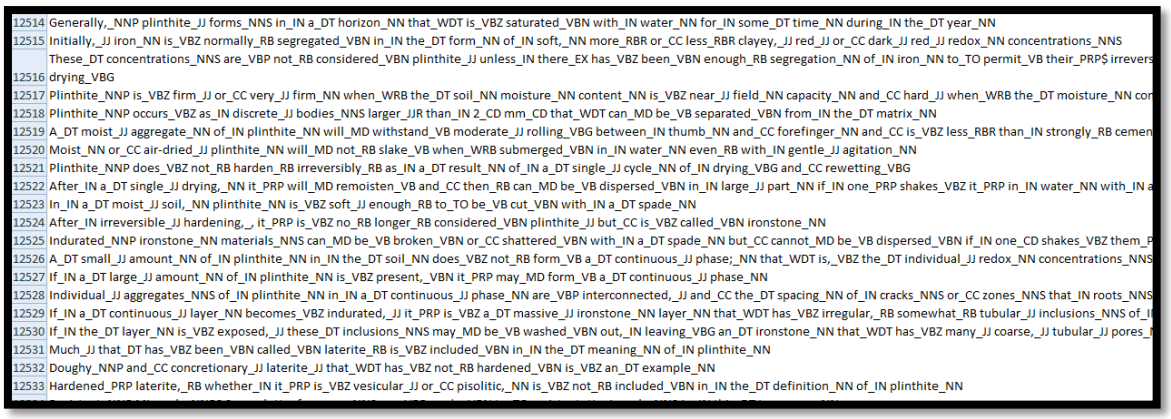


Figure 4.34 Snapshot of the De Tokenization of the sentences with parts of speech tagging done by using NLP

The above figures (4. 32 - 4.34) depicted the snapshots of the results of a NLP module from the Corpus.

4.2.6 Extraction of Hierarchical relationships from Non taxonomic text

Non taxonomic text is very important for capturing the properties of the taxonomic text. The taxonomic class is handled by the methodology described in the sub section **Extraction of hierarchical relationship using enhanced Hearst Pattern from Taxonomic text** and sub section **Development of a framework for Ontology Learning from the taxonomic text** of the section 4.2.5. The second part of the text i.e. non taxonomic text is handled in this section. For dealing with this kind of text, we have followed the methodology described in the section 4.1.2 i.e. the traditional taxonomy induction methodology. In sub section **Segregation of the taxonomic text** of the section 4.2.5 we have seen that there is no significant influence of the encoding techniques in the classifications for the taxonomic text. Here, the extraction of the hierarchy is done by the hierarchical clustering techniques with two kinds of encoding techniques - firstly, TFIDF and secondly, the W2V representation of data.

Table 4.10 : The packages used for hierarchical clustering of non taxonomic text

Packages	Parameters
`gensim.models.word2vec`	`size=150, window=15, min_count=2, workers=multiprocessing.cpu_count()`
`sklearn.decomposition`	`default`
`nltk.stem.porter`	`default`
`nltk.Corpus`	`default`

sklearn.cluster:AgglomerativeClustering	default
scipy.cluster.hierarchy:dendrogram, linkage,leaves_list	truncate_mode='lastp', p=12, show_leaf_counts=False, leaf_rotation=90., leaf_font_size=12., show_contracted=True,

For hierarchical clustering, the objects should be arranged in the Euclidean space for capturing the inherent hierarchy present in the text. Below (Figure 4.35 and 4.36) shows the text object representation in TFIDF and W2V encoding techniques respectively.

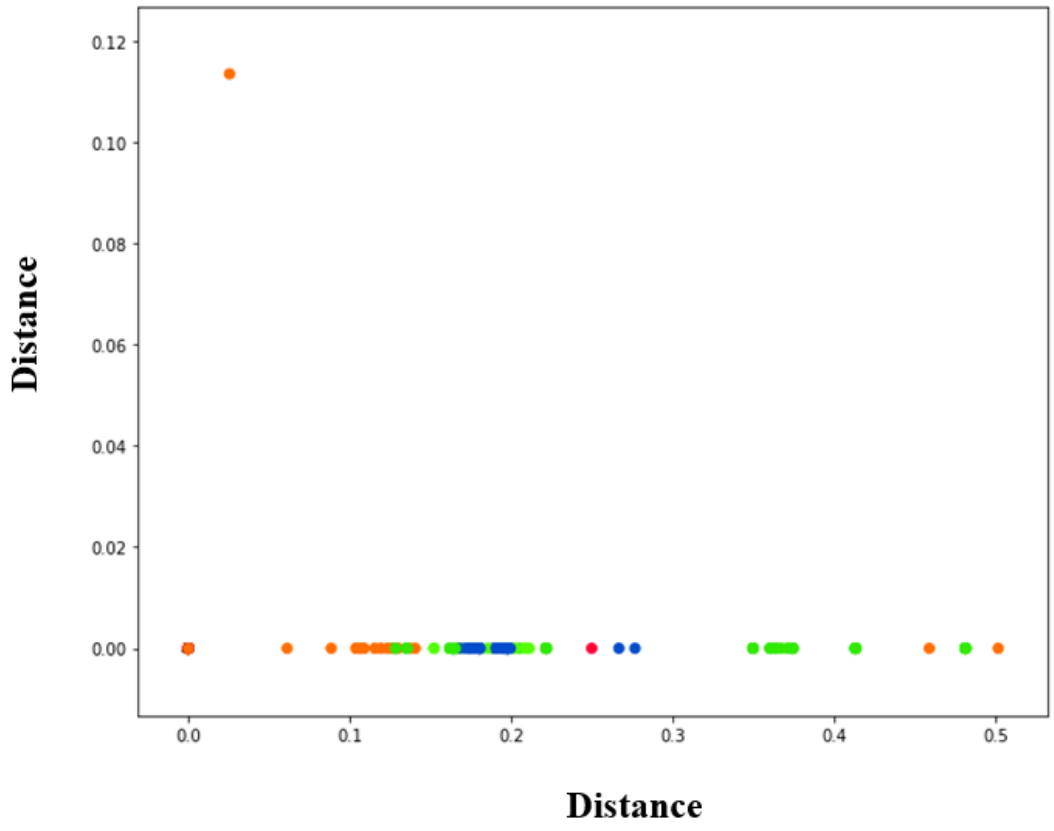


Figure 4.35 Snapshot of Text object representation in TFIDF

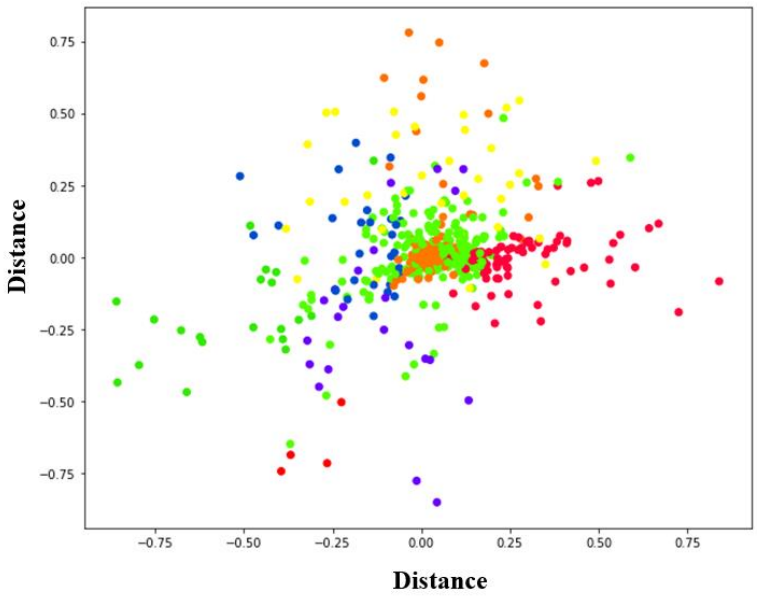


Figure 4.36 Snapshot of Text Object Representation inW2V

The results from the above graphs (Figure 4.35 and 4.36) depicted the plot of the same text object denoting that TFIDF is a poorer representation technique than W2V.

It is at par with the report, that W2V uses approximately 300 dimensions of object representation (Mikolov *et al.*, 2013) which is lacking in TFDIF.

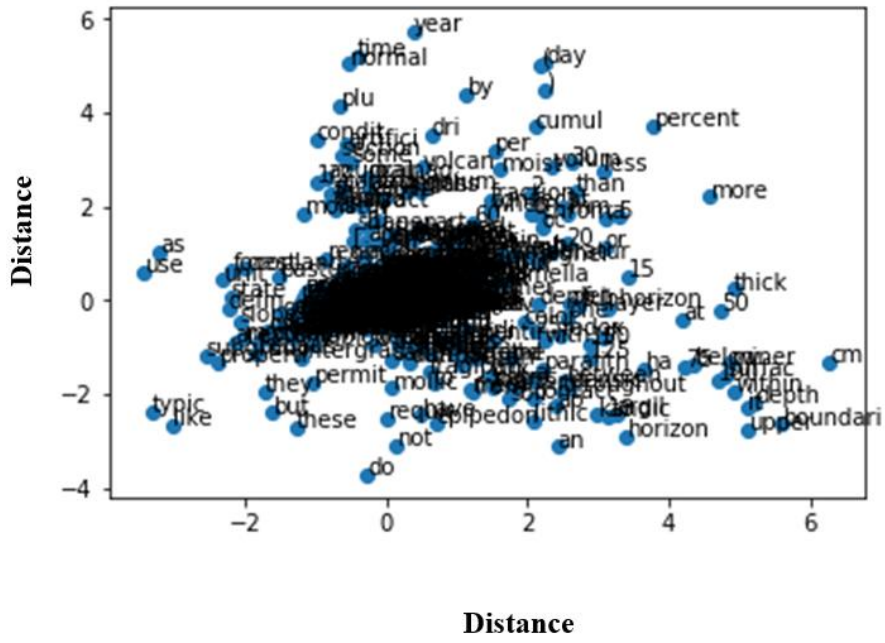


Figure 4.37 Snapshot of Text Object Representation with true label

The snapshot (Figure 4.37) denoted the representation of W2V format with true labeling of domain terms

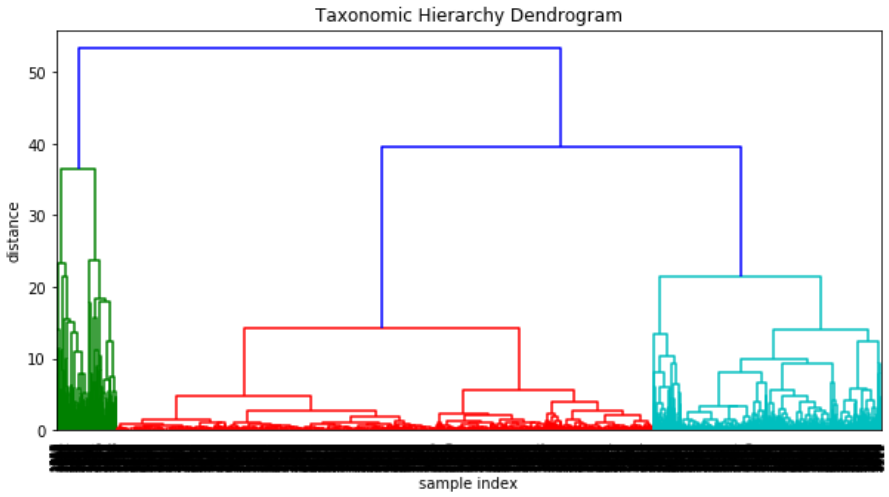


Figure 4.38 Snapshot of the Dendogram of Hierarchical Cluster using W2V encoded text as input

It has been observed that for hierarchical clustering, the W2V representation works well; so among both the representation techniques, the W2V representation is chosen and the further tasks for extracting the hierarchical relationships from the non taxonomic text is done.

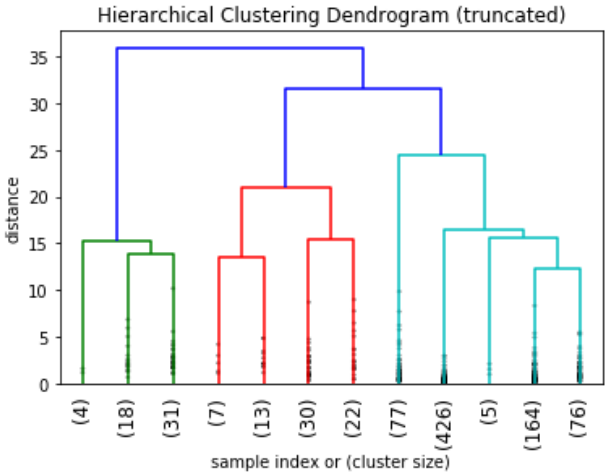


Figure 4.39 Snapshot of the Truncated Dendogram of Hierarchical Cluster using W2V encoded text as input

In the above figures (4.38 and 4.39), non taxonomic text encoded in W2V is taken as input and hierarchical clustering is done to extract the generalized class hierarchy available in the non taxonomic text. Using WordNet the content of the cluster is checked and the output is given in the form of a generalized class. This generalized class broadens the scope of capturing the properties (for e.g. location, geographical area, colour of soil etc.) of the identified taxonomic class (for e.g. Alfisols, Aqualfs etc.)

Table 4.11 : Some snippet of the identified generalized class from Dendogram of Hierarchical Cluster using W2V encoded text as input using WordNet

Identified Generalized Class
Hypernyms: location
Hypernyms: taxonomic_group, taxonomic_category, taxon
Hypernyms: geographical_area, geographic_area, geographical_region, geographic_region
Hypernyms: blood_group, blood_type
Hypernyms: property
Hypernyms: large_integer

It has been noticed that the generalized class captured by WordNet gave consistent result except one.

4.2.7 Association rule mining between taxonomic class and generalized class from the text

The principles of Association rule mining is discussed in the section 3.4.5. This principle will help in the adherence of the generalized class, which is generated from the non taxonomic text with the taxonomic class, generated from taxonomic text.

This result will pave a path ahead towards automated Ontology Learning.

Table 4.12: Some examples of the rule generation from soil taxonomy

Sl.No.	Antecedent	Consequent
1.	{mollic_epipedon, sandy, United_States}	Typic_Albaqualfs
2.	{Aqualfs, kandic_horizon, natric_horizon}	Endoaqualfs
3.	{moist}	Chromic_Vertic_Endoaqualfs
4.	{United States, forests}	Aquandic_Endoaqualfs
5.	{Overlie cindery, fragmental, fragmental,}	Histels
6.	{Alaska,Siberia, and Canada.}	Typic_Hemistels
7.	{gypsic, petrogypsic}	Nitric Anhyorthels

The above table signifies that the individuals of the generalized class which is present in the column Antecedent is matching appropriately with the individuals of the taxonomic class present in the column Consequent.

4.3 Validation of the Ontology Learning Algorithms in Agricultural Domain

Table 4.13: Activity list under the Objective Validation of the Ontology Learning Algorithms in Agricultural Domain

- **Develop and populate the Ontology from selected domain by the developed prototype**

  The following tasks are done:
  - **Lexical Entry Extraction**
  - **Concept Hierarchy From core Taxonomy**
  - **Concept Hierarchy from Non Taxonomy**
  - **Relation Extraction Using ARM (Association Rule Minning)**
- **Validate the developed Ontology with manually developed Ontology by suitable tools and techniques**

  The following tasks are done:
  - **Evaluation of Lexical Entry Extraction**
  - **Evaluation of Concept Hierarchy Core Taxonomy**
  - **Evaluation of Concept Hierarchy from non taxonomic text**
  - **Evaluation of Non Taxonomic Relation Extraction**
- **Compare the enhanced algorithms with the existing algorithms of Ontology Learning**

4.3.1 Development and population of the Ontology from a selected domain by the developed prototype

A working pipeline for Ontology Learning from Taxonomic text has been developed. This section describes the output of populated from the Taxonomic Text. The developed methodology is tested upon the USDA soil taxonomy.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
    xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
    xmlns="http://www.owl-ontologies.com/SoilOntology.owl#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.owl-ontologies.com/SoilOntology.owl">
  <owl:Ontology rdf:about="">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >This is a OWL ontology of soil taxonomic classification.</rdfs:comment>
    <rdfs:label rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Soil Ontology</rdfs:label>
  </owl:Ontology>
  <owl:Class rdf:ID="Basic_Property_entic_dystrusterts">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Basic_Property_dystrusterts"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Petrocalcic_Calcixererts">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Calcixererts"/>
    </rdfs:subClassOf>
  </owl:Class>
```

Figure 4.40: Snapshot of populated Ontology from Taxonomic text

```
<owl:Class rdf:ID="Aquolls">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Mollisols"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Pachic_Argiudolls">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Argiudolls"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Basic_Property_lamellic_haploxerults">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Basic_Property_haploxerults"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Basic_Property_fragic_paleudults">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Basic_Property_paleudults"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:ID="Aquic_Fragiorthods">
  <rdfs:subClassOf>
    <owl:Class rdf:ID="Fragiorthods"/>
  </rdfs:subClassOf>
```

Figure 4.41: Snapshot of the populated soil properties from Taxonomic text

Figure 4.40, Figure 4.41 depicts snippet of the results from populated Ontology. The automated process of Ontology Learning from Taxonomic Text has populated approximately 1500 taxonomic classes out of 1900 taxonomic classes. Approximately, 0.26 million taxonomic relationships have been captured out of 0.31 million. Approximately, 11 generalized classes has been identified which has the potential to capture the property of the taxonomic class.

4.3.2 Validation of the developed Ontology with manually developed Ontology by suitable tools and techniques

To validate the developed methodology, a set of algorithms have been tested in the USDA soil taxonomy. The validation is done in the soil taxonomy chapter available in the mentioned taxonomy text. In the following sub section, each chapter is used as a dataset for proper validation of the methodology that has been developed.

- **Validation of Lexical Entry Extraction from Taxonomic Text**

The lexical entry extraction has been done by the developed methodology described in the section 3.4.4. In this study, 10 chapters of the book "**Key to soil Taxonomy**" is taken as dataset and validated.

Table 4.14: Performance measure of Lexical entry extraction methodology of term identification using F- Score

Dataset	Term Identified	Total Term	Precision	Recall	F-Score
1	7385	10401	0.83	0.73	0.78
2	4296	6072	0.84	0.73	0.78
3	4102	6107	0.81	0.69	0.74
4	3494	5948	0.8	0.60	0.68
5	2238	2858	0.79	0.79	0.79
6	1096	1645	0.8	0.68	0.74
7	4901	6936	0.82	0.72	0.76
8	7593	10686	0.84	0.73	0.78
9	3022	4460	0.81	0.69	0.75
10	2168	2799	0.84	0.79	0.82

Table 4.13 depicted the performance measurement of the Lexical Entry Extraction. This extraction of the lexical entry is for two class identification i.e. whether a term should be included in the Lexical Entry List or not. The observation shows that the term extraction unit gives reasonably good results by looking into the F score column of the table.

- **Validation of extraction of Concept Hierarchy from the Core Taxonomy**

In this study, a comparison between the identified class and actual taxonomic class is done. The taxonomic text used for the study consisted of 4 levels (Order to Subgroup). The developed algorithm is able to identify all the classes available in the taxonomic text upto the Great Group level. It also identified nearly 78% of the classes in the Sub Group level. It is shown in the bar diagram below:

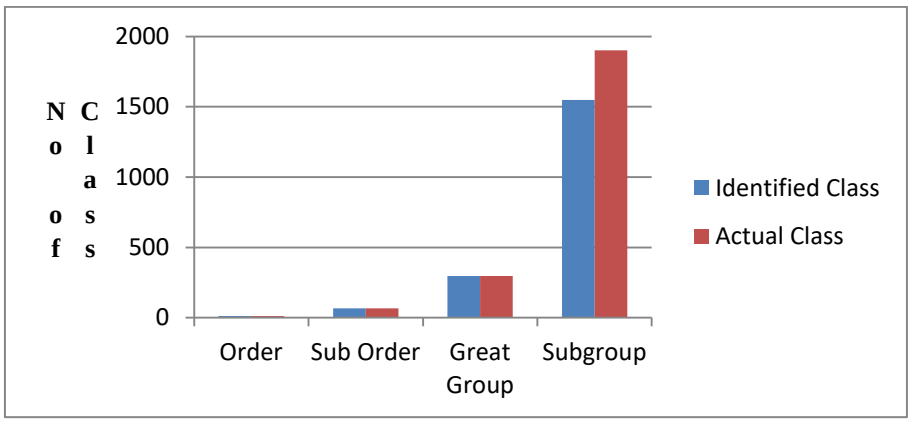


Figure 4.42 Comparison between the identified and the actual classes from the USDA soil Taxonomy

Table 4.15: Results of the taxonomic relation extraction available in the taxonomic text among the taxonomic classes

Dataset(Book Chapters)	Relations Existed in the taxonomic text	Identified Extracted Relations
1	72000	58200
2	52332	43904
3	29250	23700
4	5400	4260
5	3840	3264
6	38976	32480
7	80928	64224
8	23100	18700
9	8240	8240
10	15900	12750

For an Ontology Learning Task identification of the Class is not wholly sufficient; the hierarchical relationships among the classes are also important. In the Table 4.15, it has been seen that approximately 81% of the hierarchical relationships has been extracted using methodology.

- **Validation Relations present in the Taxonomic and non Taxonomic Text**

Using Association rule mining a relation is tried to be extracted between the taxonomic and non taxonomic text. Here, 10 chapters of the book "**Key to soil Taxonomy**" is taken as dataset. A total of 8564 relations are existing, among which 4320 relations are extracted, which is 50% of the total existing relationships.

Table 4.16 Comparisons between the existed relations and the extracted relations in the taxonomic text

Dataset	Relations Exists	Extracted Relations
1	1584	800
2	1212	560
3	456	234
4	780	453
5	1284	640
6	876	472
7	952	460
8	512	258
9	568	282
10	340	161

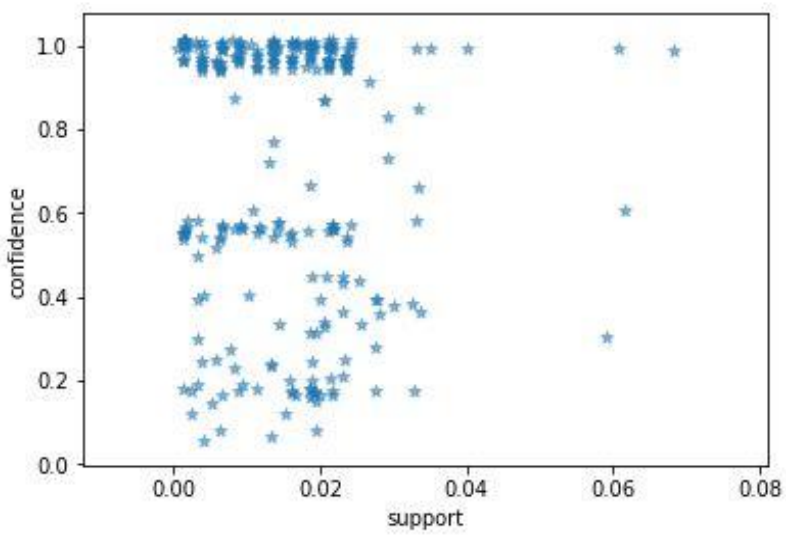


Figure 4.43 Scatter plots of Support and Confidence metrics of the generated rules from the Taxonomic Text

Figure 4.43 depicts the support confidence scatter plot of the rules. The rules are depicted as star symbol in the plot, and most of the generated rules are in the middle portion of the graph that means the rules generated have a sufficient chance to be repeated.

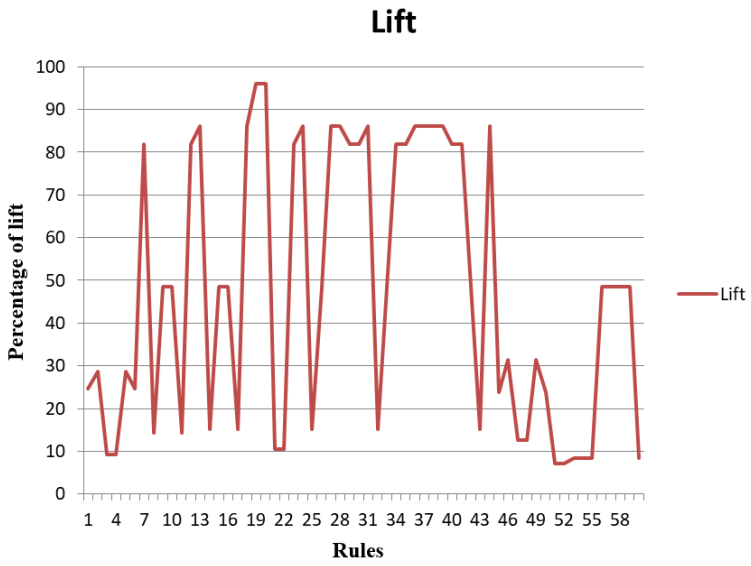


Figure 4.44 Description of lift of generated rule where Y axis depicts the percentage value of the lift and the X- axis depicts the rule number

Figure 4.44 depicts the extracted rules from the given text. Where the Y-axis depicts the percentage value of lift and the X-axis depicts the rule number. The graph suggests that the lifts of the extracted rules are stable. Lift describes the ratio between the expected confidence and the actual confidence. That means it is safe to rely on the extracted rule because most of the rules are equal or above 10 % of the lift.

4.3.3 Comparative discussion of the enhanced algorithms with the existing algorithms of Ontology Learning

Comparing of the existing Ontology Learning algorithms with the enhanced one is difficult. In many aspects, this study is totally different from the existing Ontology Learning problem. Introduction of segregation of the hierarchical text into non hierarchical one through the classification technique is done. For the conventional Ontology Learning, consideration of the whole text into a single category was there, but in this study, it has been realized that the taxonomic text has significant morphological differences than the plain text. So, introduction of the segregation of the taxonomic text into hierarchical and non-hierarchical is done. This approach is not comparable with existing Ontology Learning approaches.

In the section 4.1.1 the existing lexical entry extraction algorithms has been discussed. In subsequent subsections, it is seen that the conventional methods are unable to extract multi word important keywords. For combating the situation, introduction of a heuristic method in the taxonomic text for extracting actual important words are done. In the sub section **Heuristic Methods of identification of multiword keyword (lexical**

entry) by using hybrid method comprising of RAKE and W2V of the section 4.2.5 the results depicted betterment over the conventional methods of lexical entry extraction.

In the sub section **Taxonomy induction From Hierarchical Part of the Text** of the section 3.4.4 we have discussed the extraction of hierarchy from plain text using Hearst pattern. The traditional method was unable to extract even a single true hierarchy from the taxonomic text. Here the enhancement of the pattern according to our text is done. This enhancement is very much dependent on the domain and text available in that domain. It worked well in this domain and it need not necessarily mean that it will work in the next domain. To combat this situation, Ontology engineer may need to find the new pattern that will be capable of extraction of hierarchy in the new domain.

In conclusion, it can be said that after a thorough study of the Taxonomic text, it is seen that it can be classified into hierarchical and non-hierarchical one. A hybrid methodology for keyword extraction using RAKE and W2V is done which heuristically can extract the domain related important multiword keywords. Using enhanced Hearst Pattern and Hierarchical clustering with the help of WordNet, the hierarchical relationship is extracted from both the core taxonomic and non taxonomic text respectively. Association rule mining has been used in the study, to find the relationship between the taxonomic and non taxonomic text which is pertinent for automation of Ontology Learning. Validation of the developed as well as enhanced algorithm has been done in the Agricultural domain (USDA Soil Taxonomy) using standard validation technique of information retrieval like Precision recall, F-Score, Percentage success for relationship extraction, Support-Confidence –Lift for the generated rule from taxonomic text.

CHAPTER V

SUMMARY AND CONCLUSION

In this research work, Ontology Learning from specialized text i.e. the taxonomic text has been studied. This study is unique for dealing with the mentioned text. The study dealt with the challenges of natural language processing of taxonomic text. On the other hand, it also exploits the typical characteristics of the taxonomic text.

This study has developed a framework of Ontology Learning and it is tested in the taxonomic text available in agriculture particularly the USDA soil taxonomy. The Ontology Leaning task starts with the population of the Corpus. Here the Corpus development starts with two sources of data one is the taxonomic text of the user and the second source is the automated scraping from the Wikipedia, which increases the domain related content in the Corpus.

After the development of the Corpus, the pipelines go through some pre-processing of the natural language available in the Corpus. The pre-processing includes the sentence detection, tokenization, POS tagging, stemming and lemmatization.

After the basic pre-processing task, the input text is trained and tested on some classification techniques. The developed model helped in classifying the text into hierarchical and nonhierarchical text.

Before entering into the core task of the Ontology Learning extraction of the key phrase identification is very important. A detailed study has been done for the existing keyword extraction methodology used in the Ontology Learning. The popular methods W2V (Word to Vector) and RAKE (Rapid Automatic Keywords Extraction) methods are taken into considerations and a heuristic hybrid method of keyword extraction has been developed. The developed hybrid method gives better results than the existing two methods.

From the above description, it is clear that the pipeline is bifurcated into the taxonomy extraction and non taxonomic relationship extraction. In taxonomic hierarchy, the extraction is done on the POS tagged input text by the lexico syntactic filter (Hearst Pattern) with tree based (Connective based) given approach. The extracted taxonomy is

compared with the manually developed ontology. The lexico-syntactic taxonomy extraction method gives reasonably good result compared to the manual ontology.

The second part of the bifurcation is to identify the non taxonomic relationship among the entity that is not the part of the taxonomic hierarchy. To extract the non taxonomic relationship the Word2Vec model is used to encode the input text.

The present study deals with the Ontology Learning from taxonomic texts. The developed methodology has been validated in the agricultural domain. This study critically analyzes the present method of ontology development and proposed an approach that works better in the taxonomic text.

In future, the Ontology pruning and maintenance may be considered. Further the present framework can be extended for deep learning enabled pipeline that may improve the result of Ontology Learning.

Ontology Learning From Taxonomic Text for Agricultural Knowledge Management

ABSTRACT

Ontology Learning from Taxonomic text is a novel approach of learning Ontology from semi structured taxonomic text. The traditional ontology learning within the available literature mostly focuses on the ontology learning from the huge corpus of text. The present study mainly concentrates on the ontology learning from the specialized kind of text i.e. the taxonomic text. This study dealt with the exploitation of the typical characteristics of the taxonomic text. The study eventually is subdivided into mainly four broad areas. First, it has developed a text corpus from the taxonomic text of USDA soil taxonomy and enhanced the corpus by the automated scraping of the Wikipedia, with the help of seed word given by the domain experts. After the development of the corpus, the keyword extraction was a challenging task for this research. A heuristic methodology has been developed which is used for the extraction of the keyword. The heuristic method is based on the RAKE guided by the W2V methodology. Second part of the study dealt with the taxonomy induction from the text which contains the core taxonomy. To segregate the core taxonomy part, we have used machine learning techniques for the classification of text. Third part of the study dealt with the taxonomy induction from the non taxonomic part of the text. We have used the hierarchical clustering to induct the taxonomy from the text. Fourth and last part of the work is dealt with the finding of the connections between the taxonomic and non taxonomic class that has been inducted by the second and third part of the work. Several empirical results have been provided and validated using suitable tools and techniques in USDA Soil Taxonomy. Total study involved a wide range of technologies and software. Most of the algorithms are implemented in Python programming language. Some of the experiment involves Java and SQL server. We have also used protégé for the study of existing manually developed ontology.

Keywords: *Ontology Learning; taxonomic text; USDA soil taxonomy; RAKE; W2V*

सार

टैक्सोनॉमिक टेक्स्ट से ओन्टोलॉजी सीखना अर्ध-संरचित टैक्सोनॉमिक टेक्स्ट से ओन्टोलॉजी सीखने का एक नया तरीका है। पारंपरिक ओन्टोलॉजी सीखने के मुख्य तरीकों में उपलब्ध साहित्य में मौजूद टेक्स्ट के विशाल ग्रंथ-संग्रह (कॉर्पस) से ओन्टोलॉजी सीखना रहा है। वर्तमान अध्ययन मुख्य रूप से विशेष प्रकार के टेक्स्ट यानी टैक्सोनॉमिक टेक्स्ट से ओन्टोलॉजी सीखने पर केंद्रित है। वर्तमान अध्ययन टैक्सोनॉमिक टेक्स्ट के विशिष्ट विशेषताओं के उपयोग से संबंधित है। इस अध्ययन को मुख्य रूप से चार व्यापक क्षेत्रों में विभाजित किया गया है। सबसे पहले, हमने यूएसडीए मिट्टी वर्गीकरण के टैक्सोनॉमिक टेक्स्ट से एक टेक्स्ट ग्रंथ-संग्रह विकसित किया है और क्षेत्र-विशेषज्ञों द्वारा दिए गए बीज-शब्द की मदद से विकिपीडिया से स्क्रैप करके स्वचालित रूप से ग्रंथ-संग्रह को बढ़ाया है। ग्रंथ-संग्रह के विकास के बाद, कीवर्ड निष्कर्षण एक चुनौतीपूर्ण कार्य था जिसके लिए हेयुरिस्टिक विधि (अनुमानी कार्य-प्रणाली) विकसित की गयी जिसका उपयोग कीवर्ड के निष्कर्षण के लिए किया जाता है। हेयुरिस्टिक विधि W2V पद्धति द्वारा निर्देशित RAKE पर आधारित है। अध्ययन का दूसरा हिस्सा उस टेक्स्ट के टैक्सोनोमी प्रवर्तन (अनुमान) से संबंधित है जिसमें मुख्य टैक्सोनोमी शामिल होता है। मुख्य टैक्सोनॉमिक पार्ट को अलग करने हेतु, टेक्स्ट के वर्गीकरण के लिए मशीन लर्निंग तकनीक का इस्तेमाल किया है। अध्ययन का तीसरा भाग टेक्स्ट के गैर-टैक्सोनॉमिक भाग से टैक्सोनॉमी को शामिल करने से सम्बंधित है। टेक्स्ट से टैक्सोनोमी को शामिल करने के लिए श्रेणीबद्ध क्लस्टरिंग का उपयोग किया है। कार्य का चौथा और अंतिम भाग उस टैक्सोनॉमिक और गैर-टैक्सोनॉमिक वर्ग के बीच संबंध की खोज से सम्बंधित है जिसे कार्य के दूसरे और तीसरे भाग द्वारा सम्मिलित किया गया है। इसके लिए कई अनुभवजन्य परिणाम प्रदान भी किए गए हैं। प्रस्तुत अध्ययन में प्रौद्योगिकी और सॉफ्टवेयर की एक विस्तृत श्रृंखला शामिल रही है। अधिकांश एल्गोरिदम पायथन प्रोग्रामिंग भाषा में कार्यान्वित किए गए हैं। शोध के कुछ हिस्सों में जावा और एसक्यूएल सर्वर का उपयोग किया गया हैं। मैन्युअल रूप से विकसित ओन्टोलॉजी के अध्ययन के लिए 'प्रोटेग' का भी उपयोग किया है।

कीवर्ड: *ऑन्टोलॉजी लर्निंग; वर्गीकरण पाठ ; यूएसडीए मिट्टी वर्गीकरण ; RAKE; W2V*

BIBLIOGRAPHY

Angeli, G., Premkumar, M. J. J., & Manning, C. D. (2015, July). Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 344-354).

Asim, M. N., Wasim, M., Khan, M. U. G., Mahmood, W., & Abbasi, H. M. (2018). A survey of ontology learning techniques and applications. Database, 2018.

Atapattu, T., Falkner, K., & Falkner, N. (2017). A comprehensive text analysis of lecture slides to generate concept maps. *Computers & Education*, *115*, 96-113.

Baader, F., Lutz, C., Sturm, H., & Wolter, F. (2002). Fusions of description logics and abstract description systems. *Journal of Artificial Intelligence Research*, *16*, 1-58.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., & Etzioni, O. (2007, January). Open information extraction from the web. In *IJCAI* (Vol. 7, pp. 2670-2676).

Bedi P., Marwaha S., 2004. Designing Ontologies from Traditional Taxonomies, In the Proceedings of International Conference on Cognitive Science, Allahabad, India.

Biswas, S., Marwaha, S., Malhotra, P. K., Wahi, S. D., Dhar, D. W., & Singh, R. (2013). Building and querying microbial ontology. *Procedia Technology*, *10*, 13-19.

Boole, G. (1847). The mathematical analysis of logic. Philosophical Library.

Boole, G. (1848). The calculus of logic. Cambridge and Dublin Mathematical Journal, 3(1848), 183-198.

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Brickley, D., & Guha, R. (1999). Resource Description Framework (RDF) schema specification. W3C Proposed Recommendation, March 1999. *W3C-World Wide Web Consortium,[Online] http://www. w3. org/TR/PR-rdfschema*.

Brickley, D., & Guha, R. V. (2000). Resource Description Framework (RDF) Schema Specification 1.0, W3C Candidate Recommendation 27 March 2000.

Brickley, D., Guha, R. V., & McBride, B. (2004). RDF vocabulary description language 1.0: RDF Schema. W3C Recommendation (2004). *URL http://www. w3. org/tr/2004/rec-rdf-schema-20040210*.

Brill, E. (1992, March). A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing* (pp. 152-155). Association for Computational Linguistics.

Buitelaar, P., Cimiano, P., & Magnini, B. (Eds.). (2005). *Ontology learning from text: methods, evaluation and applications* (Vol. 123). IOS press.

Bunescu, R. C., & Mooney, R. J. (2005, October). A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing* (pp. 724-731). Association for Computational Linguistics.

Caraballo, S. A., & Charniak, E. (2001). *Automatic construction of a hypernym-labeled noun hierarchy from text*. Brown University.

Cederberg, S., &Widdows, D. (2003, May). Using LSA and noun coordination information to improve the precision and recall of automatic hyponymy extraction. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4* (pp. 111-118). Association for Computational Linguistics.

Chatterjee, N., &Kaushik, N. (2017). RENT: Regular expression and NLP-based term extraction scheme for agricultural domain. In *Proceedings of the international conference on data engineering and communication technology* (pp. 511-522). Springer, Singapore.

Ciaramita, M., Gangemi, A., Ratsch, E., Saric, J., & Rojas, I. (2005, July). Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In *IJCAI*(pp. 659-664).

Dasgupta, S., Padia, A., Shah, K., KaPatel, R., & Majumder, P. (2013). DLOLIS-A: Description Logic based Text Ontology Learning. *arXiv preprint arXiv:1303.5929*.

De Chalendar, G., &Grau, B. (2000, October). SVETLAN'or how to Classify Words using their Context. In *International Conference on Knowledge Engineering and Knowledge Management* (pp. 203-216). Springer, Berlin, Heidelberg.

De Morgan, A. (1847). Formal logic: or, the calculus of inference, necessary and probable. Taylor and Walton.

Deb, C. K., Marwaha, S., Malhotra, P. K., Wahi, S. D., &Pandey, R. N. (2015, March). Strengthening soil taxonomy ontology software for description and classification of USDA soil taxonomy up to soil series. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 1180-1184). IEEE.

Decker, S., Fensel, D., Van Harmelen, F., Horrocks, I., Melnik, S., Klein, M. C., & Broekstra, J. (2000). Knowledge Representation on the Web. *Description Logics*, *33*, 89-97.

Dong, H., &Hussain, F. K. (2013). SOF: a semi-supervised ontology-learning-based focused crawler. *Concurrency and Computation: Practice and Experience*, *25*(12), 1755-1770.

Dong, H., Hussain, F. K., & Chang, E. (2013). Semantic Web Service matchmakers: state of the art and challenges. *Concurrency and Computation: Practice and Experience*, *25*(7), 961-988.

Dorr, B. J., Lee, J. H., Lin, D., & Suh, S. (1995). Squibs and Discussions: Efficient Parsing for Korean and English: A Parameterized Message-Passing Approach. *Computational Linguistics*, *21*(2).

Drymonas, E. G. (2009). Ontology learning from text based on multiword term concepts: the ontogain method. *Master of Science thesis, Technical University of Crete, Greece*.

Drymonas, E., Zervanou, K., & Petrakis, E. G. (2010, June). Unsupervised ontology acquisition from plain texts: the OntoGain system. In *International Conference on Application of Natural Language to Information Systems* (pp. 277-287). Springer, Berlin, Heidelberg.

El Ghosh, M., Naja, H., Abdulrab, H., & Khalil, M. (2017). Towards a legal rule-based system grounded on the integration of criminal domain ontology and rules. *Procedia Computer Science*, *112*, 632-642.

Faure, D., &Poibeau, T. (2000, August). First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX. In *Ontology Learning ECAI-2000 Workshop* (pp. 7-12).

Faure, D., Nédellec, C., &Rouveirol, C. (1998). Acquisition of Semantic Knowledge using Machine learning methods: The System" ASIUM". In *Universite Paris Sud*.

Fortuna, B., Grobelnik, M., &Mladenic, D. (2007, July). OntoGen: semi-automatic ontology editor. In *Symposium on Human Interface and the Management of Information* (pp. 309-318). Springer, Berlin, Heidelberg.

Fortuna, B., Lavrač, N., &Velardi, P. (2008, December). Advancing topic ontology learning through term extraction. In *Pacific Rim International Conference on Artificial Intelligence*(pp. 626-635). Springer, Berlin, Heidelberg.

Frantzi, K., Ananiadou, S., &Mima, H. (2000). Automatic recognition of multi-word terms:.the c-value/nc-value method. *International journal on digital libraries*, *3*(2), 115-130.

Frikh, B., Djaanfar, A. S., &Ouhbi, B. (2011). A new methodology for domain ontology construction from the web. *International Journal on Artificial Intelligence Tools*, *20*(06), 1157-1170.

Fundel, K., Küffner, R., & Zimmer, R. (2006). RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, *23*(3), 365-371.

Gangemi, A., Catenacci, C., Ciaramita, M., & Lehmann, J. (2005, December). A theoretical framework for ontology evaluation and validation. In *SWAP* (Vol. 166, p. 16).

Gennari, J. H., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., ... & Tu, S. W. (2003). The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-computer studies*, *58*(1), 89-123.

Gil, R. J., & Martín-Bautista, M. J. (2012). A novel integrated knowledge support system based on ontology learning: Model specification and a case study. *Knowledge-Based Systems*, *36*, 340-352.

Gil, R., & Martin-Bautista, M. J. (2014). SMOL: a systemic methodology for ontology learning from heterogeneous sources. *Journal of Intelligent Information Systems*, *42*(3), 415-455.

Girju, R., Badulescu, A., & Moldovan, D. (2003, May). Learning semantic constraints for the automatic discovery of part-whole relations. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1* (pp. 1-8). Association for Computational Linguistics.

Hahn, U., &Markó, K. G. (2002). An integrated, dual learner for grammars and ontologies. *Data & Knowledge Engineering*, *42*(3), 273-291.

Hahn, U., &Romacker, M. (2001, March). The SYNDIKATE text Knowledge base generator. In *Proceedings of the first international conference on Human language technology research* (pp. 1-6). Association for Computational Linguistics.

Hahn, U., &Schnattinger, K. (1998). Towards text knowledge engineering. *Hypothesis*, *1*(2).

Hayes, P. J. (1981). The logic of frames. In *Readings in Artificial Intelligence* (pp. 451-458). Morgan Kaufmann.

Hearst, M. A. (1998). Automated discovery of WordNet relations. *WordNet: an electronic lexical database*, 131-153.

Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., &Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their applications*, *13*(4), 18-28.

Idoudi, R., Ettabaa, K. S., Solaiman, B., &Hamrouni, K. (2016). Ontology knowledge mining based association rules ranking. *Procedia Computer Science*, *96*, 345-354.

Ismail, R., Bakar, Z. A., & Rahman, N. A. (2015). Extracting knowledge from English translated Quran using NLP pattern. *Jurnal Teknologi*, *77*(19).

Jiang, X., & Tan, A. H. (2010). CRCTOL: A semantic-based domain ontology learning system. *Journal of the American Society for Information Science and Technology*, *61*(1), 150-168.

Kambhatla, N. (2004, July). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions* (p. 22). Association for Computational Linguistics.

Kang, S., Patil, L., Rangarajan, A., Moitra, A., Jia, T., Robinson, D., & Dutta, D. (2015, August). Extraction of manufacturing rules from unstructured text using a semantic framework. In *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. American Society of Mechanical Engineers Digital Collection.

Karoui, L., Aufaure, M. A., & Bennacer, N. (2007, April). Analyses and fundamental ideas for a relation extraction approach. In *2007 IEEE 23rd International Conference on Data Engineering Workshop* (pp. 880-887). IEEE.

Kaushik, N., & Chatterjee, N. (2018). Automatic relationship extraction from agricultural text for ontology construction. *Information processing in agriculture*, *5*(1), 60-73.

Klyne, G., & Carroll, J. J. (2003). Resource Description Framework (RDF): Concepts and abstract syntax. W3C Working Draft.

Kowalski, R. (1974, June). Predicate logic as programming language. In IFIP congress (Vol. 74, pp. 569-544).

Kumara, B. T., Paik, I., Chen, W., &Ryu, K. H. (2014). Web service clustering using a hybrid term-similarity measure with ontology learning. *International Journal of Web Services Research (IJWSR)*, *11*(2), 24-45.

Lewis, C. I., & Leibniz, G. W. (1918). A survey of symbolic logic. University of California Press.

Lin, D. (1994). PRINCIPAR---An Efficient, broad-coverage, principle-based parser. *arXiv preprint cmp-lg/9407024*.

Lin, D. (1998, July). An information-theoretic definition of similarity. In *Icml* (Vol. 98, No. 1998, pp. 296-304).

Liu, J. N., He, Y. L., Lim, E. H., & Wang, X. Z. (2014). Domain ontology graph model and its application in Chinese text classification. *Neural Computing and Applications*, *24*(3-4), 779-798.

Lyal, C., Kirk, P., Smith, D., & Smith, R. (2008). The value of taxonomy to biodiversity and agriculture. *Biodiversity*, *9*(1-2), 8-13.

Maedche, A., & Staab, S. (2001). Ontology learning for the semantic web. *IEEE Intelligent systems*, *16*(2), 72-79.

Maedche, A., &Staab, S. (2000, August). Discovering conceptual relations from text. In *Ecai* (Vol. 321, No. 325, p. 27).

Maedche, A., &Staab, S. (2000, August). The text-to-onto ontology learning environment. In *Software Demonstration at ICCS-2000-Eight International Conference on Conceptual Structures* (Vol. 38, pp. 890-930). sn.

Maedche, A., &Volz, R. (2001, November). The ontology extraction & maintenance framework Text-To-Onto. In *Proc. Workshop on Integrating Data Mining and Knowledge Management, USA* (pp. 1-12).

Manoranjan, D., Malhotra, P. K., Sudeep, M., & Pandey, R. N. (2012). Building and querying soil ontology. *Journal of the Indian society of agricultural statistics*, *66*(3), 459-464.

Manoranjan, D., Malhotra, P. K., Sudeep, M., &Pandey, R. N. (2012). Building and querying soil ontology. *Journal of the Indian society of agricultural statistics*, *66*(3), 459-464.

Martschat, S., Cai, J., Broscheit, S., Mújdricza-Maydt, E., &Strube, M. (2012, July). A multigraph model for coreference resolution. In *Joint Conference on EMNLP and CoNLL-Shared Task* (pp. 100-106). Association for Computational Linguistics.

Morita, T., Shigeta, Y., Sugiura, N., Fukuta, N., Izumi, N., & Yamaguchi, T. (2004). DOODLE-OWL: OWL-based Semi-Automatic Ontology Development Environment. In *EON*.

Navigli, R., Velardi, P., &Gangemi, A. (2003). Ontology learning and its application to automated terminology translation. *IEEE Intelligent systems*, *18*(1), 22-31.

Oliveira, A., Pereira, F. C., & Cardoso, A. (2001). Automatic reading and learning from text. In *Proceedings of the international symposium on artificial intelligence (ISAI)*. sn.

Paiva, L., Costa, R., Figueiras, P., & Lima, C. (2014, June). Discovering semantic relations from unstructured data for ontology enrichment: Asssociation rules based approach. In *2014 9th Iberian Conference on Information Systems and Technologies (CISTI)* (pp. 1-6). IEEE.

Panchenko, A., Faralli, S., Ruppert, E., Remus, S., Naets, H., Fairon, C., ...&Biemann, C. (2016). Taxi at semeval-2016 task 13: a taxonomy induction method based on lexico-syntactic patterns, substrings and focused crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* (pp. 1320-1327).

Pantel, P., & Pennacchiotti, M. (2006, July). Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* (pp. 113-120). Association for Computational Linguistics.

Pantel, P., &Ravichandran, D. (2004). Automatically labeling semantic classes. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*.

Parte, A. (2012). *Bergey's manual of systematic bacteriology: Volume 5: The actinobacteria*. Springer Science & Business Media.

Petit, J., Boisson, J. C., & Rousseaux, F. (2017, April). Discovering cultural conceptual structures from texts for ontology generation. In *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)* (pp. 0225-0229). IEEE.

Petrucci, G., &Dragoni, M. (2016, May). The IRMUDOSA system at ESWC-2016 challenge on semantic sentiment analysis. In *Semantic Web Evaluation Challenge* (pp. 126-140). Springer, Cham.

Qi, C., Fourie, A., & Chen, Q. (2018). Neural network and particle swarm optimization for predicting the unconfined compressive strength of cemented paste backfill. *Construction and Building Materials*, *159*, 473-478.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of artificial intelligence research*, *11*, 95-130.

Richens, R. H. (1956). Preprogramming for mechanical translation. *Mechanical Translation*, *3*(1), 20-25.

Riloff, E., & Shepherd, J. (1997). A corpus-based approach for building semantic lexicons. *arXiv preprint cmp-lg/9706013*.

Riloff, E., & Shepherd, J. (1999). A corpus-based bootstrapping algorithm for semi-automated semantic lexicon construction. *Natural Language Engineering*, *5*(2), 147-156.

Roark, B., &Charniak, E. (1998, August). Noun-phrase co-occurrence statistics for semiautomatic semantic lexicon construction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 2* (pp. 1110-1116). Association for Computational Linguistics.

Schmid, H. (1994, August). Part-of-speech tagging with neural networks. In *Proceedings of the 15th conference on Computational linguistics-Volume 1* (pp. 172-176). Association for Computational Linguistics.

Sen, S., Tao, J., & Deokar, A. V. (2015). On the role of ontologies in information extraction. In *Reshaping Society through Analytics, Collaboration, and Decision Support* (pp. 115-133). Springer, Cham.

Shamsfard, M., &Barforoush, A. A. (2002, June). An introduction to HASTI: an ontology learning system. In *Proceedings of the iasted international conference artificial intelligence and soft computing, Acta Press, Galgary, Canada*(pp. 242-247).

Shamsfard, M., &Barforoush, A. A. (2003). The state of the art in ontology learning: a framework for comparison. *The Knowledge Engineering Review*, *18*(4), 293-316.

Shamsfard, M., &Barforoush, A. A. (2004). Learning ontologies from natural language texts. *International journal of human-computer studies*, *60*(1), 17-63.

Sleator, D. D., & Temperley, D. (1995). Parsing English with a link grammar. *arXiv preprint cmp-lg/9508004*.

Smith, R., Rassmann, K., Davies, H., & King, N. (2011). Why taxonomy matters. *BioNET-International, Egham, UK*.

Snow, R., Jurafsky, D., & Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In *Advances in neural information processing systems* (pp. 1297-1304).

Sombatsrisomboon, R., Matsuo, Y., & Ishizuka, M. (2003). Acquisition of hypernyms and hyponyms from the WWW. In *Proceedings of the 2nd International Workshop on Active Mining*.

Sordo, M., Oramas, S., & Espinosa-Anke, L. (2015, June). Extracting relations from unstructured text sources for music recommendation. In *International Conference on Applications of Natural Language to Information Systems* (pp. 369-382). Springer, Cham.

Suresu, S., &Elamparithi, M. (2016). Probabilistic relational concept extraction in ontology learning. *Int. J. Inform. Technol*, *2*.

Tamagawa, S., Sakurai, S., Tejima, T., Morita, T., Izumi, N., & Yamaguchi, T. (2010, August). Learning a large scale of ontology from Japanese Wikipedia. In *2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology* (Vol. 1, pp. 279-286). IEEE.

Turcato, D., Popowich, F., Toole, J., Fass, D., Nicholson, D., &Tisher, G. (2000, October). Adapting a synonym database to specific domains. In *Proceedings of the ACL-2000 workshop on Recent advances in natural language processing and information retrieval: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 11* (pp. 1-11). Association for Computational Linguistics.

Velardi, P., Navigli, R., Cucchiarelli, A., &Neri, F. (2005). Evaluation of Ontolearn, a methodology for automatic learning of domain. *Ontology Learning from Text: Methods, evaluation and applications*, *123*, 92.

Weichselbraun, A., Wohlgenannt, G., &Scharl, A. (2010). Refining non-taxonomic relation labels with external structured data to support ontology learning. *Data & Knowledge Engineering*, *69*(8), 763-778.

Welty, C., McGuinness, D. L., & Smith, M. K. (2004). Owl web ontology language guide. *W3C recommendation, W3C (February 2004) http://www. w3. org/TR/2004/REC-owl-guide-20040210*.

Welty, C., McGuinness, D. L., & Smith, M. K. (2004). Owl web ontology language guide. *W3C recommendation, W3C (February 2004) http://www. w3. org/TR/2004/REC-owl-guide-20040210*.

Xiao, L., Ruan, C., Yang, A., Zhang, J., & Hu, J. (2016, May). Domain ontology learning enhanced by optimized relation instance in dbpedia. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp. 1452-1456).

Yamaguchi, T. (2001, August). Acquiring Conceptual Relationships from Domain-Specific Texts. In *Workshop on Ontology Learning* (Vol. 38, pp. 69-113).

Zhao, C., Dong, C., & Zhang, X. (2018). ROCP: A rapid ontology construction platform from unstructured data. *Data Science Journal*, *17*.

Zhao, L., & Ichise, R. (2013). Integrating ontologies using ontology learning approach. *IEICE TRANSACTIONS on Information and Systems*, *96*(1), 40-50.

Zouaq, A., Gasevic, D., &Hatala, M. (2011). Unresolved Issues in Ontology Learning-Position Paper. *WS2*, 52.

ANNEXURE

Code Snippet #1 Sentence detection

```
tokenizer=nltk.data.load('tokenizers/punkt/english.pickle')
raw_sentences = tokenizer.tokenize(text)
print(raw_sentences)
```

Code Snippet #2 POS tagging of the sentences

```
for sentence in raw_sentences:
    pos_tagged=nltk.pos_tag(word_tokenize(sentence))
    print(pos_tagged)
    pos_sentence=[]
    for i in range(0,len(pos_tagged)):
         pos_sentence.append(pos_tagged[i][1])
    pos_joined_sentence=" ".join(pos_sentence)

pos_joined_sentence="("+str(line_number)+")"+pos_joined_senten
ce
```

Code Snippet #3 Corpus enhancement from Wikipedia

```
for user_keyword in user_keywords:
    try:
        text=wikipedia.summary(user_keyword)
        file.write(text)
        #file.close()
        keywords=[]
        r.extract_keywords_from_text(text)
        keywords=r.get_ranked_phrases()
        for keyword in keywords:
            try:
                text=wikipedia.summary(keyword)
                file.write(text)
                keywordsL3=[]
                r.extract_keywords_from_text(text)
                keywordsL3=r.get_ranked_phrases()
                for keyword in keywordsL3:
                    try:
                        text=wikipedia.summary(keyword)
                        file(write(text))
                        i=i+1
            except:
                        pass
            except:
                pass

    except:
```

```
        pass
```

Code Snippet #4 Identification of Nouns in the text

```
dataset=[]
for raw_sentence in raw_sentences:
    pos_tagged=nltk.pos_tag(word_tokenize(raw_sentence))
    pos_sentence=[]
    for i in range(0,len(pos_tagged)):
        if(pos_tagged[i][1]=="NN"   or   pos_tagged[i][1]=="NNS"
or pos_tagged[i][1]=="NNP" or pos_tagged[i][1]=="NNPS"):
            pos_sentence.append(pos_tagged[i][0])
    pos_joined_sentence=" ".join(pos_sentence)
    #print(pos_joined_sentence)
    dataset.append(pos_joined_sentence.split(" "))
print(dataset)
```

Code Snippet #5

 # Create the node to add to the Graph

```
soil_taxonmy = URIRef(LDT["soil taxonomy"])
# Add the OWL data to the graph
graph.add((soil_taxonmy, RDF.type, OWL.Class))
graph.add((soil_taxonmy, RDFS.subClassOf, OWL.Thing))
graph.add((soil_taxonmy, RDFS.label, Literal("text from NLP at
runtime")))
graph.add((soil_taxonmy, RDFS.comment, Literal("text from NLP
at runtime")))
# Bind the OWL and LDT name spaces
graph.bind("owl", OWL)
graph.bind("ldt", LDT)
```

Code Snippet #6

```
frequent_itemsets     =     apriori(df,     min_support=0.01,
use_colnames=True)
association_rules(frequent_itemsets,       metric="confidence",
min_threshold=0.01)
rules  =  association_rules(frequent_itemsets,  metric="lift",
min_threshold=1.2)
f.write(str(rules))
```

Code snippet #7 use of Word Net for non hierarchical class identification

```
sentences=sent_tokenize(" sentence extracted from cluster")
dataset=[]
for sentence in sentences:
    dataset.append(sentence.split(" "))
print(dataset)
for group in leaves:
    for leaf in group:
        try:
            for i,j in enumerate(wn.synsets(leaf)):
```

```
        #print('Meaning', i, 'NLTK ID', j.name())
        #print('Definition:', j.definition())
                print('Hypernyms:',                                ',
'.join(list(chain(*[l.lemma_names()       for       l       in
j.hypernyms()])))))
        except:
            pass
```

Seed for the RNG, to make the results reproducible.

```
seed = 1
thrones2vec = w2v.Word2Vec(
    sg=1,
    seed=1,
    workers=multiprocessing.cpu_count(),
    size=300,
    min_count=50,
    window=10,
    sample=1e-3
)
thrones2vec.build_vocab(sentences)
model    =    w2v.Word2Vec(sentences,    size=300,    window=5,
min_count=3, workers=multiprocessing.cpu_count())
# fit a 2d PCA model to the vectors
X = model[model.wv.vocab]
pca = PCA(n_components=2)
result = pca.fit_transform(X)

# create a scatter plot of the projection
pyplot.scatter(result[:, 0], result[:, 1])
words = list(model.wv.vocab)
for i, word in enumerate(words):
     pyplot.annotate(word, xy=(result[i, 0], result[i, 1]))
pyplot.show()
```