CONCEPTUAL KNOWLEDGE MODEL FOR IMPROVING TERM SIMILARITY IN RETRIEVAL OF WEB DOCUMENTS

KHADIJHA-KUBURAT ADEBISI **ABDULLAH**

CONCEPTUAL KNOWLEDGE MODEL FOR IMPROVING TERM SIMILARITY IN RETRIEVAL OF WEB DOCUMENTS

BY

KHADIJHA-KUBURAT ADEBISI **ABDULLAH**
M.Sc. Computer Science (Ibadan), B.Sc. (Hons.) Computer Science (Ogun)

A thesis in the Department of COMPUTER SCIENCE

Submitted to the Faculty of Science in partial fulfilment of the requirements for the Degree of

DOCTOR OF PHILOSOPHY

of the
UNIVERSITY OF IBADAN

Department of Computer Science
University of Ibadan
Ibadan

MAY, 2016.

Abstract

Terms Similarity (TS) in retrieval systems are based on lexical matching, which determines if query terms are useful and reflect the users' information need in related domains. Existing works on TS use Term Frequency-Inverse Document Frequency (TF-IDF) to determine the occurrence of terms in web documents (snippets) is incapable of capturing the problem of semantic language mismatch. This study was designed to develop a conceptual knowledge model to solve the problem of TS in web documents retrieval by amplifying structured semantic network in Multiple Document Sources (MDSs) to reduce mismatch in retrieval results.

Four hundred and forty-two IS-A hierarchy concepts were extracted from Internet using a web ontology language. These hierarchies were structured in MDSs to determine similarities. The concepts were used to formulate queries with the addition of terms from knowledge domain. Suffix Tree Clustering (STC) was adapted to cluster, structure the web and reduce dimensionality of features. The IS-A hierarchy concept on parent and child relationship was incorporated into the STC to select the best cluster, consisting of 100 snippets, four web page counts and WordNet as MDSs. Similarity was estimated on Cosine, Euclidean and Radial Basis Function (RBF) on the TF-IDF. Based on STC, TF-IDF was modified to develop Concept Weighting (CW) estimation on snippets and web page count. Similarity was estimated between TF-IDF and developed Concept Weighting; Cosine and CW-Cosine, Euclidean and CW-Euclidean and RBF and CW-RBF. Semantic network (WordNetSimilarity) LIn' measure was extended with PAth length of the taxonomy concept to develop LIPA. The LIPA was compared with other WordNetSimilarity distance measures: Jiang and Conrath (JCN) and Wu and Palmer (WUP) as well as LIn and PAth length separately. Concept Weighting and WordNetSimilarity scores were combined using machine learning techniques to leverage a robust semantic similarity score and accuracy measure using Mean Absolute Error (MAE).

The RBF and CW-RBF generated inconsistent values $(0.9 \leq x \leq 1)$ for null and zero snippets. Similarity estimation obtained on Cosine, Euclidean, CW-Cosine and CW-Euclidean were 0.881, 0.446, 0.950 and 0.964, respectively. The retrieved snippets removed irrelevant features and enhanced precisions. WordNetSimilarity JCN, WUP, LIn, PAth, and LIPA values were 0.868, 0.953, 0.995, 0.955 and 0.998, respectively.

The WordNetSimilarity improved the semantic similarity of concepts. The Concept Weighting and WordNetSimilarity; CW-Cosine, CW-Euclidean, JCN, WUP, LIn, PAth, and LIPA were combined to generate similarity coefficient scores 0.941, 0.944, 0.661, 0.928, 0.996, 0.924 and 0.998, respectively. The MAE on Cosine, Euclidean, CW-Cosine and CW Euclidean were 0.058, 0.011, 0.014 and 0.009, respectively while for JCN, WUP, LIn, PAth, and LIPA were 0.022, 0.004, 0.022, 0.019 and 0.020, respectively. The accuracy of the combined similarity for JCN, WUP, LIn, PAth, CW-Cosine, CW-Euclidean and LIPA were 0.023, 0.050, 0.008, 0.011, 0.024, 0.015 and 0.009, respectively.

The developed conceptual knowledge model improved retrieval of web documents with structured multiple document sources. This improved precision of information retrieval system and solved the problem of semantic language mismatch with robust similarity between the terms.

Keywords: Term similarity, Multiple document sources, WordNetSimilarity, Web ontology language

Word Count: 484

Acknowledgements

I give thanks to the Almighty Allah for His infinite mercy on me throughout this programme. I am grateful to my supervisors, Dr. A.B.C. Robert and Dr. A.B. Adeyemo. My interaction with both of you has been splendid. May God bless your homes and your descendants. May He continue to guide you aright. I thank Dr. Robert, especially for his immense tolerance and for scheduling meetings with me on weekends to supervise the work. And for Dr. Adeyemo, you really took your time to go through the chapters' one after the other. I appreciate both your efforts,

I cannot forget my Head of Department; Dr. (Mrs) Bolanle Oladejo and Dr. Onifade for their wonderful contributions and commitment towards the entire work. Thanks a great deal. God will richly reward you and bless your home. I appreciate the kind gesture of Dr. Akinkunmi, Dr. (Mrs.) Ayorinde, Dr. Akinola, Dr. Oladosu and other lecturers for painstakingly reading the dissertation within a short time and for encouraging me all along. Thank you all for providing an exceptionally intellectual fertile environment to undertake a PhD programme.

Okeke Emmanuel and Sodimu Segun graciously provided me with Internet facilities and some important software packages used in this research work. Also, I cannot forget my baby friend Kukoyi Adeola Khadijha for your support and words of encouragement. May God increase your wisdom and understanding and elevate all of you. I appreciate my colleagues in the Department of Mathematical Sciences, Olabisi Onabanjo University, Ago-Iwoye, Ogun State. All along, I have enjoyed your company. My association with you over the years has been rewarding. You all made the research work very fruitful. Worthy to be specially mentioned is Dr. (Mrs) Folorunso who has always inspired, challenged, and encouraged me. My sister-friend, May the Almighty God bless your descendants and keep your home. Also, highly appreciated are Prof. 'Kola Odunaike and Prof. A.O. Lawal. You encouraged me all the way. May the Almighty God bless your children. I am also grateful to Dr. (Mrs) 'Kemi Olayemi for proofreading this thesis. Thanks a great deal. May God bless your home.

I appreciate Dr. Adamu, Dr. J.A. Osilagun and Tajudeen Elder. The statistical analysis of this work shows that you have contributed greatly to my PhD research. Thanks a lot.

I am also grateful to my missionary, Sodiq AbdulRaheem. Thanks for your prayers at all times.

I am highly indebted to my parents – Prince Adebola Lamina Aroyewun and Deaconess Adedayo Aroyewun for their prayer, moral and financial support. May you live long. I cannot forget my siblings, late Ayodeji Akinlosotun (we miss you), Modupe Awoniyi, Bolanle Annefat Yusuf, Adeniyi, Adetola, Yetunde, and Adebayo Aroyewun. Thanks for your support. We shall attain greatness in life. I also thank my sister, Iyabode Mogbolu Oni. I appreciate all my nieces and nephews for their concern.

To my husband, Abdullah Olatunbosun Abdul-Majid, and my son, Mu`d Uwaiz Adetunji Oluwatomisin, I say "thank you". Words are not enough to express my gratitude to you for your boundless support and encouragement all through. You sacrificed a lot to ensure my happiness and success. Thanks my lovely ones. Both of you are dear to me!

Certification

I certify that this doctoral research work was carried out by Mrs Khadijha-Kuburat Adebisi Abdullah in the Department of Computer Science, Faculty of Science, University of Ibadan, Nigeria, under my supervision.

……………………………………………………………

Supervisor
A. B. C. Robert,
B,Eng (Minna), M.Inf.Sc. (Ibadan), Ph.D (Nancy France)
Senior Lecturer, Department of Computer Science,
University of Ibadan, Nigeria.

…………………………………………………

Co-Supervisor
A. B. Adeyemo,
B.Sc. (Ife), PGD, M.Tech, Ph.D (Akure)
Senior Lecturer, Department of Computer Science,
University of Ibadan, Nigeria.

Dedication

This thesis is dedicated to the Almighty Allah, orphans and my parents - Prince Adebola Aroyewun and Deaconess Adedayo Aroyewun. You are wonderful!

TABLE OF CONTENTS

LIST OF FIGURE

LIST OF TABLE

LIST OF NOTATIONS AND ABBREVIATIONS

C_i	a set of terms
$C \in C_i$	a term
R	a set of semantic relations
$\hat{R}$	a set of extracted semantic relations
$\hat{R} \subset$R	a subset of extracted semantic relations of type *i*
f_{ij}	the j-th feature representing the term c_i
$\lVert f \rVert$	Euclidean (L2) norm of a feature vector
$\lVert f \rVert$	Manhattan (L1) norm of a feature vector
$P(C_i)$	Probability of the *i*-th term
$S_{a,b} \in S$	a similarity score between terms/concept C_a and C_b
wt	a weight vector (w_1; : : : ;w_K)
m : n.	many-to-many relations
$Cos(a_i, b_i)$	Cosine between two feature vectors
$\chi 2$	Chi-Square
AI	Artificial Intelligence
ARFF	Attribute Relation File Format
CGI	Common Gateway Interface
Corpus-Based measure	a measure that derives similarity scores from a text corpus measure a semantic similarity measure
DISCO	DIStributionally CO-occurrences
DRM	Domain Relevance Measure
ERIC	Education Resources Information Centre
FLogic	Frame Logic
FS	Feature selection
HTML	Hyper Text Mark-up Language
Hybrid Measure	a measure that relies on several resources to derive the similarity scores
IR	Information Retrieval
IS	Information Source
Jiang and Conrath	the network-based measure of Jiang and Conrath
LCS	least common subsumer

Leacock Chodorow	the network-based measure of Leacock and Chodorow
Wu and Palmer	the network-based measure of Wu and Palmer
Lin	the network-based measure of Lin
LSA/I	Latent Semantic Analysis/ Indexing
MAE	Mean Absolute Error
MDSs	Multiple Document Sources
ML	Machine Learning
NLP	Natural Language Processing
NP	noun phrase
ODP	Open Directory Project
OWL	Web Ontology Language
OWL-DL	OWL-Description logic
POS	Part of Speech, e. g. “noun”, “verb” or “adjective
QE	Query Expansion
RBF	Radial Basis Function
RDF Schema	Resource Description Framework Schema
RDF	Resource Description Framework
RIB	Recommender Intelligent Browser
Single Measure	a measure that relies on one resource to derive the similarity scores such as a semantic network
SKlearn	SciKit-Learn
STC	Suffix Tree Clustering
SVD	Singular Value Decomposition
SW	Semantic Web
SWEET	Semantic Web for Earth and Environmental Terminology
SWOOP	Semantic Web Ontology Overview and Perusal
TC	Text Categorization
TF-IDF	Term Frequency-Inverse Document Frequency
URI/L	Uniform Resource Identifier/ Locator
WordNet	Princeton WordNet 3.0 lexical database
WSD	Word Sense Disambiguation
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

CHAPTER ONE

INTRODUCTION

1.1 Background of the Study

Similarity is an aspect of information retrieval of web documents and filtering. It is an important component of various tasks on the web such as information extraction, text mining, word sense disambiguation and document clustering. The available web documents are in different forms and the information they contain is difficult to access. This is due to the growth of web information that is so enormous. The search engine plays a more critical role in finding relation among input keywords but fails in retrieving semantically related documents. As a result, it, retrieves more irrelevant documents than needed. The attempt is to match the user's query to the source documents and present it to the user, documents that match the user keyword.

The retrieval system depends on the similarity between indexer and the queries, which is measured by comparing the values of certain attributes to indexer and user requests. The indexers and the user do not always use the same terms because synonymy terms fail to retrieve relevant documents with a decrease in recall. Subsequently, polysemy causes retrieval of irrelevant documents, which implies a decrease in precision retrieval. Therefore, Terms Similarity (TS) in retrieval systems is based on lexical matching, which determines if query terms are useful and reflect the users' information need in related domains. Most approaches developed to enhance Term Similarity in retrieval of web documents were based on a single information source. In such approaches, there is representation of documents in a linear feature vector in which similarity or relation among features is considered without context or structure.

Finding appropriate information sources that contain the data for computing similarity or relatedness of terms in a more structured way than only a single source is necessary. The idea of using web resource and other structured information sources was intended for computing semantic similarity of terms in retrieval of web documents.

The traditional retrieval systems have limited abilities to exploit the conceptualisations involved in user needs and content meanings due to inability to describe the relation among search terms. The search engines are keyword based which have not bridged the gap of vocabulary mismatch problem in retrieval system. The word mismatch is a problem in the usage of natural language (Croft *et al.*, 2010). Language mismatch and ambiguity of words in documents' repository on web content causes difficulties in retrieving relevant documents in related domains (Alipanah *et al.* 2010; Rinaldi, 2009 and Lee and Soo, 2005). User request must be understandable by the retrieval system to avoid mismatch of terms because query may reflect multiple domain of interest. Also, different researchers in the same field name the same term differently (Landauer and Dumais, 1997) but this poses difficulty in text or large database expressing the same concept (Voorhees, 1994).

The goal of semantic indexing is to use semantic information to improve the quality of information retrieval; unlike the traditional indexing methods that are based on keyword matching. The use of semantic indexing is based on the hypothesis that a document is viewed as a set of concept. However, the importance of a concept depends on the number of links with other concepts that share the same document. The Information Retrieval System (IRS) needs to focus on using additional knowledge in order to retrieve relevance of IRS. Consequently, this knowledge is also used to index and describe the content of documents. The idea is that high-level semantic content information is accurately modelled using conceptual indexing so that related documents that do not share terms are still represented by nearby conceptual descriptors (Berry *et al.*, 1999). In order to compare the similarity between the resources and the queries, both need to be represented in a compatible way. This makes it possible to automate the process of calculating the relevance between queries and resources. The concept of Semantic Web (Berners-Lee *et al.*, 2001) manually or automatically constructs taxonomy of semantic concepts and its relations and is also used to map documents and queries. This outlines how information is meaningful and brings context not only to humans but also to machines. The most vital tool in searching for information and related resources in a Semantic Web (SW) is the ontology.

Ontology represents knowledge that could be understood by machine, used and shared among distributed applications to improve knowledge management systems. It is used in information retrieval (Egozi *et al*., 2011), for query expansion, indexing and retrieval (Carpineto and Romano, 2012). Jian-liang *et al*. (2009) presented domain ontology that represents knowledge that can be shared and re-used. Therefore, ontology captures the semantic relationship between concepts or vocabulary used in a particular domain which discovers relationships between descriptions of entities.

In order to overcome the limitations of existing web search systems and difficulty of keywords search engines, queries need to be represented with context through ontology structure for effective search (Khan and Marvon, 2006) but Sahami and Heilman (2006) suggested a web-kernel function that expanded the text by issuing it to a search engine as the query. The system concentrates on searching the ontology structure and not on the individual keywords or terms. Moreover, concepts from the domain ontology that are semantically structured are used to distinguish the structure in any given natural language documents. The ontology structure also known as IS-A hierarchy is used to access information from web instead of keywords from terms. Therefore, the hierarchies are used to describe the structures of documents and search queries. The queries are formulated using concepts of the ontology. But they expand with terms from domain of reference to give the queries a better meaning. Such knowledge is used to enhance the precision and recall of information retrieval. Since query vocabulary has been controlled, the web resource needs to be structured as well to improve search term.

Information is organised in a way that makes it easier for the end users to find the information efficiently and accurately (Oikonomakou and Vazirgiannis, 2005). But Jain *et al*. (1999) discovered that documents are grouped (cluster) together with similar or related documents for easy search but there is a need to optimise the search engine. Therefore, Eissen *et al.*, (2005) and Chim and Deng, (2008) described a better way to achieve more accurate document clustering based on phrase that is more informative and semantic representation than feature term clustering. A Suffix Tree Clustering (STC) algorithm is adapted to group the input texts according to the identical phrases (Hammouda and Kamel, 2004). The web resource is specified with a particular page number (refers to multiple document sources) to reduce dimensionality of features. But due to succinctness of natural language, words can represent multiple concepts and different terms may represent the same or similar concepts. In disambiguating terms that occurred in natural language, the available context information from ontology is used to extract concept description from the clusters that match query (Navigli and Ponzetto, 2010). The clusters that relate to the concepts' description are chosen for further processing.

The retrieved Multiple Document Sources (MDSs) from clusters are filtered (preprocessed) by removing unwanted words such as stopwords and stemming appropriate terms with their lemma. A WordNet-based lemmatisation that belongs to the group of dictionary algorithms is used to reduce words that are not in stem forms to their corresponding lemma. These preprocessed documents are ranked based on the similarity which was attained by means of the extracted domain concepts hierarchy. A concept weighting estimation model is developed to determine the feature vector of concepts in the retrieved MDSs. This reduces the features and normalises the similarity weight (*wt*) values between $0 \leq wt \leq 1$ for the values to be based on a scale.

The Multiple Document Sources (MDSs) from web as information source is not sufficient to determine the semantic implication of term similarity due to the diversity of words. This implies there is a need for additional knowledge information source to adjust and add meaning to the similarity. The knowledge source (semantic network) requires higher accuracy of the semantic similarity of terms with the help of conceptual knowledge. A hybrid measure is developed combining information content from knowledge source with path length in the taxonomy. Thus, similarity is imparted with a semantic meaning to solve the language mismatch. Two concepts from related domain searched with the documents retrieved (D_a and D_b) are semantically related if the similarity function maps to a real-valued number between $0.5 \leq wt \leq 1$ which has a higher value when measured. Therefore, similarity between documents retrieved from related domains queries is determined.

The MDSs and semantic network information sources are integrated forming conceptual knowledge model using machine learning techniques. This solves the problem of semantic language mismatch of terms in related or overlapping domains in retrieval system.

1.2 Statement of the Problem

If users use different synonymous terms as query, the information or documents retrieved are not always similar. This is because the current web lacks semantic meaning?. Therefore, similarity is important in retrieval of web document. But most similarity techniques that have been used in information retrieval systems do not consider the semantic of terms in retrieval of web document in related domains. However, one or more information sources were used in these techniques (Meng *et al.*, (2014); Prathvi and Ravishankar, (2013) and Bollegala *et al*, (2007)). Those that used two information sources were based on the same source or two different structured corpora, for instance, latent semantic indexing and knowledge sources. These are structured texts and not unstructured large database as web documents (Mihalcea *et al.*, (2006) and Nitish *et al.*, (2012)). The most difficult part of retrieval system is when indexers analyse the content of a given document in two different ways resulting in two different index entries.

Information source by multiple document collection (web resources) used term frequency-inverse document frequency (tf-idf) to determine the weight of term vectors' occurrence in form of

$$tf - idf = tf_{i,j} * \log\left(\frac{|D|}{df(j)+1}\right) \qquad (1.1)$$

It has been argued that *tf-idf* is not directly derived from a mathematical model of term distribution or relevancy analysis. But it was derived from the theory of language modelling where the terms in a given document are divided into with and without the property of eliteness (Roberston, 2004). However, the *idf* measures the importance of term but there is effect on *idf* for terms that do not occur in the document training sets in 1.1. The *tf-idf* is based on bag-of-words (BOW) because it does not consider the ordering of words. This reduces the weight of the terms occurrence and increases the dimensionality of features while the similarity metrics are lexically based (Christopher and Hinrich, 2001).

Although, query is not consistent with vocabulary used in multiple document collections but it has been observed by information retrieval researchers that indexing tends to be more consistent when the vocabulary used in the query is controlled. The indexers are more likely to agree on the terms needed to describe a particular context. Therefore, query can be modelled using ontology structure. However, the web needs to be structured as well (clustering web document) to improve the search result. Consequently, using semantic network for similarity controls the vocabulary and this involves the semantic meaning of two concepts or terms in WordNet. But most existing semantic networks use a single similarity method with one weakness or the other (Pirrò, (2009); Mihalcea, (2006)). Semantic network similarity on Information Content (IC) does not consider the path length of taxonomy in WordNetSimilarity. However, similarity methods that consider the position of concepts in the taxonomy perform better than only path length. (Li *et al*., 2003). Furthermore, exploiting the information content with structure of the taxonomy also performs better than hybrid and feature based method (Rodriguez and Egenhofer, 2003). Although, the knowledge source method has the advantage to be fast and makes it possible to have a reusable resource even though the corpus changes. Its drawback is the possibility to omit some concepts with different forms that appeared in the source text and in the ontology.

High level semantic contents are modelled using conceptual indexing. A hyrid conceptual knowledge model is developed in which similarity is based on context. This overcomes the problem of language mismatch to some extent. The research work adjusts multiple document sources with semantic network to solve the language mismatch in retrieval of relevant documents in related domains. The similarity level is evaluated using different machine learning techniques to determine the similarity coefficient.

1.3 Research aim and objectives

This study was designed to develop a conceptual knowledge model to solve the problem of term similarity in web documents retrieval by amplifying structured semantic network in Multiple Document Sources (MDSs) to reduce mismatch in retrieval results.

The objectives are as follows:

i. Extract ontology hierarchy concepts for querying the search engine.

ii. Develop adaptation suffix tree for clustering and structuring search result

iii. Reduce the dimensionality of features by developing a concept weighting model for the similarity process.

iv. Developing a model that uses a web based interface which incorporates length of taxonomy into the information content to optimise the semantic similarity in WordNet.

v. Integration of the data from the sources into machine learning techniques to determine the level of similarity correlation coefficient of Term Similarity.

1.4 Significance of the Study

Recent technology in web search should enable machine to understand user' request and respond to request based on the context. The understanding requires that relevant information sources be semantically structured. The algorithm and model developed in this study will improve the search result of web document retrieved. As a result, this would minimise irrelevant documents, reduce time constraint of the users and enhance precision and recall.

1.5 Scope and Limitation

The approaches and techniques of semantic information retrieval are based on many-to-many relationship (m:n) between different terms used in searching web documents from related domain. The development of concept weighting estimation for the feature vector construction is leveraged with semantic. However, an extension of information content similarity with the path length in WordNetSimilarity is also considered. Although, the semantic similarity between terms or concepts in document collection and semantic network is not automatic because it requires human intervention; therefore, the approach is semi-automatic.

1.6 Thesis Organisation

The remaining part of the thesis is organised into four sections. Chapter Two is centred on literature review; it provides an overview of related work with a focus on the basic concepts of information retrieval, Semantic Web and its goal to provide basic understanding of ontologies and ontological representations. Also, it includes document indexing, search clustering techniques and query expansion methods. Similarity measures methods were also considered in terms of knowledge methods with different information sources. In Chapter Three, the research methodology is presented on research phases and tasks. Furthermore, the research methods were used to achieve the objectives. Chapter Four presents the results of the work, discussion and evaluation of the results of the work presented in chapter three. The objectives and the research questions are revisited. The research questions with regard to the results are evaluated and hence the contribution of this work is achieved. Finally, Chapter Five discusses the summary, conclusion and recommendations.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

There has been a tremendous growth in documents information especially on the web. Researchers have begun to delve into the potential of associating web content with explicit meaning so as to create a web with meaning. This chapter introduces the concept of information retrieval and the basic concepts of Semantic Web. The goal of semantic web to provide basic understanding of ontologies is the basis of this research work. Rather than rely on natural language processing (NLP) to extract the meaning from existing documents, the approach requires describing the use of ontology. Ontology is described in different language knowledge representations. Different search result clusterings are also explored to provide solution to the problem with ranked ordering so that users can find relevant documents easily. However, users are provided with the options to select the minimum frequency to be considered as well as the maximum number of words in a term. Different domain applications or uses of ontologies are also considered. Different similarity measures and information sources are discussed both in document collections and knowledge sources in order to increase the coverage of different sources and some related literatures.

2.2 The Concept of Information Retrieval

Sparck and Willet (1997) invented information retrieval in 1952 and gained popularity in the research communities in 1961. Therefore, information retrieval organises function in libraries that are no longer just storehouses of books but as places where information is catalogued and indexed. However, the concept of information retrieval presupposes that there are some documents or records containing information that has been organised in an order suitable for easy retrieval.

An information retrieval system was designed to retrieve documents or information required by the user. Thus, an information retrieval system aims at collecting and organising information in one or more subject areas in order to provide answer to the query of the users. Lancaster (1968) as cited by Chowdhury and Chowdhury (2002) commented that an information retrieval system does change the knowledge of the user on the subject of enquiry. The information retrieval system however serves as a bridge between the generation of information and the users of that information. There are two categories of information retrieval systems that have been identified. These are:

i. In-house Information Retrieval Systems: These are set up by a particular library or information centre to serve mainly the users within the organisation. An example is the database of catalogue. Online public access catalogue (OPAC) provides facilities for library users to carry out online catalogue searches and check the availability of the item required.

ii. Online Information Retrieval Systems: These are designed to provide access to remote and different collections of databases for a variety of users. Such services are available mostly on commercial basis, academic research and there are a number of vendors that handle the services.

This thesis is more concerned about the online information retrieval system that is the web. Web (Internet) resources vary significantly in terms of their content (text, numeric, audio, image video, etc.), file format availability and URL (Uniform Resource Locator) or the address of a web page. There are rules and guidelines required to help make information retrieval easy and effective. Schwartz (2001) mentioned the term "metadata" which is used primarily in the field of database management. Metadata has been classified into the following based on its use:

i. Metadata based on administrative was used in managing and administering information resources.
ii. Metadata based on descriptive was used to describe or identify information.

iii. Metadata based on resources preservation was related to the preservation and management of information resources

iv. Used metadata was related to the level and types of use of information resources.

Moreso, metadata supports a variety of operations and the users of the metadata are either human or computer programmes. But the primary functions of metadata are to facilitate the identification, location, retrieval, manipulation and use of digital objects in networked environments. It has become an important issue in information organisation since the advent of the Internet and the web.

Xin (1990) wrote on the ideal document retrieval environment. A query statement is represented by a group of distinct index terms as well as the semantic relationship between these terms, so that retrieval could be based on the structured of semantic relationship. Moreover, documents are retrieved on the basis of the correspondence between search terms expressed in the query and the index terms in the document. Indexing systems designed to assist in the retrieval of documents operate by assigning index terms to the analysed subject of each document either manually or automatically.

The most difficult part of retrieval system is where two different indexers analyse the content of a given document in two different ways resulting in two different index entries. However, it has been observed by information retrieval experts that indexing tends to be more consistent when the vocabulary used is controlled because indexers are more likely to agree on the terms needed to describe a particular context. The different kinds of vocabulary control tools have been introduced. Examples are the thesauri and WordNet. These are part of the major concern of this research work.

Rowley (1994) defined the thesaurus as a compilation of words and phrases showing synonyms of hierarchical, relationships and dependencies, the function of which is to provide a standardised vocabulary for information systems. The thesaurus is to exert terminology control in indexing and to aid in searching by alerting the searcher to the index terms that have been applied.

Other advantages of automatic indexing are the maintenance of consistency in indexing, indexing time is saved, index entries are produced at a lower cost and better retrieval effectiveness is achieved.

In conclusion, Chowdhury and Chowdhury (2002) evaluated an information retrieval system as a measure of effectiveness and efficiency. Effectiveness may be a measure of how far it can retrieve relevant information while withholding non-relevant information but efficiency is how economically the system is achieving its objectives.

2.3 Semantic Web

The potentials of World Wide Web (WWW) as an information source are relatively untapped because it is difficult for machines to process and integrate information meaningfully. In response to this problem, many new research initiatives have been set up to enrich available information with machine-processable semantics. Tim Berners-Lee (2001), Director of the World Wide Web Consortium (W3C), referred to the future of the WWW as the "*Semantic Web*". This is an extended web of machine-readable information and automated services. The Semantic Web is an emerging research area which builds on the foundations of diverse prior work, that is, Semantic Web was built on top of the existing web. It is important to have a clear understanding of existing web standards and anticipate how the Semantic Web will interact with other web technologies as described by the World Wide Web Consortium (W3C) using standardised languages.

Semantic Web (SW) data cannot explicitly describe the knowledge content in Hyper Text Mark-up Language (HTML) pages. But Semantic Web can specify the implicit information contained in the web and the usable representation of inaccessible database or other resource. Consequently, information must be provided in such a way that computers can understand it. To grow with the realisation of the SW vision, the SW technologies have been developed. These form part of the SW layers which are illustrated in Figure 2.1 below. The descriptive information made available by these languages allows for characterising individually and precisely, the type and the relationships between the resources in the web.

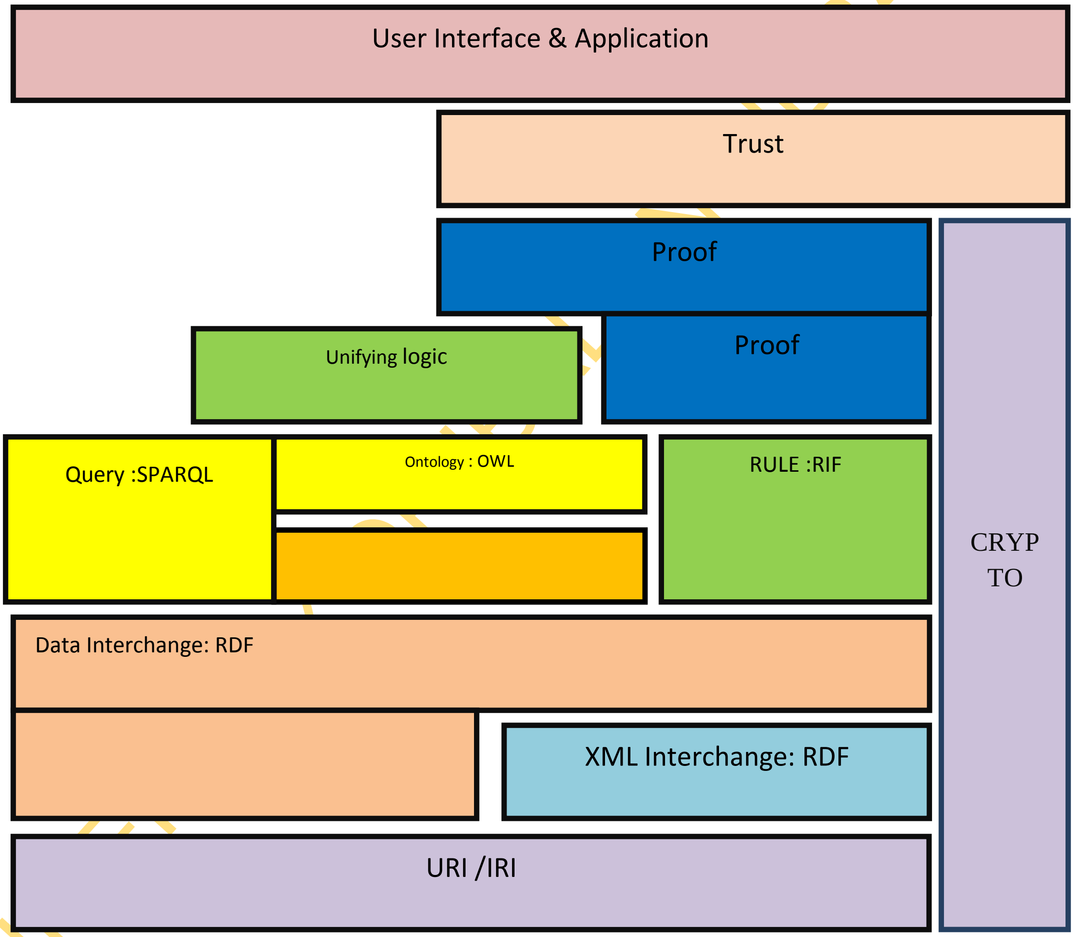


Figure 2.1: Semantic Web Stack (adapted by Berners-Lee *et al*. 2006)

The Semantic Web consists of two formats. The first includes common formats for the integration and combination of data drawn from diverse sources while the original web mainly concentrates on the interchange of documents. The second involves the language recording how the data relates to real world objects. This allows a person or a machine to start in one database and then move through an endless set of databases which is connected by being about the same thing not by internet. Finally, the Semantic Web aims at defining ways to allow web information to be used by computers for interoperability and integration purposes between systems and applications. Moreover, the formal logical model to represent knowledge in such description is ontology.

2.3.1 Ontology

The term “ontology” is borrowed from philosophy, where it refers to a systematic account of what can exist or “be” in the world. In the fields of computer science, ontology is referred to as a formal specification of the concepts of an interest domain, where the relationships, constraints and axioms are expressed, thus defining a common vocabulary for sharing knowledge. A well-known definition of the ontology was given by Gruber (1993) as "a set of representational primitives with which to model a domain of knowledge or discourse".

Ontologies are used in the fields of the computer science as Artificial Intelligence (AI), software engineering, Semantic Web (SW) and Natural Language Processing (NLP). The objective of using ontologies is to share knowledge between computers or computers and humans. It builds upon a hierarchical structure and is categorised into two layers: These are Upper ontology and Domain ontology.

i. *Upper Ontology:* SUMO (Pease and Niles, 2002) and CyC (Lenat, 1995) are examples of ontologies that describe the most general entities (concepts, properties and relationship). This subsumes a large number of more specific concepts and contains very generic specifications that serve as a foundation for specialisations.

ii. *Domain Ontology:* This describes subject domain, entities and relations of a specific domain and expresses directly the texts it belongs to or the text from which it is extracted. It offers the highest levels of both formalisation and semantic expressiveness. It formally models a domain of interest by the definition of classes and its semantic interrelations. But the ontology class structure is not necessarily linear. It allows classes to have more than one superclass.

As far as relations are concerned, ontology offers a superior level of expressiveness than thesauri. However, all imaginable semantic relations are defined between all kinds of objects in ontology as shown in Figure 2.1 below. Ontology is far more expressive than topic maps.

2.3.2 Knowledge Structures

There are some other knowledge structures that exist varying from the levels of formalisation and semantic expressiveness. As shown in Figure 2.2, the higher the level of formalisation, the higher is the semantic expressiveness. Taxonomies show the lowest level of both formalisation and semantic expressiveness while ontologies have the highest levels of formalisation and semantic expressiveness.

i. *Taxonomy*

Taxonomy and ontology are occasionally used interchangeably. Taxonomies are collections of entities ordered by a classification scheme for a certain domain and are usually arranged hierarchically in a linear structure. Each category is assigned to maximally one super-ordinate category, forming a tree structure of category hierarchies. An example of taxonomy is the Open Directory Project (ODP).

ii. *Thesaurus*

Burkart *et al.* (2004) describe thesaurus as controlled vocabularies which are usually developed for the purposes of document indexing and retrieval. Compared to taxonomies, it offers a higher level of expressiveness by allowing a structured vocabulary of a certain domain. This is not only in a hierarchical order but also by a set of predefined semantic relations. Burkart *et al.* (2004) represent relations between compound terms and their components in a thesaurus. A popular thesaurus from the domain of education is the English-language ERIC Thesaurus of the Education Resources Information Centre.

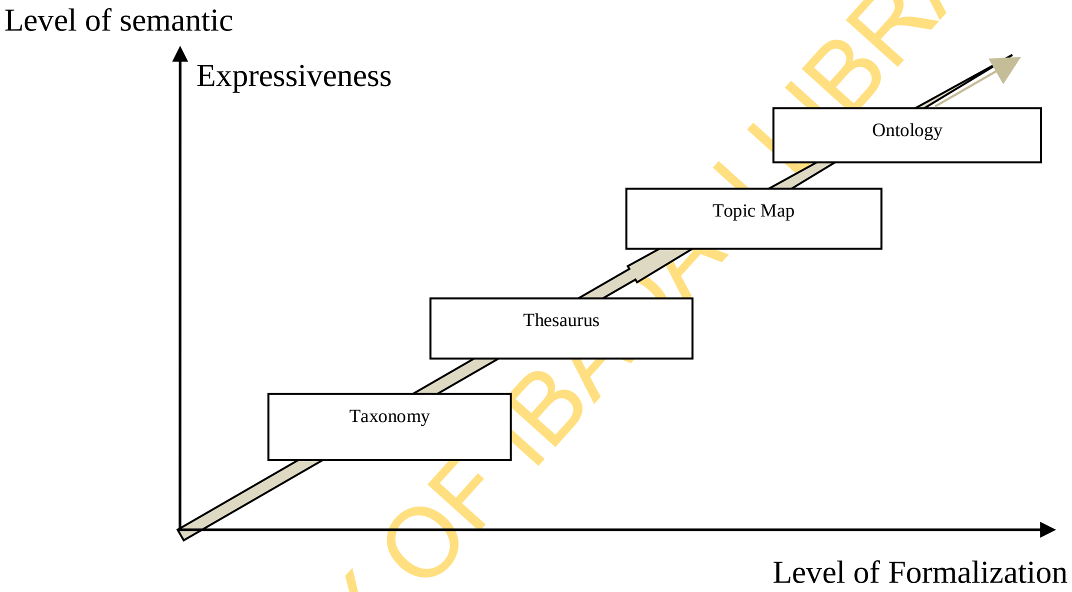


Figure 2.2: Ontology and Knowledge Structures (Ullrich *et al.* 2003)

iii. Topic Map

The topic map is used to simplify the exchange of knowledge structures on the web standard. It adds a level of complexity to classification systems and controlled vocabularies such as thesauri and taxonomies. It differentiates between abstract subjects and real representations and occurrences in documents which are referred to as topics. It also offers similar possibilities as ontologies which distinguish between abstract classes and their concrete instantiations. However, the standard does not allow for drawing inferences on the represented knowledge and no formal query languages are available for querying topic maps.

2.3.3 Types of Ontology

Ontologies are classified into three categories based on their strengths and weaknesses. These categories are explained below:

i. Formal Ontologies

In logic, formal ontology supports complex inferences and computations and has a conceptualisation structure that is distinguished by axioms and definitions. It directly induces an inference mechanism and specific properties of entities which are derived when needed. A drawback is the high effort of encoding and the danger of running into inconsistencies. Therefore, exact interference may become intractable in large formal ontologies.

ii. Prototype-Based Ontologies

This is distinguished by typical instances or prototypes rather than by axioms and definitions in logic. Categories are formed by collecting instances extensionally and selecting the most typical members for description. For its selection, a similarity metric on instance terms has to be defined. The disadvantage of the prototype-based ontology is the absence of concept labels which makes it impossible to answer queries.

iii. Terminological Ontologies

These are partially specified by subtype-supertype relations and describe concepts by concept labels or synonyms rather than by prototypical instances but lack an axiomatic grounding. A well-known example of a terminological ontology is WordNet (Miller, 1995).

A terminological and prototype-based ontology cannot be used in a straightforward way for inference but is easier to construct and to maintain. Due to the absense of concept label during construction, these are directly induced by term clustering and therefore easier to construct but less utilisable than terminological ontologies.

2.3.4 Ontology Language Representation

The use of ontology for different purposes in the context of information retrieval is based on the nature of the ontology used. Ontologies are implemented in a great variety of languages. The most representative languages are XML which has been adopted as a standard language used to exchange information on the web and along with some other languages as shown in the Figure 2.3.

i. XML

eXtensible Markup Language (XML) is used as data exchange format in different domains. It allows different parties to exchange data by providing common understanding of the basic concepts in the domain. Shabo (2006) described the syntactic level of XML but this lacks support for reasoning (semantics). Thus, problems arise when it is necessary to manipulate and integrate different XML data sources; therefore, organisations are shifting from a syntactic to a semantic level. Ontologies are necessary to express the semantics of the data. The data sources are heterogeneous in syntax, schema, or semantics thus making data communication a difficult task. Syntactic heterogeneity is caused by the use of different models or languages. Schematic heterogeneity results from structural differences and is caused by different meanings or interpretations of data in various contexts. In implementing ontology, several languages have been created based on XML. These are discussed below:

ii. URI and Unicode

The Semantic Web is generally built on syntaxes which use Uniform Resource Identifier (URI) to represent data, usually in triples-based structures, that is many triples of URI data that can be held in databases or interchanged on the World Wide Web using a set of particular syntaxes developed especially for the task. These syntaxes are called "Resource Description Framework" syntaxes. Unicode allows supporting the international text style standard.

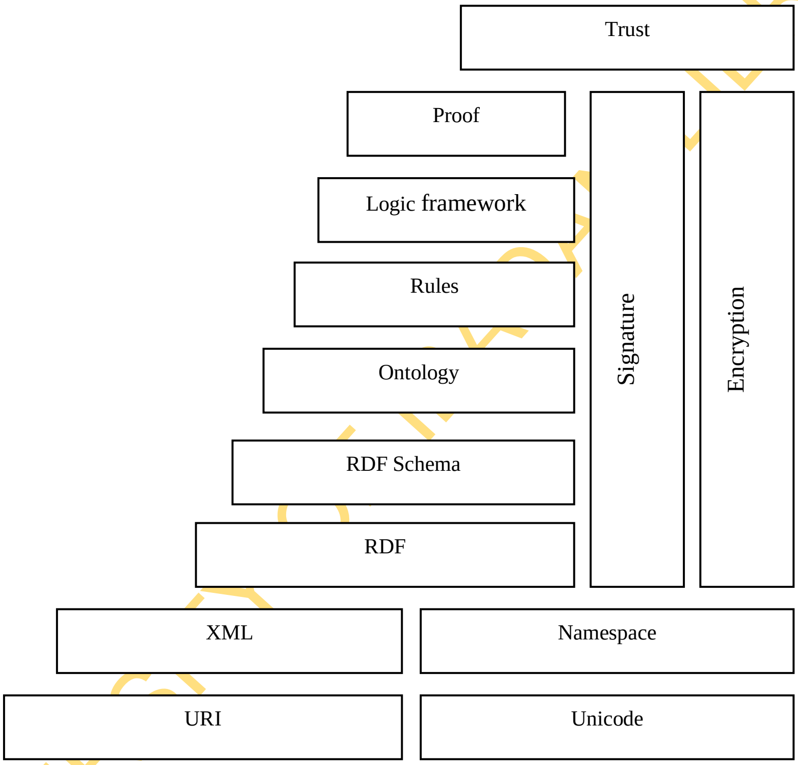


Figure 2.3: Semantic Web Layer Data Representation Standards

iii. *RDF* **(**Resource Description Framework)

Bray (2004) described XML meta-language as a standard based on meaning to map the information directly and unambiguously to a model. For processing metadata (data about data), Lassila and Swick (1999) developed the concept of RDF model which is used in a standardised way. It uses metadata format that permits to reason about data. It is used to capture and state the conceptual structure of information offered in the Web.

The RDF assertions (triples) of URIs are viewed as a data model for describing machine processable semantics of data to build the infrastructure for that which Tim Berners-Lee, the creator of WWW space, called the Semantic Web (Berners-Lee *et al*., 2001). Gil *et al*., (2005) and Daconta *et al.*, (2003) suggested that to gain benefit of the full potentials of the Semantic Web, the main idea is to publish data as RDF, a common data annotation and representation.

iv. SPARQL

Data are accessed in the form of RDF triples in ontological knowledge bases. The SPARQL syntax is similar to that of the SQL query language for relational databases as the SELECT and WHERE clauses are employed to query data from an RDF graph. SPARQL queries are similar to the triple-form of RDF statements, except that each subject, predicate or object in the SPARQL query may consist of a variable.

v. RDF Schema

Brickley and Guha, (2000) developed a simple data-typing model for RDF which is RDFS which can model simple ontologies. Web resources process class hierarchies and properties with ranges and domains. This allows the quickly building up of knowledge databases in RDF. RDFs also contain a set of properties for annotating schemata, providing comments and labels and making them easy to be understood.

vi. FLogic

Kifer *et al*., (1995) introduce Frame Logic (FLogic) which provides a semantically founded knowledge representation based on the frame-and-slot metaphor. Another formalism that fits well with the structure of RDF is Conceptual Graphs. Corcho

(2001) provided a visual metaphor for representing the conceptual structure, where languages receive the name “classic languages”. It follows a syntax based on LISP (to the exception of FLogic).

vii. *Web Ontology Language (OWL)*

Dean and Schreiber (2003) introduce a more recent Web Ontology Language (OWL) which has become a popular standard for data representation and exchange and it is the language recommended by the W3C. The OWL supports the representation of domain knowledge using classes, properties and instances of the use in a distributed environment as the World Wide Web. OWL includes three sub languages discussed below:

i. *OWL-Lite:* It supports those users who primarily need a classification hierarchy and simple constraint features. For example, while OWL Lite supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and provide a quick migration path for thesauri and other taxonomies.

ii. *OWL-DL:* OWL-Description logic (DLs) is the popular framework and it is the first order logic which aims at being expressive while retaining computational completeness. That is, all conclusions are guaranteed to be computed and decidable (all computations will finish in finite time). Baader et al. (2003) suggested OWL which influences quite a number of sources but its main representational facilities are directly based on Description Logics. OWL-DL provides a compromise supported by reasonably efficient reasoners and a language that can express large classes of ontologies and knowledge. Due to these advantages over others OWL language is used as language representation for the two domains in this research work.

iii. *OWL Full:* This is used by users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full, a class is treated simultaneously as a collection of individuals and as an individual in its own right. Another significant difference between OWL-DL and OWL Full is that owl:DatatypeProperty is marked as an owl:InverseFunctionalProperty. OWL Full allows an ontology to augment the meaning of the pre-defined RDF or OWL vocabulary. It is unlikely that any reasoning software will be able to support every feature of OWL Full.

2.3.5 Reasoning with OWL

Horrocks and Patel-Schneider (2004) provided an overview of the OWL Description Logic (OWL-DL). OWL-DL is a syntactic variant of the SHOIN (D) description logic. Although several syntaxes for OWL-DL exist, the traditional description logic notation was used since it is more compact and consistent. An ontology O is consistent if and only if O is satisfiable, that is, if ontology O has a model. To be able to define queries with domain ontologies, it relied on the notion of entailment: ontology O entails α to denote that the ontology O entails the axiom α (alternatively, is a consequence of the ontology O), if and only if α holds in any model in which ontology O holds.

Ontology learning enables obtaining the required formal representations of the knowledge available in the corpus or lexical database to be able to support such advanced types of search. To automatically learn ontologies to enhance search such ontology should be able to support queries.

2.3.6 Ontology Development

With the explosion of the amount of electronic data on the Web, the ability of creating conceptual models from textual data is a key issue for the current Semantic Web (SW) and Artificial Intelligence (AI) research. The Semantic Web relies heavily on domain ontologies as conceptual models which aim at making machines able to interpret the actual web content. However, a well-known problem of the Semantic Web is the knowledge acquisition bottleneck. This results from the difficulty of manually building or developing domain ontologies and making them evolve to reflect the actual data content. Manual acquisition of ontologies is a tedious and cumbersome task. It requires an extended knowledge of a domain and in most cases the result could be incomplete or inaccurate. Gomez-Perez *et al.* (2003) explained that building ontologies manually is expensive, biased towards developer, inflexible and specific to the purpose that motivated its construction.

However, the manual ontology created is almost always at a higher level of semantic richness. For this reason, when semantic richness is the goal, manual approaches are preferred to automatic approaches. Researchers try to overcome these disadvantages of manually building ontology by using semi-automatic or automatic methods in the building process. This deals with huge amounts of data to help speed up the manual ontology creation process. Sabou *et al.* (2005) suggested that automation of ontology construction does not only reduce costs but also results in an ontology that better matches its application. Buitelaar *et al*. (2005) organised the aspects and tasks involved in ontology development into a set of layers. Such layers are shown in Figure 2.4. In this research work, semi-automatic tools and methods are presented.

Two main methods exist in semi-automatic ontology development. The first aids ontology construction by providing tools, including editors, consistency checkers, mediators to support shared decisions, and ontology import tools. The second relies on machine learning and automated language processing techniques to extract concepts and ontological relations from structured and unstructured data such as databases and texts or more precisely it relies on ontology learning. There are two (2) ways of ontology developments. These are:

2.3.6.1 Ontology Creation From Editors

Ontology creation from editors is used by knowledge engineers or domain experts to build the ontology from the scratch or to build from existing ontology. Examples of such are discussed below:

i. *Apollo:* These are user-friendly knowledge modelling applications which allow users to model ontology with basic primitives such as classes, instances, functions, relations and so on. Uhlir *et al.* (2003) used Apollo for ontology creation but this does not support graph view, web information extraction and multi-user capabilities. It has strong features consistency checking and stores the ontologies in form of import/export format. Apollo is implemented in Java.

ii. *OntoStudio:* It is an Ontology Engineering Environment supporting the development and maintenance of ontologies by using graphical means. Cardoso (2007) described an environment that allows creating, browsing, maintaining and

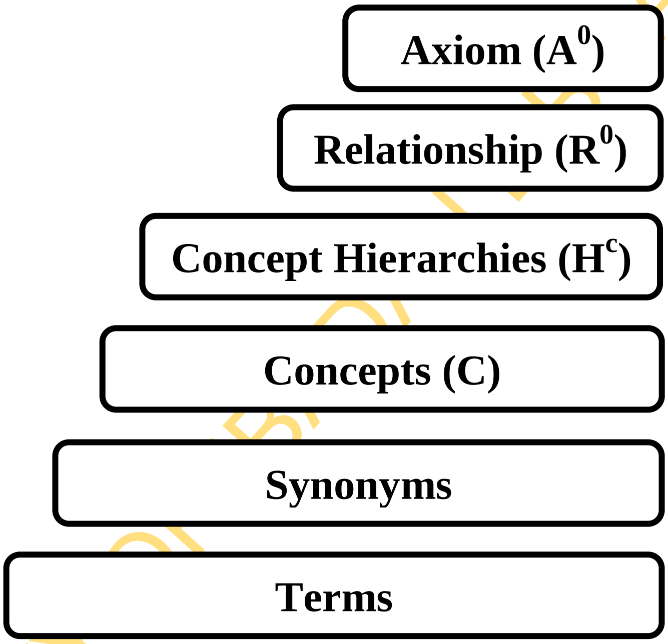


Figure 2.4: Layers of the Ontology Development Process (adapted from Buitelaar *et al.* 2005).

managing ontologies. OntoStudio is built on top of a powerful internal ontology model. The tool allows the user to edit a hierarchy of concepts or classes.

iii. *Protégé Ontology Editor:* Fergerson *et al.* (2004) developed a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. It implements a rich set of knowledge modelling structures and action that support the creation, visualization and manipulation of ontologies in various representation formats. Noy and Musen (2004) presented Protégé-OWL for reasoning API that can access an external DIG-compliant reasoner which enables the inferences about classes and individuals in ontology. The significant advantage of Protégé is its scalability and extensibility (Kapoor and Sharma, 2010). In this research work, Protégé 4.2 is used to construct domain models and knowledge-based of existing ontologies.

iv. *Swoop (Semantic Web Ontology Overview and Perusal):* Kalyanpur *et al.* (2005) developed a simple, scalable, hypermedia-inspired OWL ontology browser and editor written in Java. Swoop contains OWL validation and offers various OWL presentation syntax views (Abstract Syntax, N3 etc). Swoop does not follow a methodology for ontology construction and it is not possible to do partial imports of OWL.

v.

vi. *OntoEdit:* Sure *et al.* (2002) developed environment for ontology design and maintenance. It supports multilingual development, and the knowledge model is related to frame-based languages. It provides other features to deal with the requirements an ontology engineer has.

vii. *OilEd:* Davies *et al.* (2003) built ontologies using DAML+OIL for allowing the user to inspire the actual OWL. The basic design has been closely influenced by similar tools such as Protégé and OntoStudio. OilEd inherits only the main facilities, the rest being a little bit restricted. .

viii. *WebODE:* Arpírez *et al.* (2001**)** developed an ontological engineering workbench that provides varied ontology, related services and gives support to most of the activities involved in the ontology development process and in the ontology usage. Fernández-López *et al.* (1999) represented ontology in expressive knowledge model based on the reference set of intermediate representations of the METHONTOLOGY methodology.

In summary, a lot of similar ontology development tools (editors) do exist for the building of ontology. However, one major drawback of these methodologies is the huge amount of time and effort required by humans called “knowledge acquisition bottleneck”. This situation is even worsened when it comes to ontology evolution or mapping.

With the rapidly growing amounts of electronic data, providing (semi) automatic knowledge extraction tools is a must, to help speed up the manual ontology creation by editors. In order to reduce the effort in the design and development of ontologies, this thesis presents automatic extraction of each of the ontological components from domain texts.

2.3.6.2 Ontology Creation using Learning Techniques

The problem that ontology learning deals with is the knowledge acquisition bottleneck, which is the difficulty to actually model the knowledge relevant to the domain of interest. Gomez-Pere and Manzanor-Macho (2003) defined ontology learning as the set of methods and techniques used for building ontology from the scratch and enriching, or adapting an existing ontology in a semiautomatic fashion using several sources.

Shamsfard and Barforoush (2003) defined ontology learning as extracting ontological elements (conceptual knowledge) from input and building ontology from them. It aims at semi-automatically or automatically building ontologies from a given text corpus or other sources with a limited human intervention. Ontology learning uses methods from a diverse spectrum of fields such as machine learning, knowledge acquisition, natural-language processing, information retrieval, artificial intelligence, reasoning and database management (Sabou *et al*. 2005). The automatic or semi-automatic support for the instantiation of a given ontology is referred to as ontology population (Buitelaar *et al*. 2005).

According to Benz and Hotto, (2007) there are two fundamental aspects on ontology learning:

i. The availability of prior knowledge: The learning process is performed from scratch or some prior knowledge is available. Such prior knowledge is needed in the construction of a first version of the ontology. Thus, a source of prior knowledge demands little effort to be transformed into the first version of the ontology. This version is then extended automatically through learning procedures and manually by a knowledge engineer.

There are inputs used by the learning process. Benz and Hotto (2007) define three different kinds as:

i. Structured Data e.g. Database schemes.
ii. Semi-Structured Data e.g. Dictionaries like WordNet (Miller,1995)
iii. Unstructured Data e.g. Natural language text documents.

Some systems build ontology from scratch (Sabou *et al.*, 2005; Karoui *et al.* 2004) while others used keywords for the representation of specific domain (Sanchez and Moreno, 2004; Hazman *et al.*, 2009). Ontology learning systems are different by the degree of automation from semi-automatic, cooperative and fully automatic.

i. Ontology Learning from Structured Data

These ontology learning procedures extract parts of the ontology using the available structured information. Examples of structured information sources are database schemas, existing ontologies and knowledge bases. The central problem in learning from structured data is to determine which pieces of structural information would provide relevant knowledge.

ii. Ontology Learning from Semi-Structured Data

Semi-Structured data creates richer results on comparison with unstructured data. It provides more semantics in the data and responds better in inference. However, deduction is performed. Examples of Semi-structured data are WordNet, HTML and XML documents. Building ontology from semi-structured data uses both traditional data mining and web content mining techniques.

Karoui *et al.* (2004) used clustering techniques to group similar words into clusters in order to define a concept hierarchy. Bennacer and Karoui (2005) transformed HTML web pages into structured data represented by a relational Table (database). Davulcu *et al.* (2004) and Hazman *et al*. (2009) developed OntoMiner which learns from HTML pages to build taxonomy using structure only. Ontologies can also be built from converting the HTML pages to hierarchical semantic structures as XML to mine it for generating taxonomy (Davulcu *et al.* 2004).

iii. Ontology Learning from Unstructured Data

Building ontology from unstructured data or domain corpus (text) has been targeted by many researchers. The building approaches are classified into three (3) approaches. Buitelaar *et al.* (2005); Cimiano and Völker (2005) built ontology from scratch. Navigli and Velardi (2004) extended a predefined general ontology such as WordNet with possible application domain concepts and relations. The last approach is to built ontology as a composition of other predefined ontologies (Cimiano *et al.* 2007). Battista *et al.* (2007) build design decision where the resultant ontology is either single layer ontology or a multi-layered ontology.

All these need more processing than the semi-structure data, although the quality of the results of ontology learning procedures using structural information is better than the ones using completely unstructured input data (Dellschaft, 2005). Unfortunately, most of the available knowledge is in the form of unstructured text such as Word, PDF documents or Web pages.

2.3.7 Tool for Building Ontology from Unstructured Data

There have been many attempts to reduce this bottleneck through ontology learning tools:

i. *ASIUM:* Faure (1999) learnt ontology from sub-categorisation frames and restrictions of selection. ASIUM learns semantic relations by clustering the nouns based on occurrence with the verbs. In ASIUM, each of the clusters of nouns is presented to the user for labelling.

ii. *Text-To-Onto:* Maedche and Volz (2001) developed ontology learning based on a general architecture and discovered conceptual structures from text. Text-To-Onto implemented a variety of algorithms for diverse ontology learning subtasks particularly relevance measures for term extraction, different algorithms for taxonomy construction as well as techniques for learning relations between concepts (Maedche and Staab, 2000). The focus of Text-To-Onto has been on the algorithmic backbone with the result that combines different algorithms but the interaction with the user has been neglected.

iii. *Hasti:* Shamsfard and Barfooush (2003) extracted the candidate concepts where a set of rules were defined to identify the structural sentences. This used predefined semantic templates to extract the knowledge from the candidate sentences. Hasti is an on-going project but does not report any new methods on identification of relations.

iv. *OntoLT:* Buitelaar *et al.* (2005) built from scratch where protégés were used to extracts ontology from text by defining a number of linguistic patterns over an annotation format that automatically extracts class and slot candidates. Alternatively, the user can define additional rules, either manually or by the integration of a machine learning process. The extracted elements were validated by the user before being inserted into the ontology.

v. *OntoLearn:* Navigli and Velardi (2004) employed population method based on text mining and machine learning techniques. OntoLearn started with an existing generic ontology (WordNet) and a set of documents in a given domain. This produced a domain extended and trimmed version of the initial ontology. The final ontology is output in OWL language. OntoLearn has been applied to different domains (tourism, computer networks, economy).

vi. *Text2Onto:* Cimiano and Volker (2005) employed framework which has been developed to support the acquisition of ontologies from textual documents. It provides an extensible set of methods for learning atomic classes, class subsumption, as well as object properties and instantiation.

vii. *OntoGen:* Fortuna *et al.* (2006) suggested the possible new topics and visualised the topic ontology created in real time. It aims at assisting the user in a fast semi-automatic construction of the topic ontology from a large document collection.

Consequently, it helps by automatically assigning documents to the topics by suggesting names for the topics.

viii. *OntoCmaps:* Zouaq *et al* (2011) developed a domain-independent and ontology learning tool that extracted deep semantic representations from corpora. OntoCmaps generated richer conceptual representations in the form of concept maps and proposed an innovative filtering mechanism. It accepts both unstructured text and other concept maps as input. It is considered as a semi-automatic ontology construction tool. The structure created goes under the subclass of ontology.

ix. *LexOnt:* Arabshian *et al.* (2012) developed a semi-automatic ontology generator that helps in the ontology creation of high-level service ontology. It uses the programmable web directory of services such as Wikipedia and WordNet. It also accepts unstructured text as input and also is considered as a semi-automatic ontology construction tool.

According to Hotho *et al.* (2002), techniques from text learning and information retrieval can be used to build ontologies in a semi or automatic way. The use of this semi or automatic method of building ontology can provide statistically significant terms that could serve as potential concepts in domain ontology. This can be presented as candidate concept words to the domain expert constructing the ontology. In order to evaluate the learnt ontology, the usefulness of the ontology for text classification is investigated. Bloehdorn and Hotho (2004) performed text classification that improved the presence of concept represented in domain knowledge such as ontologies. Therefore, text classification provides a good context for evaluating the results of ontology learning.

2.3.8 Application of Ontology in Different Domains

Ontology application and usage in various domains are discussed in this section. Such domains include agriculture, biology, education, medicine, computer, linguistics etc. where the usage of ontology is proved to be extremely helpful.

i. *Ontology in the Domain of Agriculture:* Jing *et al* (2008) used ontology in various parts of agriculture such as AGROVOC which were used for agriculture controlled-vocabulary searching in various systems such as Food and Agriculture Organisation (FAO). Salvador and Miguel-Angel (2009) organised terms with multi-languages supported terminology of agriculture, forestry, fisheries, food and other related domains for accessing the structure and standardised agricultural terminology in multiple languages by the system and the users.

ii. *Ontology in the Domain of Medicine:* Hoffman *et al.* (2005) organised ontology in form of structured, controlled vocabularies and classifications for several domains of molecular and cellular biology. It is available in the annotation of genes, gene products and sequences.

iii. *Ontology in the Domain of Biology:* Ontology in Plant Database can describe the controlled vocabulary (ontology) for plants. Avraham *et al* (2008) implemented a semantic framework to make meaningful cross-species and database comparisons. Jaiswa *et al.* (2005) described plant structure development stage consisting of a controlled vocabulary of growth and developmental stages in various plants and relationships

iv. *Ontology in the Domain of News:* In the News domain, information extraction does not rely on the page structure but the result of information extraction cooperates with the predefined ontology. Junfang and Li (2010) developed news domain ontology consisting of subconcepts such as navigation page, seed page, content page, navigation page marker path, content page marker path, title, time, picture and content.

v. *Ontology in the Domain of Linguistic:* Talita *et al.* (2010) developed Iban WordNet (IbaWN) using domain ontology as the main language. However, ontology for agricultural domain was constructed. Saad *et al.* (2011) developed SOLAT-based ontology which involves the Al Qur'an, the authentic Hadith and books that focus on the Shafie's school of thought. It involves the types and characteristics of *Solat, hukm*, purification such as *ghusl, wudu and Tayammu*. It also includes Quran verses in Arabic language, images and video. There are 48 concepts, 51 properties and 282 instances.

vi. *Ontology in the Domain of Computer Science:* The ITiCSE (Innovation and Technology in Computer Science Education 2007) computes ontology that describes various disciplines, topics and subtopics belonging to the domain of Computer Sciences. Perich *et al.* (2004) used web ontology language (OWL) and modular component vocabularies to represent intelligent agents with associated beliefs, desires and intentions, time, space, events, user profiles, actions and policies for security and privacy.

2.3.9 Ontology Language Mismatch and Ambiguity

In order to find similarity between terms in MDSs or ontology concepts in heterogeneous environment, Kamolvil (2000) categorised the problems of language mismatch and ambiguity of terms into syntactic, structural and semantic.

i. *Syntactical Mismatch*

Klein (2001) explained that the same ontology is represented in different ways based on representation conflict or different ontology languages. For instance, RDF or OWL is used by different syntactical representations. However, representation conflict requires normalisation before similarity. Moreover, the mismatch is not about the representation of concepts but about the representation of logical notions.

To solve the problem of syntactic mismatch, standardised formats such as RDF or RDFS and OWL (discussed in section 2.2.4) must be used to describe data in a uniform way so that it makes automatic processing of shared information easier. Though standardisation plays an important role in syntactic mismatch, it does not overcome structural mismatch which occurs as a result of the way information is structured even in homogeneous syntactic environments.

ii. *Structural Mismatch*

The ontologies for the same domain knowledge may have different structures as depicted in Figure 2.5 and still refer to the same meaning expressed in different forms.

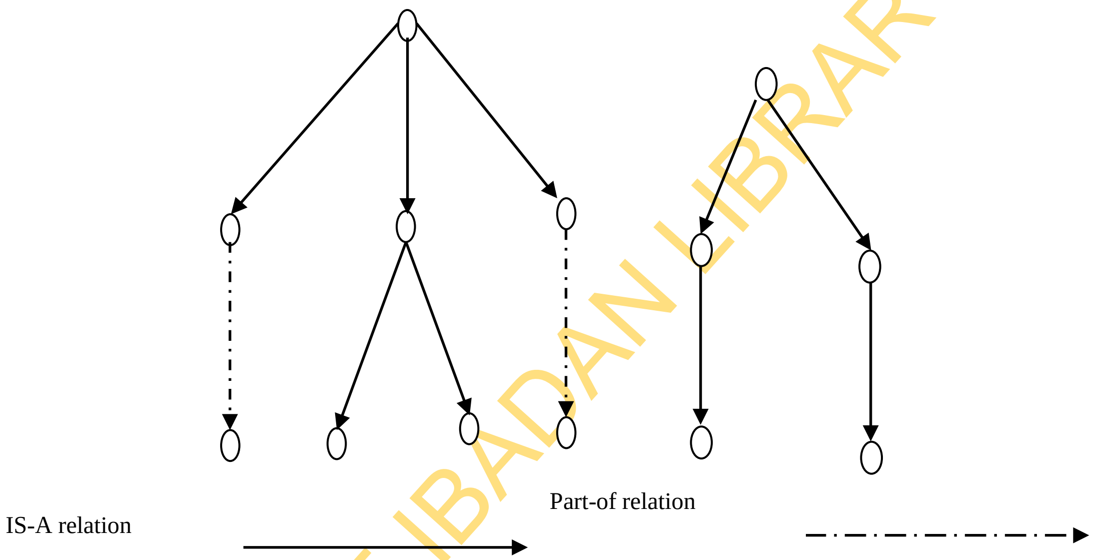


Figure 2.5: Structural Mismatch

Though solutions to syntactic and structural heterogeneity have been developed, the syntactic heterogeneity has been solved using standardised web ontology language (OWL), the problem of semantic heterogeneity is still only partially solved.

iii. Semantic Mismatch and Ambiguity

Visser *et al.* (1997) described semantic mismatch and ambiguity as when two contexts do not share the same interpretation of information, for instance synonyms and homonyms. The problem is referred to as a term or concept mismatch and ambiguity respectively. For example, two terms or concepts may use different names (synonyms) as in *subject* and *course*.

This refers to terms or concepts with same names but different contextual meaning, (homonym) for instance, *bank* (financial institution) and *bank* (edge of a river). However, to solve these problems (mismatch and ambiguity) in web document requires the use of different knowledge sources and human effort.

.

Similarity provides means to address the language mismatch and ambiguity gap between web documents by identifying related terms or concepts present in the retrieved documents. Traditionally, similarity was undertaken by human domain experts and only recently have approaches been developed to semi-automate or automate the process. The resolution of similarity of terms is achieved through semantic processing which is a major concern in this research work.

2.4 Information Retrieval

This section presents basic concepts from the field of Information Retrieval (IR), an overview of established standards and major textual retrieval methods. It also focuses on the drawbacks and evaluation of the information retrieval system measures. The purpose of information retrieval (IR) is to retrieve all the relevant documents (snippets) while at the same time to retrieve few of the irrelevant snippets. This is expressed in the definition by Baeza-Yates and Ribeiro-Neto (1999) which explains that

> "Information Retrieval (IR) deals with the representation, storage, organization of, and access to information items. The representation and organization of the information items should provide the user with easy access to the information in which he is interested."

The definition deals with the storage and the retrieval of data which are usually represented as vectors in multidimensional space. This is especially suitable for text retrieval which stores a collection of text documents. In Information Retrieval (IR) system as depicted in Figure 2.6, the user issues a query to the retrieval system through the query search engine. The retrieval system uses the document index to retrieve documents that contain some query terms, computes relevant scores (similarity) and ranks the retrieved documents according to the scores. The ranked documents are presented to the user. The document collection is also called the text database which is indexed by the indexer for efficient retrieval.

Due to the semantic disconnection between query and documents, information retrieval system is liable to return a lot of irrelevant documents. The IR system has to be interpreted and rank its document according to how relevant it is to the user's query. Consequently, the notion of relevance is at the centre of information retrieval. Salton and Mcgill (1983) considered matching between the user's information need represented by a query formulation and the system-internal representations of documents.

2.4.1 Information Retrieval Models

Model is an idealization or abstraction of an actual process. It is used to study properties, draw conclusions and make predictions. The quality of the conclusions depends upon how closely the model represents reality. Moreso, the retrieval models describe the computational process in terms of how documents are ranked. It can also describe the human process. For example, it can describe information need and interaction. Types of Retrieval Models are based on the following concepts**:**

i. *Exact Match Retrieval:* The query specifies precise retrieval criteria in which every document either matches or fails to match query. The result is a set of documents usually in no particular order or often in reverse-chronological order.

ii. *Best Match Retrieval:* The query describes retrieval criteria for desired documents in which every document matches a query to some degree. The result is a ranked list of documents with the best and it is usually more accurate.

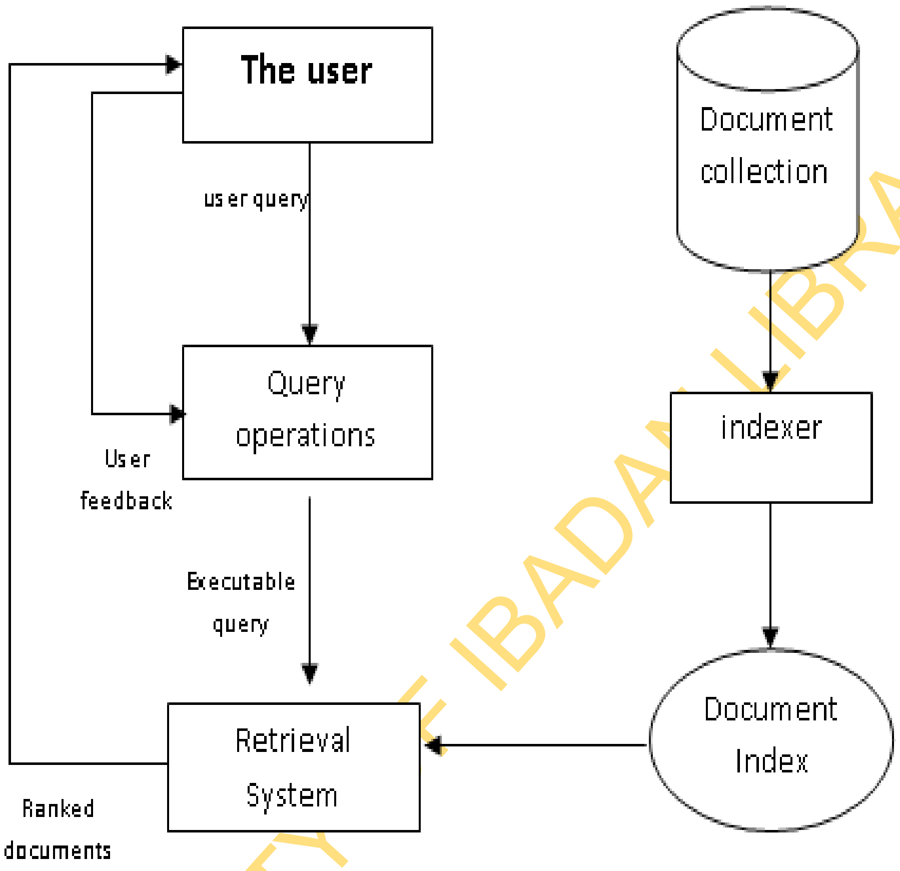


Figure 2.6: Information Retrieval System Architecture

Traditional information retrieval system relies on keyword to index documents and queries. In such systems, documents retrieved are based on the number of shared keywords with the query. An information retrieval system employs a trade-off between the precision and recall to quantitatively measure the performance of information retrieval. Following the definition in Yates and Neto (1999), however, information retrieval model is a quadruple D, Q, F, Sim where

D : set of documents collection.

Q : set of queries.

F : framework for modelling snippets, queries, and relationships and

$Sim : Q \times D \rightarrow U$ a ranking function that defines an association between queries and documents where U is a totally ordered set [0, 1].

The following types of model have been developed in retrieval system:

1. Boolean model,
2. Statistical Model this includes
   - i. Vector Space Model
   - ii. Probabilistic Model) and
   - iii. Latent Semantic Indexing
3. Linguistic and Knowledge Models.

Belkin and Croft (1992) described the first model as the "exact match" models while refined to the latter ones as the "best match" models.

2.4.1.1 Boolean Model

The Boolean information retrieval system is based on Boolean logic and classical set theory because both the documents to be searched and the user's query are conceived as sets of terms. However, retrieval is based on whether or not the documents contain the query terms given a finite set $T = (t_1, t_2 ... t_n)$ of elements called index terms (keywords), a finite set $D = (d_1, d_2 ... d_n)$ and D_i is a set of T of elements in documents.

Based on the previous notation in the retrieval model;

i. D: The elements of D are represented as sets of keywords occur in each document. The term is either present (1) or absent (0) in the document. Documents can thus be seen as the conjunction of terms.

ii. Q: Queries are represented as a Boolean expression as keywords and logic operators (AND (∧), OR (∨), NOT (¬)) which are normalised to a disjunction of conjunctive vectors.

iii. F : Sets of terms and documents.

iv. Sim : This is defined by considering that a document is predicted to be relevant to a query if its keywords satisfy the query expression.

Frakes and Yates (1992) described the Boolean model as being easy to implement and also computationally efficient. The model expresses structural and conceptual constraints that describe important linguistic features. The Boolean retrieval model is very effective if a query requires an exhaustive and unambiguous selection. However, Belkin and Croft (1992) explained that users find it difficult to construct effective Boolean queries for several reasons because users use natural language terms (AND, OR or NOT) that have a different meaning when used in a query. Thus, users make errors while constructing a Boolean query.

2.4.1.2 Statistical Model

The vector space and probabilistic models are the two major examples of the statistical retrieval approach. Both models use statistical information in the form of term frequencies to determine the relevance of documents with respect to a query. Although they differ in the way the term frequencies are used, both produce output as list of documents ranked by estimated relevance. In addition,the statistical retrieval model addresses some of the problems of the Boolean retrieval method but still has some disadvantages.

These enable users to control the output by setting a relevance threshold or by specifying a certain number of documents to display in web documents. However, queries are easier formulated because users do not have to learn a query language and can therefore use natural language. But, it has a limited expressive power. For example, the NOT operation cannot be represented because only positive weights are used. Hearst (1994) explained that the model provides users with a limited view of the information space and it does not directly suggest how to modify a query if the need arises.

2.4.1.2.1 Vector Space Model

In information retrieval, the documents stored are normally identified by sets of terms or keywords. Salton and Yang (1973) developed SMART for the representation of the collection of document content and used vector space models for representation. In the Vector Space Model (VSD), documents and queries are represented as vectors in a t-dimensional space based on the notation of information retrieval.

i. D: Documents (snippets) are represented by a vector of terms or keywords which occur in the document. Each term in the document has a pair (t_i, d_i) with a positive non-binary associated weight $(w_{i,j})$

ii. Q: Queries are represented as a vector of terms or keywords terms occurred in the query. Each term in the query has a pair (t_i, q) has a positive non-binary associated weight $(w_{i,q})$.

iii. F : This is an algebraic model as vectors in a t-dimensional space.

iv. Sim: This estimates the degree of similarity of a document d_j to a query q as the correlation between the vectors d_j and q in the retrieval system. This correlation is quantified, for instance, by the similarity measure such as cosine of the angle between the two vectors (discussed in section 2.6.1). The same is applied to two different documents as well.

2.4.1.2.2 Probabilistic Retrieval Model

This is based on the probability ranking principle which states that an information retrieval system ranks the documents based on probability of relevance to the query (Belkin and Croft, 1992). The principle takes into account uncertainty in the representation of the information needed and the documents. The probabilistic retrieval model ranks documents in a decreasing order of probability which is noted as $P(R|q, d_j)$ where d_j is a document in D.

i. D: Documents (snippets) are represented as a vector of terms or keywords which occur in a document.

ii. Each term in the document with a pair (t_i, d_i) has a binary associated weight 1 or 0, which denotes the presence or absence of the term in the document.

iii. Q: Queries are represented by a vector of terms or keywords that occur in the query. Each term in the query with a pair (t_i, q) has a binary weight 1 or 0, denoting the presence or absence of the term in the query.

iv. F : This is a probabilistic model that ranks documents in the order of probability of relevance to the query.

v. Sim: This measures the degree of similarity of a document d_j to a query q_i as the probability of d_j to be part of the subset R of relevant documents for q. This measure in the probabilistic model is given by:

$$Sim(d_j, q) = \frac{P(R.|d_j)}{P(\neg R.|d_j)} \qquad (2.1)$$

where

$\neg R$ denotes the set of non-relevant documents,

$P(R.|d_j)$ is the probability of d_j being relevant to the query q, and

$P(\neg R|d_j)$ is the probability of d_j being non relevant to q.

2.4.1.2.3 Latent Semantic Indexing (LSI)

Deerwester *et al.* (1990**)** developed Latent Semantic Indexing in an attempt to overcome the problems of lexical matching. It uses statistically technique to derive conceptual indices instead of individual words for retrieval. These are used to determine the set of terms, word relations (e.g. synonyms) and the strength of these relations. Using LSI to understand the semantic content of a document collection enables the definition of a logical semantic view. LSI is a similarity metric which is an alternative to word overlap measure and dimensionality reduction.

LSI adds step to the indexing process by estimating statistical techniques using an association matrix of term-to-document $(t \times d)$ measure. Latent Semantic Analysis (LSA) computes the arrangement of a k-dimensional semantic space and reflects the major associative patterns in the data. This is done by deriving a set of k uncorrelated indexing factors. These factors are artificial concepts whose lexicalisation is not important for LSI. The meaning of each term or document is expressed by k factor values. These are represented in 2 or 3-dimensional space for visualisation. However, a mathematical technique named Singular Value Decomposition (SVD) is used for the representation. In LSI model, term-by-document matrix is performed by low rank and yields a new representation for each document in the collections. This represents projection into the latent semantic space, therefore, SVD decomposes an n-dimensional space (original space representation) into a k-dimensional (lower) space, this implies $n \succ k$. The SVD takes matrix A and represents it as $\hat{A}$ in a lower dimension space.

where A is orthogonal matrix U, a diagonal matrix Σ and transposes of an orthogonal matrix V . SVD breaks a $(t \times d)$ matrix A into 3 matrices U, Σ, V such that:

$$A = U \Sigma V^T \tag{2.2a}$$

An orthogonal matrix U consists of term in each of the document collections and transpose of an orthogonal matrix V consists of document D in the new space, the diagonal matrix Σ contains the singular values of A in descending order. Also, a user's query is represented as a vector in k-dimensional space and which is compared to document collection.

The matrix A is represented as: $Term(t) = t_1, t_2 \cdots t_n$ that appears in each document $d = d_1, d_2 \cdots d_n$ of a given query *(q)*. The matrix A is decomposed so that U, V and Σ are found. Then, rank $(A) = r$ for $k_i \le r$.

where

U is a $t \times t$ orthogonal matrix whose column vector is left singular vector of A.

V is a $d \times d$ orthogonal matrix whose column is right singular vector of A and

The $rank(A) = r$ is the number of its non-zero singular value.

Σ is a $d \times d$ diagonal matrix having the singular values of A. This ordered matrix decreases along its diagonal such that $\Sigma = diag(\delta_1, \delta_2 ... \delta_n)$ $\delta_i \succ 0$

The latent semantic space has fewer dimensions (dimensionality reduction) than the original vector space model. The search engine algorithms take the words in the query and determine how relevant it is to each other. For instance, in searching for “losing weight”, it looks for words that relate to losing weight such as weight loss, diet, exercise, eating right etc. These are latent semantic indexing terms. However, the new document vector and query coordinates in the reduced 2-dimensional space are found and finally the similarities (for instance Cosine similarity) between the rank documents in decreasing order of query are calculated. This approach only works well with relatively small number of dimension with query compared to every document in the collection.

Osinski and Weiss, (2005) presented a concept-driven algorithm for clustering search results, the Lingo algorithm, which uses LSI techniques to separate search results into meaningful group but does not consider building semantic relationships between the groups. However, common phrases are extracted using suffix tree clustering technique and concept induction using latent semantic analysis is conducted. According to Chooghyun and Choi (2010), LSI uses K-means clustering algorithm on medical document as the document which contains many acronyms of clinical terms used as retrieval method for analysing broken web links. Biatov *et al.* (2009) applied an audio clips-feature vectors matrix mapping the clips content into low dimensional latent semantic space. The clips were compared using document-to-document comparison measure. As mentioned, a document is represented by the weighted sum of its component term vectors. The similarity between two documents is computed by means of the similarity metrics between the corresponding representation vectors. The cosine similarity between term and documents is defined for LSI as in 2.2a is modified in 2.2b below:

$$Cos(t_i, d_j) = \frac{U_i \Sigma V_j^T}{\left\| U_i \Sigma^{\frac{1}{2}} \right\| \bullet \left\| V_j \Sigma^{\frac{1}{2}} \right\|} \qquad (2.2b)$$

The indices of LSI are less meaningful semantically and it is difficult to find out similarities between terms. This bag-of-word (BOW) method or approach leads to unstructured information which is less semantic. However, compound terms are treated as two terms and finally, there is time complexity for SVD in dynamic collections.

Many information retrieval systems represent documents and queries with a bag of words in documents collections. This results in inaccuracy and the user's queries seem imprecise. This approach discussed so far, misses many relevant documents because it does not capture the complete meaning of the user's query. These disadvantages are mainly due to the ambiguity and limited expressiveness of single words. Lancaster and Warner (1993) addressed representation in linguistic and knowledge-based search by performing a morphological, syntactic and semantic analysis to retrieve documents more effectively. In a morphological analysis, roots and affixes are analysed to determine the parts of speech (noun, verb, adjective etc.) of the words.

Developing linguistic retrieval system is difficult and requires complex knowledge bases of semantic information retrieval. Hence, it requires techniques that are commonly referred to as artificial intelligence or expert systems techniques.

2.4.1.3 Linguistic and Knowledge Model

The alternative way to investigate words in the documents is to use concepts to capture the context of documents. This is done by creating a concept-based document representation model. Concepts are units of knowledge with a unique meaning (ISO, 2009). There are three advantages of concepts over words. Firstly, concepts are less redundant because synonyms such as U.S. and United States unify to the same concepts. Secondly, they disambiguate words such as “bank” that have multiple meanings. Thirdly, semantic relations between concepts are defined, quantified and taken into account when computing the similarity between terms in the documents.

Using concepts for document representation is more discriminative than the bag-of-words model. Conceptual indexing concerns the representation of document semantics and its proper use in retrieval. It aims at representing the context of the document by semantic knowledge principled approach. The knowledge was exploited through a simple ontology. Concept document representations can solve the problem of language mismatch and the ambiguity by expanding the representation to incorporate concepts from a document (Bloehdorn & Hotho, 2004). This is done by using supervised machine learning techniques to determine how to combine concepts and their semantic relations into a document similarity measure.

Concepts and their relations have been exploited in many text processing tasks. These include information retrieval (Milne et al, 2007) and semantic analysis (Mihalcea *et al*., 2006). Consequently, document clustering and snippets similarity measures are enriched based on lexical or conceptual overlap with semantic relations between concepts (Hu, et al., 2008). Therefore, additional external semantic structures (information resources) are needed for mapping document representations to concepts. Such resources are dictionaries, thesauri or ontologies (Guarino *et al*.,, 1999). However, Gonzalo *et al*. (1998) suggested that indexing with WordNet synsets can improve information retrieval. Khan and Luo, (2002) presented a method of relating concept from ontology to the documents in the retrieval system. This shows the relationship between concepts from one category of ontology would be different from concept in other categories of ontology.

Ontology and its integration into text representation describe ontology hierarchy but various approaches have been developed with this approach. This includes WordNet and Domain Ontology.

2.4.1.3.1 WordNet Ontology

This definition describes ontology as a set of concepts, sub-concepts and in relation to the ontology hierarchy in WordNet.

Definition 2.1: Ontology is a tuples $O = (c \subset C)$ which consists of a set C whose elements are concept and a partial order on c that is concept hierarchy or taxonomy.

Definition 2.2: If $c_1 \subset C, c_1 \subset c_2$ for $c_1, c_2 \in C$ then c_1 is a sub-concept of c_2 and c_2 is a super-concept of c_1 if $c_1 \subset c$, c_2 and there is $c_3 \in C$ with $c_1 \subset c$, $c_3 \in c, c_2$ c_1 is a direct sub-concept of c_2 and c_2 is a direct super-concept of c_1 therefore $c_1 \subset c_2$.
Therefore, WordNet ontology is defined based on the ontology definition.

Definition 2.3: A lexicon (WordNet) of ontology O is a tuples Lex = ($\mathrm{S_c}$, $\mathrm{Ref_c}$) consist a set $\mathrm{S_c}$ whose elements are lexical entries for concepts and a relation $\mathrm{Ref_c} \subseteq S_c * \mathrm{C}$ that is lexical reference for concepts.

Let

$(c, C) \in \mathrm{Ref_c}\ \mathrm{S}$ which hold for all $c \in C \cap S_c$ such that $\mathrm{Re}\, f_c^{-i}(s) = \{c \in C | (s, c) \in, \mathrm{Re}\, f_c\}$, for $c \in C$, $\mathrm{Re}\, f_c^{-i}(c) = \{s \in S | (s, c) \in, \mathrm{Re}\, f_c\}$.

An ontology with lexicon is a pair $(O : Lex)$ where O is an ontology and Lex is a lexicon for O.

For example: Let assume two synsets of “ft and feet” and the corresponding word is “foot”. In WordNet, the function Ref_c relates terms with a lexical entry (e.g s_1=“ft” and s_2 = “feet”) then the corresponding concepts (e.g., synsets c_1 = “ft, c_2 = “feet (human foot)). Thus, for a term t appearing in a document d, $\mathrm{Ref_c}$(t) allows for retrieving its corresponding concepts. Enriching document similarity with semantic relations is to expand each concept with its hypernym. The concepts in WordNet are ambiguous. Therefore, adding or replacing terms by concepts may add noise to the representation and may induce a loss of information. Moreover, most multi-word terms have no senses in WordNet. Thus, one can only compute the intended meaning for each component word of the term. WordNet ontology can perform well with local repository by finding relation between one or more ontologies. If there is no appropriate ontology in local repository, then, WordNet cannot extend the query term to get the semantic terms.

2.4.1.3.2 Domain Ontology

According to Zhao and Karypis (2005), the hierarchical structure of the ontology defines various language mismatch and ambiguity of texts and provides the methods for a uniform processing of text. However, ontology is defined in relation to the domain ontology as a 5 tuples $O = \{L, C, F, H, ROOT\}$ which consists of lexicon L (specific to certain domain) and contains a set of natural language terms. A set of concepts C, Function **(**F) $L \rightarrow C$ and F links sets of terms to the set of concepts it refers to in ontology. The Concepts C in the Hierarchy H are hierarchically related by the directed, acyclic and relations $(H \subset C \times C)$ and finally ROOT as a top concept is an element $H(C, ROOT)$.

In domain ontology, the relationships of a concept and its associated sub-concepts form a tree-like structure. These are used to discover the user's interest and form autonomous searching of related web content. Such hierarchical structure is in form of categories, attributes or activities. Therefore, the OWL domain ontologies were normalised using Protégé 4.2; then the concepts from the ontology were used to model user' request. Jian-liang *et al.* (2009) presented domain ontology that represents knowledge that can be shared and reused. Consequently, concepts that are semantically clustered together are used by domain ontology to discern the main structure in a natural language sentence or snippet.

Vallet *et al.* (2005); Castells *et al.* (2007) described ontology-based information retrieval model. The model uses ontology for indexing, query interpretation and query expansion purposes. This is achieved by annotating text documents with ontology entities. But the index terms are weighted based on an adaptation of the Vector Space Model which determines TF-IDF values for the ontology based index terms.

Tomassen and Strasunskas (2009) presented method of indexing documents with ontology vocabulary based on the Vector Space Model. In this approach, a feature vector is calculated for each concept term from the ontology, based on the term's occurrences in the document corpus. This way, the index terms derived from the ontology are adapted to the domain terminology. This method is employed in this thesis but with the addition of terms from domain of interest and concept weighting estimation is employed instead of tf-idf weighting.

2.4.2 Text Categorisation

Text Categorization (TC) is a supervised learning technique that automatically assigns predefined categories to free documents. Yang (1999) described it as that which classifies documents according to the topics while Kessler *et al.* (1997) explained that it depends on the way a text was created, edited or published. Text representation or categorisation has impact on text retrieval effectiveness but, it is not an ideal approach. It contains textual requests with fewer number and size. Text categorisation addresses these problems of large amounts of information and challenges resulting from the polysemous characteristics of natural languages.

With the drastically increased number of electronic documents, there is an urgent demand for high quality of text categorization with various classification criteria. The machine learning approach was introduced but there still exist several difficulties. The approach lacks semantic support and makes it impossible to match a term (concept) from the words it represents, especially where the concept is represented by a phrase in the context. Also, there exists multi-presentation of information (polysemy). Finally, incomplete information about a concept in a document causes incomplete results of machine learning.

In Montanes *et al.* (2003), documents in text classification are represented by a great amount of features and most could be irrelevant or noisy. However, web snippets return information not as full text document. This lack of information makes it difficult to judge the relevance of snippets of information. Therefore, there is need for more structured way of representing information on web documents.

2.4.3 Text Document Clustering

Text Document Clustering is an unsupervised learning technique that provides solution to a single ordered list and clusters the search results. It presents a list of clusters to the users. Jain *et al*. (1999) discovered how documents are grouped together with similar documents but this needs to be optimised. Oikonomakou and Vazirgiannis (2005) organised information in a way that made it easier for the end users to find the information efficiently and accurately. Document clustering can be defined as follows:

Definition 2.4: A document collection $D = \{d_1, d_2, d_3 ... d_n\}$ contains N documents. The documents are sub-grouped based on the semantics of the text contents present in each document. Let K be sub-groups, the clustering process generates $C = \{c_1, c_2 ... c_k\}$ clusters with each c_i being non empty.

Most of the existing text clustering methods uses clustering techniques which depend only on term strength and document frequency. The single terms are used as features for representing the documents and can be treated independently. Beyer *et al.* (1999) explained that text document clustering has the problem of big volume, high dimensionality and complex semantics. This results in computational inefficiency because clustering in high-dimensional spaces is very difficult and every data point tends to have the same distance from all other data points.

2.4.3.1 Clustering Approaches

Researchers in the data mining community have proposed many clustering algorithms in order to perform unsupervised learning. These algorithms can be classified into at least six categories. These are fuzzy clustering, nearest-neighbour clustering, hierarchical clustering, and artificial neural networks for clustering, statistical clustering algorithms and density-based clustering. In this research work, only the hierarchical clustering algorithm would be focused on based on its ontological structure. Hierarchical algorithms can be categorized into two subcategories as discussed below:

i. *Conceptual-Based Clustering:* Fisher (1987) introduced incremental conceptual clustering algorithm (COBWEB). The representation was done by a set of attribute-value pairs where two or more objects belong to the same cluster if they share common concepts. Seo and Ozden (2004) used COBWEB to generate ontology from files description in order to perform ontology-based file naming but the process was too complex.

ii. *Distance-Based Clustering:* It represents objects in a well-defined space as vector in a 2D Cartesian space. Thus, two or more objects would be assigned to the same cluster if close according to a given distance function. These can be called agglomerative clustering and partitional clustering. Zhao and Karypis (2005) used hierarchical clustering to cluster for its optimisation.

Agglomerative algorithms give better clustering solutions than partitional algorithms. The main advantage of partitional algorithms is low complexities which allow clustering millions of elements. Consequently, partitional clustering algorithms have been recognised to be better suited for handling large document datasets than agglomerative hierarchical clustering due to relatively low computational requirements (Steinbach *et al.*, 2000). However, partitional algorithms can suffer from local minima and this depends on the input order of the items. The standard K-means clustering algorithm is an iterative partitional clustering process that aims at minimising the least squares error criterion (Salton and Buckley, 1988). These partition algorithms are discussed below:

i. K-means algorithm

The K-means algorithm (Hartigan and Wong, 1979) is a popular clustering tool used in scientific and industrial applications. The name comes from the representation that each cluster C is the mean (or weighted average) m with a points called centroid as in Figure 2.7. The goal in k-means is to produce clusters from a set of n objects, so that the squared-error objectives function is minimised.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2 \qquad (2.3)$$

In the 2.3,

p is a point in a cluster C_i and

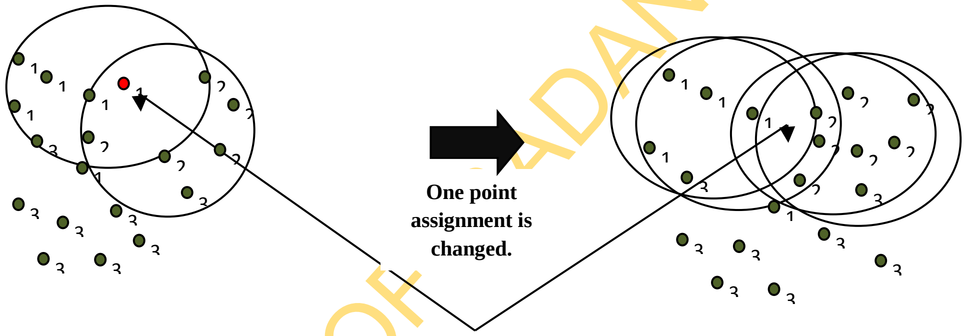


Figure 2.7: Clustering Optimisation with Centroid Computation

m_i is the mean of cluster C_i.

The mean of a cluster is given by a vector which contains, for each attribute, the mean values of the data objects in the cluster.

ii. Lingo

There is need for better ways of clustering web that would provide sound knowledge of the content present in the documents. The existing latent structure of diverse topics is discovered for search result. Stanislaw and Dawid (2005) combined common phrase. Ahmed and Amar (2010) and Chatterjee and Pushplata, *(*2012) presented semantic Lingo algorithm that extended the techniques. This adds semantic recognition using WordNet database to achieve semantics. However, incremental processing significantly improved the efficiency of search results clustering but the semantic lingo algorithm is not incremental.

iii. Suffix Tree Clustering (STC)

Document clustering algorithms discussed so far are based on vector model for computation. Eissen *et al.* (2005) and Chim and Deng (2008) described a better way to achieve more accurate document clustering based on phrase which is more informative than feature term-base.

Zamir and Etzioni (1998) proposed Suffix Tree Clustering (STC) which is based on the suffix tree document model. According to Hammouda and Kamel (2004), Suffix Tree Clustering (STC) algorithm groups the input texts according to the identical phrases it shares. However, the principle behind the approach is that when phrases are compared to single keywords, they have greater descriptive power. Hence, the clustering results produced by phrase based similarity measure are of high quality when compared to the semantic interpretation of the corpus. Chim and Deng (2008) and Chung *et al*. (2008) utilised suffix tree model based approach. A great advantage of STC is that phrases are used to provide concise, meaningful descriptions of groups and offer more semantic representations of the text present in the document**.**

The algorithm for STC is theoretically fast with a runtime of $O(n)$, where n is the total number of words in all combined snippets. The Suffix Tree Clustering algorithm works in two main phases. These are: Base cluster discovery phase and Base cluster merging phase. The algorithm is as follows:

i. It computes two frequent terms sets based on user-specified minimum requirements.
ii. All terms not in any of the two frequent terms sets are removed from the documents resulted in compact representation of snippets.
iii. The compact documents are added one-by-one into a generalised suffix tree data structure. The algorithm then traverses the generalised suffix tree in a depth-first format.
iv. Every node labelled by a substring of the compact document set containing at least two terms and supported by at least two snippets becomes a cluster candidate.
v. From the set of terms, the clusters with the longest sequence are selected and all clusters with k-mismatched sequences are merged with it.
vi. This process is repeated until there are no cluster candidates left. Thus, it produces the clusters.

*The Algorithm for the suffix tree clustering using Goggle API is as follows***:**

i. The snippets are retrieved from Google
ii. Construct Suffix Tree by inserting the strings associated with each snippets onto the suffix tree
iii. Merge Clusters by combining similar nodes of the suffix tree
iv. A label is generated for each cluster
v. Score by ranking clusters

The algorithm has an important characteristic that outputted clusters have overlapping documents. This advantage ensures that a large number of substantial clusters are generated, each of which can be labelled fairly accurately. These algorithms are implemented using Carrot2 API. It turns out that STC works well when quick overviews of documents relevant to distinct subtopics are needed. Moreover, clustering is more useful when one is interested in retrieving multiple documents relevant to each subtopic.

2.4.3.2 Tool for Implementing Clustering Algorithms

Carrot2 is an open source framework for building search clustering engines. It can automatically organise small collection of documents into thematic categories. Apart from the specialised document clustering algorithms, Carrot2 offers ready-to-use components for fetching search results from various sources such as Google API (Application Program Interface), Bing API, eTools Meta Search, Lucene, SOLR and more. Carrot2 is implemented in Java but also in a native C#/.NET API. Consequently, all the clustering techniques discussed can be implemented using Carrot2.

Madsen *et al.* (2004) suggested that not all the words presented in a document retrieved through clustering can be used for training and text documents must be in a clear word format.

2.4.4 Document Indexing (Documents Representation)

Leopold and Kindermann (2002) defined document as a sequence of words made up of a joint membership of terms while indexing involves the selection and assignment of terms or the extraction of terms from a documentary unit in order to indicate topic, features or possible use of the unit. A document is usually represented by an array of words or terms.

The documents representation (indexing) is one of the preprocessing techniques that are used to reduce the features' complexity of the documents. Documents can be represented by a wide range of different feature descriptions. There are two kinds of processes involved in text documents representation; these are document indexing and term weighting. This makes documents easier to handle in processing and is characterized by the following:

i. Feature Extraction (Linguistic Analysis)
ii. Feature Selection (Feature Vector).
iii. Conceptual Indexing (Linguistic and knowledge Model)

2.4.4.1 Feature Extraction (Linguistic Analysis)

The aim of feature extraction methods is the reduction of the dimensionality of the features by removing features that are considered irrelevant for the training to allow an efficient data manipulation and representation.

Three major linguistic properties of documents are identified:

i. Morphological
ii. Syntactical and
iii. Semantics.

i. Morphology Analysis

Wang and Wang (2005) presented a clear border of each language structure and also the language dependent factors like stop words removal, stemming or lemmatisation and finally tokenisation. The aspect of morphology identifies, analyses and describes the structure of a word in a given document such as root words, affixes, parts of speech, lexeme and lemma of each word in the documents.

Tokenisation is used in order to mark a set of characters and distinguish it as a word or phrase. Concepts are mostly nouns, proper nouns, and noun phrases. POS tags play an essential role in both syntactic- and semantic-based learning. The POS tagger uses tokenized documents as input and assigns a POS label for all tokens. Words such as auxiliary verbs, conjunctions and articles are called stopwords. This is done because these words appear in most of the documents often. Word normalisation involves stemming and lemmatisation but these techniques produce a normalised form of web documents retrieved. Word stemming does not usually produce a basic form for example, (e.g. "teeth and tooth") but only an approximation of the form. For example, the words "train", "training", "trainer" and "trains" can be replaced with "train". On the other hand, lemmatization replaces the suffix of a word with a different one or removes the suffix of a word completely to get the basic word forms (lemma). In order to recognise the basic form of the corresponding lemma, WordNet-based which belongs to the group of dictionary lemmatisation algorithms is used. This makes it possible to generate subsets of word forms for each stem and look for the corresponding lemma in WordNet.

The algorithm can be described as follows:

i. for each set of forms (word) generated from one stem do
ii. for each form do

iii. search corresponding lemma in WordNet

iv. if found, assign the basic form (lemma) for each item from the set

v. if not, continue with the next form

ii. Syntactic Analysis

In syntactic analysis (parsing), linear sequences of words are transformed into structures that show how the words relate to each other. Syntactic analysis exploits the results of morphological analysis to build a structural description of the sentence. The goal of this process is to convert the sequence of words that forms the sentence into a structure that defines the units (token) represented. This is to determine the Part of Speech (POS) and grammatical constituents each word belongs to. The POS is the process of making a word in a text corresponds to a particular part of speech based on both its definition and its context. That is relationship with adjacent and related words in a phrase sentence i.e. noun, verb, adverb, adjectives etc.

The POS tagger processes the token and attaches a part of speech to each word. The tagger is used to tag the retrieved snippet to perform linguistic transformation using Natural Language Toolkit which is a rule-based tagging method. The Natural Language Toolkit (NLTK) provides documentation for each tag by converting a tagged token representation using a turple consisting of the token and the tag. In processing the text documents, the auxiliary information is associated with each token (tagging) and disambiguates hypernyms by associating it with word sense labels.

For example, the input snippet of “Document is retrieved from the web and processed with SKLean tool”. This is being parsed to the parser tree (S (NPL Document) (VP is (VP retrieved (PP from (NP (NPL the web) - COMMA- (VP processed (PP with (NP (NPL SKLean) tool))))))) -PERIOD-). It is important that the sentence has been converted into a hierarchical structure. The structure corresponds to meaning units and semantic analysis is performed. Finally, POS tagging gives information about semantic constituent of a word.

iii. Semantic Analysis

The semantic analysis performs the task of extracting the semantic relationships between the selected concepts or terms in the text documents. This is performed either by the use of domain-specific ontology or by exploiting the semantic structure of the analysed sentences with the help of the user. The interactions from the user is acceptable as fully automated approach and semantic analysis can only be possible due to the requirement for deep understanding of the domain knowledge (Snoussi and Nie, 2002). Noun phrases and verb phrases are good indications of concepts to be included in the semantic documents model. Therefore, every noun phrase or verb phrase extracted from the analysed snippets are represented as concepts.

These Noun Phrases (NP) are analysed to filter determiners (such as the, a, an) that usually occur in word phrases. Furthermore, lexical ambiguity can be addressed by algorithmic methods that automatically associate the appropriate meaning with a word in context. This task is called word sense disambiguation as discussed in section 2.5.

2.4.4.2 Feature Selection (Term Vector)

Feature selection also known as term vectors reduces terms' importance by information retrieval measures. Feature selection (FS) is based on the feature vectors from vector representation and improves the scalability, efficiency and accuracy of text document. Wang *et al* (2006) presented feature selection that considers domain and algorithm characteristics as good method. Most existing feature selection techniques (Yang and Pederson, 1997) and learning algorithms (Joachims, 1998; McCallum and Nigam, 1998) have produced good results on a number of standard text collections but the majority of these works used a simple "bag of words" representation of text in which each feature corresponds to a single word.

Dimensionality reduction is overcome either by feature selection techniques such as mutual information (information gain) (Lewis and Ringuette, 1994), Chi Square (Yang and Pedersen, 1997) or gain ratio (Debole and Sebastiani, 2003). The existing term weighting methods for feature selection is use to describe each of the feature selection. The feature selection methods included in this study are as follows:

i. *Information Gain (IG):* This is frequently used as a term goodness criterion in the field of machine learning. It measures the difference in the entropy of category of documents's prediction by knowing the presence or absence of a term in a document. Let $\{c\}_{i=1}^{m}$ denote the set of categories in the target space. The information gain of term *t* is defined to be:

$$IG = -\sum_{i=1}^{m} P_r(c_i)\log\Pr(c_i) + \Pr(t)\sum_{i-1}^{m}\Pr(c_i|t)\log\Pr(c_i|t\Big|) + \Pr(t)\sum_{i-1}^{m}\Pr(c_i|t)\log\Pr(c_i|t) + \Pr(t)$$

(2.4)

ii. *Chi-Square (χ2):* It measures the lack of independence between the terms in the documents category and calculates the difference between the observed frequencies and the frequencies expected under the independence assumption.

The chi square statistic has a natural value of zero if *t* and *c* are independent. Each category of the chi square statistic between each unique term in the training corpus and that category then combines the category specific scores of each term into two scores:

If one consider the two way contingency table of a term *t* **,** Probability *Pr* and a category *ci*

$$\chi^2_{\arg}(t) = \sum_{i=1}^{m}\Pr(c_i)\chi^2(t,c_i)$$

(2.5)

$$\chi 2_{\max}(t) = \max_{i=1}^{m}\{\chi^2(t,c_i)\}$$

The measure is reliable for low frequency term.

iii. *Mutual Information:* It is derived from information theory. It gauges the reduction in the uncertainty of one random variable when the other is known. The metric is commonly applied for identifying term collocations. In a similar way, it can be used for measuring the association between a term and a specific topic of interest.

$$MI(t,c) = \log\frac{\Pr(t \wedge c)}{\Pr(t)\times\Pr(c)}$$

(2.6)

iv. *Relevant Document Frequency (RDF):* It is considered as the binary version of document frequency. Yang and Pedersen (1997) proved that document frequency has been successful in m-ary classification problem because it identifies terms that occur in many subject topics. The disadvantage is that its application to binary classification problems however resulted in frequent, non-specific terms being selected. RDF on the other hand exploits relevant information (r) to identify terms that occur frequently within the documents of interest.

$$RDF_i = r \qquad (2.7)$$

The first category of term extraction methods was based on two statistical methods. They are absolute term frequency and Term Frequency-Inverse Document Frequency weight provides options to select the minimum frequency to be considered as well as the maximum number of words in a term.

v. *Absolute term frequency* (tf_i)**:** This is defined by

$$tf_i = \sum_{i=0}^{n} n_i \qquad (2.8)$$

where n is the number of term *i appear* in the set

vi. *Term Frequency-Inverse Document Frequency (TF-IDF) (Salton and Buckley, 1988):* This term weighing method is used in Information Retrieval (IR) method. It evaluates how important a word is to a document in a collection or corpus, defined by:

$$(tf/idf)_{i,j} = tf_{i,j} * idf_i \qquad (2.9a)$$

$$idf_i = \log \frac{|D|}{j : t_i \in d_j}$$

. Therefore, 2.8(i) can be expressed as

$$(tf/idf)_{i,j} = tf_{i,j} * \log\left(\frac{|D|}{df(j)+1}\right) \qquad (2.9b)$$

Let

$tf_{i,j}$ is the absolute term frequencies of term i in document j ,

idf_i is the inverse document frequency of term *i* and

$|D|$ is the total number of documents in the text document and

$j : t_i \in d_j$ is the number of documents where term t_i appears

2.5 Word Sense Disambiguation (WSD)

The existing retrieval systems focus on matching the similarity of individual keywords. It ignores the semantic relationships that exist among the multiple keywords. In many information retrieval analyses, only one sense is associated with each word. The WSD is a process which filters a set of possible candidate senses. The resulting sets can contain multiple senses if desired by the expert who designs the system. But there are different language conflict issues that occurr in different senses such as polysemy and synonymy (Yang and Wu, 2011 and Fang *et al.* 2005). For instance, polysemous words can be accommodated in the classical problem of disambiguating words occurring in natural language; so the available context information is a body of text co-occurring with the target word (Navigli and Ponzetto, 2010).

The available context information originated from ontology is different compared to a natural language document. In ontology, natural language is a rare occurrence and is usually limited to brief concept descriptions in the form of annotations. Hence, context information must be extracted from the entire concept description, its associated properties and other related concepts. WSD can be described as the automatic identification of the correct sense(s) of a given word using the information in the proximity of that word as context. WSD techniques can use resources such as the WordNet thesaurus (Voorhees, 1994) or co-occurrence data (Schuetze and Pedersen 1995) to find possible senses of a word and map word occurrences to the correct sense. Many different approaches to WSD have been developed over the past decades. Due to the prevalence of applied machine-learning techniques, three general categories of approaches to WSD had emerged:

i. *Supervised Disambiguation:* WSD can be formulated as a classification problem. Montoyo *et al.* (2005) and Navigli (2009) used decision lists, decision trees, Naïve Bayes classifier, Neural-Networks, instance-based methods such as the k-NN approach and ensemble methods to combine different classifiers. A training set is created by tagging sentences with the correct senses of its contained words. Once the training set has reached a sufficient size, it can be used as basis for a supervised classification method.

ii. *Unsupervised Disambiguation:* These methods have the advantage that does not rely on the presence of a manually annotated training set, a situation which is also referred to as the knowledge acquisition bottleneck (Gale *et al.* 1992). However, unsupervised methods share the same intuition behind supervised methods that is words of the same sense co-occur alongside the same set of words (Pedersen, 2006). These rely on clustering methods where each cluster denotes a different word sense.

iii. *Knowledge-based Disambiguation:* Instead of applying classification techniques, Mihalcea (2006) used knowledge-based methods to exploit available knowledge resources such as dictionaries, databases or ontologies to determine the sense of a word. These techniques are related to the lexical similarity measure (LSMs) because they often exploit the same knowledge resources. WSD techniques have been applied in a variety of tasks, in the field of information retrieval. WSD can eliminate search results in a way that at least some of the query keywords occur, but in a different sense than the given query (Schütze and Pedersen, 1995). This would lead to a reduction of false positives and hence increase the performance of the retrieval system.

Existing semantic search systems (Bonino *et al.* 2004 and Varelas *et al.* 2005) expands individual keywords through domain ontology to deal with different mismatch and ambiguity challenges such as synonymy and polysemy. For example, a search for the concept can be expanded through domain ontology to the keywords. The search, checking only for a keyword may have fewer results than the search checking for domain concepts.

2.5.1 Distributed Information Retrieval

Distributed information retrieval is more accurate, efficient and stable that it becomes an important research field for information retrieval with the development of Internet. Distributed information retrieval can be applied by the user to select the most appropriate collections from a massively distributed information sources by identifying the relevance of the collections' respect to a given query. Zhang *et al.* (2006) retrieved document collections concurrently from different collections. These are sorted and merged by relevance to the query but the single result list is formed and showed to the users. In retrieval systems, query expansion is often used to overcome a vocabulary mismatch between the query and the documents' collections.

2.5.2 Query Expansion (QE)

Xu and Croft (2000) applied query expansion in information retrieval to solve the problem of word mismatch and ambiguity that arose from differences in the words used by search engines. The users referred to the the words used by content authors to describe the same concept. However, users found it difficult to formulate query in search engines. Conesa *et al.* (2006) tried to reformulate web queries based on semantic knowledge about different application domains to expand the query.

Query expansion is viewed as bridging the gap between high-level general topics expressed by the query. It is the process of augmenting the user's query with additional terms in order to improve results by including terms that would lead to retrieving more relevant documents. Bhoga *et al.* (2007) expand the user initial query by using ontology concept in order to extract the semantic domain of a word and add the related terms to the initial query. This was related to the query only under a particular context of the specific query.

2.5.3 Approaches of Query Expansion

In this section, the detailed characteristic description of the approaches for improving the initial query formulation through query expansion and term reweighting is provided. Query Expansion (QE) needs a source of relationships to provide the connections between the query words and the relevant documents. As evaluation of these relationships plays a central role in this research work, a review on various QE schemes based on two main categories are discussed:

i. Unstructured relationships derived from a document corpus analysis

ii. Hard coded relationships from human sources, such as a thesauri or ontologies.

Figure 2.8 shows the methods or types that would be discussed in this section.

1. Corpus-Based Dependent Query Expansion

Corpus-Based Dependent has three general methods: Relevance feedback, local analysis (linguistic analysis), and global analysis:

i. *Relevance Feedback:* Daqing and Dan (2010) proposed a relevance feedback method called Translation Enhancement (TE). This uses extracted translation relationships from relevant documents to revise the translation probabilities of query terms. It also identifies extra available translation alternatives so that the

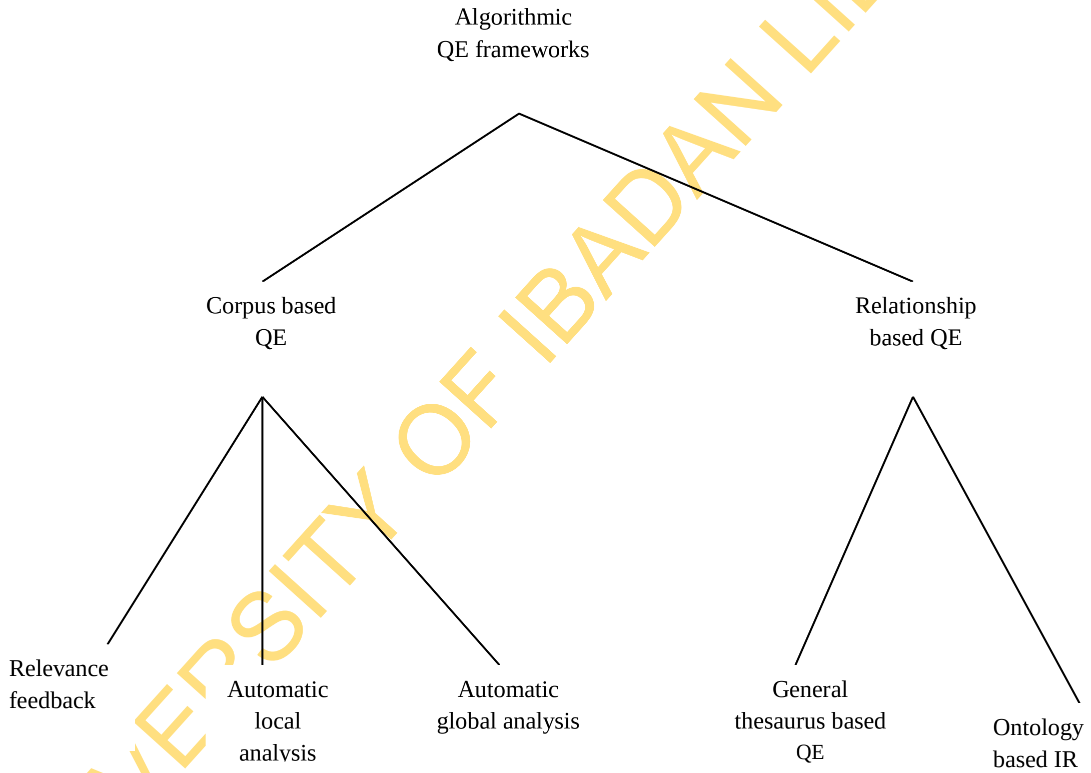


Figure 2.8: Methods of Query Expansion Algorithm

ii. translated queries are more tuned to the current search. Also, it uses user interaction to determine a few relevant documents and takes the outcomes that are returned from a given query. It uses information provided by the user to check relevance of the result and perform a new query.

iii. *Local Analysis:* Lioma and Ounis (2008) presented a syntactically-based query reformulation (SQR) technique, which is based on shallow syntactic evidence induced from various language samples. Consequently, the performance of the system was evaluated by combining pseudo relevance technique. Queries were scrutinised based on linguistic characteristics. However, this could be a challenging task as there is much depth involved.

iv. *Global Analysis:* Qiu and Frei (1993) obtained inter-term relationships from the entire corpus and expanded the query from it which was based on a similarity thesaurus. Concept space was indexed by the documents in which the terms appeared.

Additionally, this algorithm expands the query by choosing terms close in concept to the centroid of the entire query, rather than those terms close to the individual query terms.

2. External Knowledge Sources-Based Query Expansion

The relationships in corpus based QE are derived from the collocation of terms within documents. The relationship based QE in this regards uses relationships from outside sources which can be used to expand the query by solving the semantic language mismatch and ambiguity. The sources can be divided into 2 areas. These are thesauri and domain ontology.

i. *Thesauri:* Conesa *et al.* (2006) and Navigli and Velardi (2004) respectively reformulated the web queries based on semantic knowledge about different application domains from Research-Cyc and WordNet to expand the query. Some thesauri contain information about other types of relationship. Many approaches such as Bhogal’s *et al.* (2007) expanded the user initial query by using ontology in order to extract the semantic domain of a word and add the related terms to the initial query. But sometimes, these terms are not related to query terms but related to the query only under a particular context of the specific query.

ii. *Domain Ontology:* This tends to specify relationships in a more formal manner and also tends to use more of a knowledge retrieval context. To get desired information in large web environments, users must be able to formulate accurate queries efficiently. The constraints on ontology-based information retrieval are more rigid and the results more correct than those of thesaurus based QE. The concept terms of ontology can for example serve as a controlled indexing vocabulary.

Pan *et al.* (2009) applied ontology-based expansion mechanisms to reduce ambiguous queries in search. Sieg *et al.* (2007) used ontologies as the basis of the profile that allowed the initial user behaviour to be matched with existing concepts in the domain ontology concepts and relationships between the concepts.

An ontological user profile can be created and its query-related concepts would be activated. This can be achieved by matching the query with the ontological user profile. This would activate each query context' concepts and semantically relate it to concepts from the ontological user profile. The query terms would be disambiguated so that it matched to a unique ontology concept. However, a terminology can be added to the concept from the domain of reference. This method of query expansion is used in this research work.

2.6 Theoretical Framework on Information Sources

The similarity between terms or concepts can be measured by quantifying the relatedness between the words utilised in knowledge obtained from certain information sources. Zhang *et al.* (2000) measured the similarity between words in information retrieval using web documents. Navigli (2009) used semantic similarity to disambiguate word sense between words in WordNet while Kaza and Chenn (2008) improved the accuracy of semantic concepts. These information sources can be:

i. Collections of documents from web
ii. Corpus-Based Resources

iii. Thesauri and Semantic Networks

iv. Domain Ontology Knowledge

This section explores the determination of semantic similarity by a number of information sources.

2.6.1 Information Source: Multiple Document Sources from Web

Web resources provide an important source of knowledge background for similarity measures. Many researchers use web search engines' results as a resource and provide an efficient interface to the vast information. Cimiano and Staab (2004) used Google to determine relationship between pairs of concepts using Hearst pattern-based techniques. The strength is that it reduced the high cost of establishing adequate background knowledge. Indeed, the background knowledge sources are dynamically discovered and van Hage *et al.* (2005) relied on combination of online available textual sources and thesauri.

However, the page count of a given query is the estimation of the number of pages that contain the given query words while snippets are a brief window of text extracted by a search engine around the query term in a document. These two resources provided useful information regarding the context of the query. Bollegala *et al*. (2006) and Bollegala *et al.* (2007) defined semantic similarity over snippets used in query expansion, personal name disambiguation and community mining respectively. Snippets processing was more efficient compared to downloading web pages which consumed more time. Consequently, Bollegala *et al*. (2011) used page counts and snippets as two information sources provided by web search engines to generate semantic similarity results. In this thesis, both the snippets and page count are used as information sources in document collection.

2.6.1.1 Snippets Based Method

Similarity of short text snippets worked poorly with traditional document similarity measures. Sahami and Heilman (2006) addressed this problem but introduced a novel method for measuring the similarity between short text snippets by leveraging web search results to provide greater context for the short texts. Similarity kernel function defined mathematical analysis of its properties and provided examples of its efficacy. However, kernel functions have shown a large-scale system for suggesting related queries to search engine users. For each query used by Sahami and Heilman (2006), snippets were collected from a search engine and each snippet was represented as a TF-IDF feature weighted term vector.

Chen *et al.* (2006) proposed a double-checking model using text snippets returned by a web search engine to compute semantic similarity between words. Integration of semantic web data (ontology) into the enhancement of text search results enriches the snippets. Guha *et al.* (2003) developed a semantic retrieval system that pursued the goal to augment traditional search results with data pulled from the Semantic Web. Waitelonis and Sack (2009) described similar approach of augmenting information retrieval system with structured using data video search.

2.6.1.2 Web Page Counts Based Method

Cilibrasi and Vitanyi (2007) used only page counts retrieved from a web search engine as a distance metric between terms. This proposed measure used Normalised Google Distance (NGD) and is given by 2. 19:

$$NGD(A,B)=\frac{\max\{(\log H(A),\log H(B)\}-\log(A,B)}{\log(AB)-\min\{\log H(A),\log H(B)\}} \qquad (2.10)$$

where A and B are the two terms between distance NGD (A, B)

H (A) denotes the page count for the term A,

H(A, B) is the page count for the query A and B

NGD is fully based on normalised information distance using Kolmogorov complexity but NGD does not take into account the context in which the words co-occur.

2.6.1.3 Snippets and Web Page Counts Based Method

With the high growth rate of of web documents, searching increasingly becomes more difficult. Analysing each document separately extremely consumes time. Nirgude *et al.* (2013) used a method based on page count and snippets method (PCSM). This was based on a lexical pattern extraction and a pattern clustering algorithm to find semantic similarity measure between words. In order to have an effective and efficient result, web snippets need have a specified page to cluster so that dimension of feature extraction would be reduced. This method is better than other two methods because it involves multiple documents sources but used tf-idf and does not take concept of the domain into consideration.

2.6.2 Information Source: Corpus-Based Resources

This is a large collection of text documents that is used for language research and it is also used for semantic similarity measure. Furthermore, corpus-based determines the similarity between words according to information gained from large corpora. The measure provides better recall but suffers from lower precision since most of the methods rely on a simple representation (depicted in Figure 2.9). These are some of the examples of corpus information sources:

2.6.2.1 Latent Semantic Analysis (LSA)

This representation is based on the Vector Space Model (VSM) but uses the basic principle behind latent semantic analysis between the similarities of two words. This reflects the way words co-occur in language (Landauer and Dumais, 1997). LSA assumes words that are close in meaning that occurr in similar pieces of text where a matrix containing word counts is constructed from a large piece of text. LSA uses a mathematical technique called Singular Value Decomposition (SVD) which is used to reduce the number of columns while preserving the similarity structure among rows.

Mahesh *et al.* (1999) used word sense and lexical concepts for indexing and retrieval. Kwantes (2005) used SVD for representation of words that occurred in similar contexts. This did not solve the problem of co-occurrence of words. Stanislaw and Dawid (2005) and Chim and Deng (2007) presented lingo algorithm that used SVD to combine common phrase discovery and latent semantic indexing (LSI) techniques to group search results into meaningful groups. This semantic-based information retrieval system utilised LSI techniques (discussed in section 2.4.1.2.3) to enhance searches but the approach was limited by employing analysis of semantics rather than by taking different measures or inherent semantics from texts.

2.6.2.2 Normalized Google Distance (NGD)

Cilibrasi and Vitanyi (2007) derived a semantic similarity measure from the number of documents returned by the Google search engine for a set of keywords. But the keywords with the same or similar meanings in a natural language sense tend to be

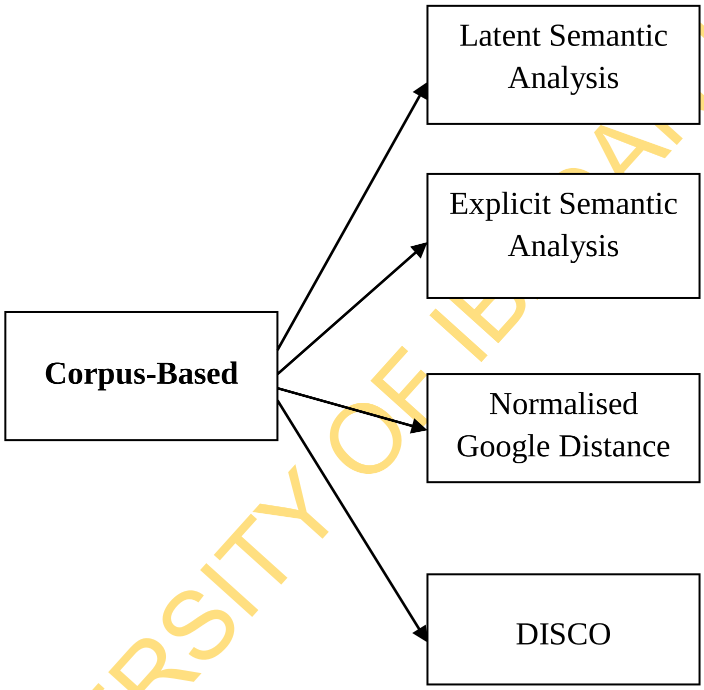


Figure 2.9: Corpus-Based Similarity Measure

close or related in Google distance. Words with dissimilar meanings tend to be farther apart and the NGD between two search terms A and B are found.

2.6.2.3 Explicit Semantic Analysis

Gabrilovich and Markovitch (2007) used Wikipedia based technique to compute the semantic relatedness between two texts which represent terms as high-dimensional vectors. The vectors of each term in Wikipedia are presented by the TF-IDF weight. However, the semantic relatedness between two terms is expressed by the similarity measure between the corresponding vectors.

2.6.2.4 Extracting DIStributional similar words using CO-occurrences (DISCO)

Peter (2009) describes distributional similarity between words and assumes that words with a similar meaning occur in a similar context. Distributional similarity between words can be statistically analysed using large text collections which can be computed using DISCO with simple context window of size ±3 words (3 n-grams) for counting co-occurrences. Lin (1998) computes similarity between two words when subjected for exact similarity. However, DISCO retrieves word vectors from the indexed data. DISCO has two main similarity measures and these are: DISCO1 and DISCO2. DISCO1 computes the first order similarity between two input words based on collocation sets while DISCO2 computes the second order similarity between two input words based on distributional similar words.

To overcome the issues of corpus-based and lexical-based techniques while maintaining the precision or enhancing precision, a semantic network-based approach to semantic similarity is used. The methods are based on linguistic knowledge and thus provide a more precise representation than co-occurrences or bag-of-word models.

2.6.3 Information Source: Semantic Network

Quillian (1968) defined semantic network as

> "*Semantic network is broadly described as any representation interlinking nodes with arcs, where the nodes are concepts and the links are various kinds of relationships between concepts*".

The concepts extracted are used to disambiguate regarding to the context of the document (Baziz *et al*., 2004). Maki *et al.* (2004) obtained semantic similarity or distance on the basis of WordNet to explain human similarity judgments independently of associative strength, lexical co-occurrence or feature similarity.

WordNet is a research project at Princeton University (Fellbaum, 1998**)** and a large lexical database of English. Ferrer-i-Cancho (2005) defined WordNet as word meaning and models. It is also defined as meaning-meaning associations which can be used as both a thesaurus and a dictionary. But WordNet senses are in form of nouns, verbs, adverbs and adjectives which were organised by a variety of semantic relations into synsets. A fragment of the WordNet is shown in Figure 2.10 as IS-A hierarchy.
WordNet is a lexical analyser used in natural language processing which contains around 150000 synsets and semantic relations. The synsets are also organised into synonym sets corresponding to different term or concept with the same meaning. These can also be organised as autonomy (opposite), hypernym (super-concept)/hyponymy (sub-concept) (also called IS-A hierarchy / taxonomy), meronymy (part-of) and homonymy (has-A). WordNet is used to compute the similarity score and can be seen as ontology for natural language terms.

Ozcan and Aslangdogan (2005) extended each concept with similar words using a combination of Latent Semantic Analysis (LSA) and WordNet (Fellbaum, 1998) but the test performance showed a promising result for short or poorly formulated queries. Ciorascu *et al.* (2003) focused on some approach using ontology to enrich query processing. However, ontology in these cases typically served as thesauri contained synonyms, hypernym /hyponyms and did not consider the context of each term relations.

2.6.4 Information Source: Domain Ontology Knowledge

WordNet has many synsets and a particular synset may have more than one sense. But word sense disambiguation results in a single decision. For example, assume a user inputs a keyword query, ‘‘Soap’’, conventional information retrieval systems retrieve thousands of snippets where soap might be used as (a) a detergent, (b) a weekly television programme, (c) a service oriented architecture (d) and a simple object access protocol.

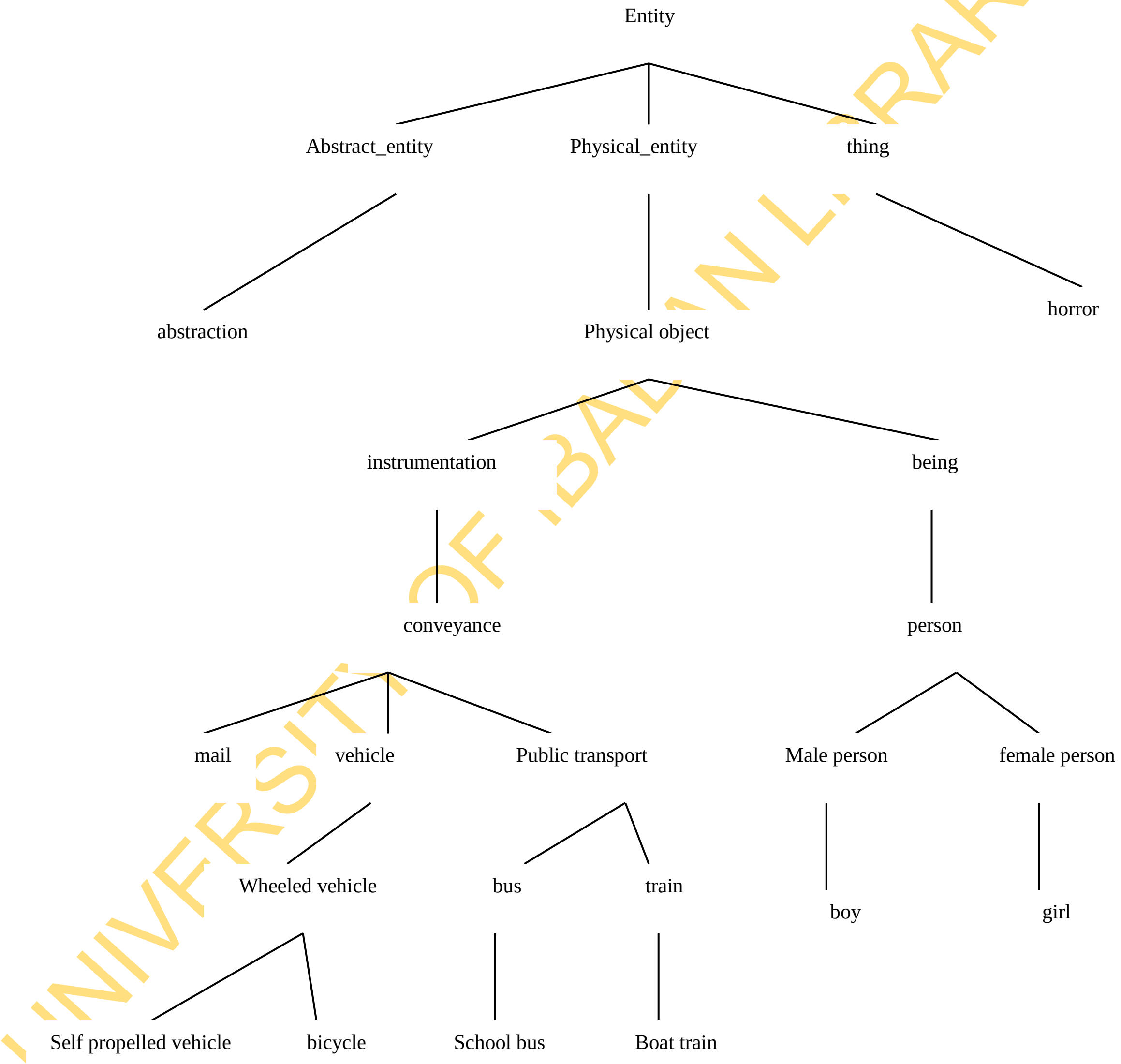


Figure 2.10: A fragment of WordNet Relations in different Domains

At times none of the search results may be relevant to a user's request or it may even give two or more meanings to the keywords in the same domain as in c and d. The systems returned less relevant answers for the query although it expands individual keywords in a query with different semantic relationships. However, more relevant documents for a keyword query can be retrieved if systems know the meanings and relationships that exist among the keywords in the query. But with keyword structure, a combination of at least two concepts and its relationship that exists in the domain ontology will appear in the retrieved snippets.

A hierarchical structure can represent the context that is, circumstances in which something happens or should be considered. However, various approaches have been used to quantify the similarity between concepts in ontology while still maintaining information contained in the hierarchical structure (Schickel-Zuber and Falting, 2007). Therefore, the existing systems (Varelas et al. 2005; Rinaldi, 2009; Alipanah et al., 2010; and Yang and Wu, 2011) cannot resolve the semantic issues of polysemy or synonyms because they require identification of the context of keywords to comprehend their actual semantics. Moreover, the existing systems also ignore other important relationships such as semantic neighbourhoods (Rodriguez and Egenhofer, 2003) that can also contribute to useful search results.

To overcome the limitations of existing semantic searching systems, one needs to represent the context of terms through *IS-A* hierarchy for effective searching using domain knowledge (Poole and Campbell, 1995 and Khan and Marvon, 2006). With domain ontology, a particular sense si chosen based on IS-A hierarchy concept by relating it to the actual domain concepts. The system concentrates on searching terms using IS-A hierarchy and not on the individual keywords.

Paralic and Kostial (2003) proposed an ontology-based approach to information retrieval where document resources are associated with concepts in ontology. They focused on query processing where concepts were matched to corresponding concepts in the ontology. The query concepts were matched with the document concepts and matched documents were retrieved.

The existing systems focused on matching the semantic similarity on individual keywords while a typical semantic search system (Varelas *et al.* 2005 and Fang *et al.* 2005) expanded individual keywords through domain ontology. This deals with different semantic language mismatch and ambiguity challenges such as polysemy and synonymy. Domain ontology provides a conceptual framework for the structured representation of context through a common vocabulary in a particular domain (Fang *et al.* 2005). Tomassen (2009) presented method of indexing documents with ontology vocabulary based on the Vector Space Model. The index terms derived from the ontology are adapted to the domain terminology. When compared to the vector model (TF-IDF), the Latent Semantic Indexing (LSI) and WordNet approaches, domain ontology-based approach performed significantly better.

Supervised machine learning techniques can be used to determine how to combine concepts and their semantic relations into a document similarity measure. However, concept document representations can solve the problem of language mismatch and the ambiguity by expanding the representation to incorporate concepts from a document (Bloehdorn & Hotho, 2004).

2.7 Theoretical Framework on Term-Based Similarity

Text representation, categorisation, clustering and other applications are at the crossroads of information retrieval and machine learning. Therefore, no matter which indexing unit is used, each term in a document vector must be associated with a value (weight) which measures the importance of the term. It denotes how much this term contributes to the classification of the document.

Newman and Girvan (2004) used the weighting measures for the information retrieval and text analysis but documents are presented in high dimensional space. This depends on the number of indexing terms that are chosen to be relevant for the collection. These are quite sparse and most coordinates are zero which is stored as classical vectors. Retrieval by classical information retrieval models, for example, Vector Space, Probabilistic and Boolean models are based on lexicographic term matching (Baeze-Yates and Ribeiro-Neto, 1999). Every unique term or concept from the documents collection is analysed and forms a separate dimension. Each document is represented by a vector space with dimensions.

For example: Vector v represents document j in a k dimensional space Ω, then the component t of vector v where $i \in [1...k]$ represents the degree of the relationship between document j and a term corresponds to dimension i in Ω .

Element a_{ij} of matrix A is therefore a numerical representation of relationship between term i and document j . There were many methods for calculating term weighting (a_{ij}). This relationship can be best expressed as a term-to-document matrix $A\,(t \times d)$ where t is the number of a unique terms and d is the number of documents as in Table 2.1. There were many methods for calculating term weighting (a_{ij}). This relationship can be best expressed as a term-to-document matrix $A\,(t \times d)$ where t is the number of a unique terms and d is the number of documents.

Let assume the following documents are retrieved from a query.

D1: A search engine for 3D Models

D2: Implementation of a string database query languages

D3: Ranking of documents by measures considering conceptual dependence between terms

After applying preprocessing techniques on the documents, the terms are broken into tokens and these are:

T1: search *T2*: engine *T3:* model *T4:* implementation *T5:* database

T6: query *T7:* language *T8:* document *T9:* measure *T10:* conceptual

T11: dependence

The matrix can be formed by comparing each term with document as it appears in the document.

Term weighting is usually solved by means of methods from text search, that is, methods that do involve a training set. Term weights can be computed as real-valued $0 \leq w_{kj} \leq 1$, when weights are non-binary. For binary value, the weight $(wt_{kj}) \in (0,1)$ indicates presence/absence of the term in the document. The normalisation of document vectors is applied during the index generation phase to make the calculation in the retrieval phase faster. The query vector of the document can be defined as follows:

Table 2.1: Term by Document Matrix

t/d	t_1	t_2	• • •	t_n
d_1	d_1t_1	d_1t_2	• • •	d_1t_n
d_2	d_2t_1	d_2t_c	• • •	d_1t_n
•	▬	▬	• • •	▬
•	▬	▬		▬
•	▬	▬		▬
d_n	d_nt_1	d_nt_2	• • •	d_nt_n

Definition 2.5: Let t_i be an index term, d_j be a document and $w_i \geq 0$ be a weighted pair (t_i, d_j). The weights described the semantic content of the document. The vector for a document d_j is represented by $\overrightarrow{\mathrm{d}}_{\mathrm{j}} = (w_{1,j}, w_{2,j}, \cdots, w_{t,j})$, where t is the total number of index terms in the system. The index terms in the query were weighted.

Let $w_{i,q}$ associated with the pair $[t_i, q]$. Then the query vector $\overrightarrow{\mathrm{d}}_{\mathrm{j}} = (w_{1,q}, w_{2,q}, \cdots, w_{t,q})$.

With term weighting, documents are presented in high dimensional space. However, lexical matching methods can be inaccurate when used to match user's query.

Polysemy terms (words having multiple meaning) in a user's query would literally match terms in irrelevant documents whereas synonym (a system in which multiple words have the same meaning) could lead to a situation in which the literal terms in a user's query are not matching those of a relevant term. This fundamental problem results in inefficient in information retrieval. Therefore, a concept weighting that is based on projection of vectors of concept in a document is presented to eliminate the limitation of classical vector.

Similarity (Distance) Measures

Similarity measures represent the similarity between two documents, two queries or one document and one query. Similarity or distance measures need to be determined in order to reflect the degree of closeness or separation of the target objects. Similarity measures can be evaluated by studying retrieval performance in terms of precision and recall in a particular application domain.

Similarity between a query q and a document d or similarity between documents to documents can be computed using different measures to normalise the vectors. Such a vector is then normalised to unit length and stored in form of term by document $(t \times d)$ or document-to-document $(d \times d)$ matrix depending on the indexing method the user actually wants.

However, not every similarity measure is a metric; however similarity measure must satisfy the following four conditions:

Let a and b be any two objects in a set and $S(a,b)$ would be the similarity or distance between a and b

i. The similarity between any two points would be non-negative: $S(a,b) \geq 0$

ii. The similarity between two objects would be one (1) if and only if the two objects are identical, that is, $S(a,b)=1$ if and only if $a=b$

iii. Similarity would be symmetric, that is, distance from a to b would be the same as the distance from b to a , i.e. $S(a,b)=S(b,a)$

iv. The measure must satisfy the triangle inequality, which would be $S(a,c)\leq S(a,b)+S(b,c)$

The similarity between two text documents A and B can be easily computed. A variety of similarity or distance measures has been proposed and widely applied using term similarity measures in Figure 2.11. As shown, some of such measures are Jaccard, Correlation Coefficient, Euclidean Distance, Block Distance, Matching Coefficient, Dice Coefficient, Cosine Similarity and Radial Basis Function (non-linear) etc.

i. Jaccard Coefficient

The Jaccard Coefficient (Tanimoto Coefficient) is a statistical measure of the extent of overlap between two vectors. It measures similarity as the intersection divided by the size of the union of the vector dimension sets. For text documents, the Jaccard coefficient compares sum weight of shared terms to the sum weight of terms that are present in the two documents. Kim and Choi (1999) analysed term similarity due to its simplicity and retrieval effectiveness but did not consider term frequency and rare term in a document collection. The definition is as follows:

$$SIM_J=(\overrightarrow{d_a},\overrightarrow{d_b})=\frac{\overrightarrow{d_a}\bullet\overrightarrow{d_b}}{\left|\overrightarrow{d_a}\right|^2+\left|\overrightarrow{d_b}\right|^2-\overrightarrow{d_a}\bullet\overrightarrow{d_b}} \tag{2.11}$$

where d is the document set a and b, SIM is similarity

The Jaccard coefficient (J) is a similarity measure values ranges between 0 and 1. If it is 1 then the $\overrightarrow{d_a}=\vec{d}_b$ and 0 when $\overrightarrow{d_a}$ and $\vec{d}_b$ are disjointed, where 1 means the two objects are the same and 0 means it is completely different. The corresponding distance measure is $D_J=1-SIM_J$.

ii. Overlap Coefficient

The overlap coefficient (Szymkiewicz-Simpson coefficient) is a similarity measure related to the Jaccard coefficient that measures the overlap between two sets.

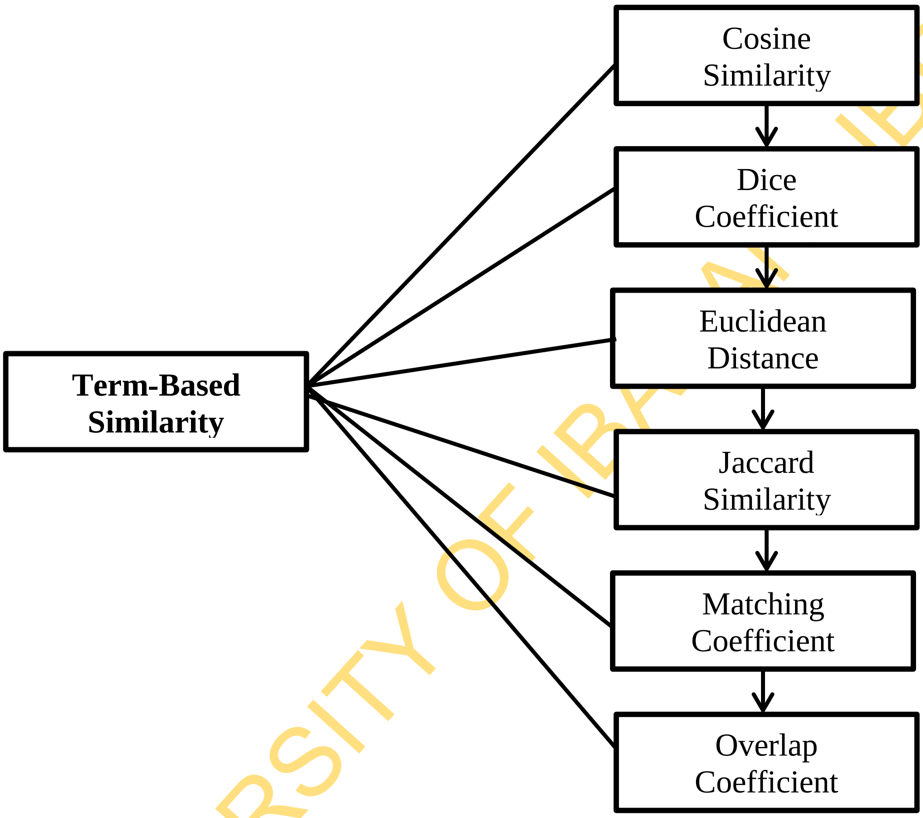


Figure 2.11 Term-Based Similarity Measures

It is defined as the size of the intersection divided by the smaller of the size of the two sets but considers two strings a full match if one is a subset of the others.

$$\text{Overlap}\,(a,b) = \frac{|a \cap b|}{\min|a| \cdot |b|} \tag{2.12}$$

If the set a is a subset of b or the converse, then the overlap coefficient is equal to 1.

iii. Manhattan (Block) Distance

Eugene (1987) proposed a distance measure between two points along axes at right angle in a plane with p_1 at $(a_{1,}b_1)$ and p_2 at $(a_{2,}b_2)$. Manhattan distance returns the maximum absolute difference in coordinates which corresponds to D = 1.

$$manh(a,b) = |a_2 - a_1| + |b_2 - b_1|$$

Therefore, it can be represented in form of weight (w) as:

$$manh(\vec{w}_a - \vec{w}_b) = \sum_{j-1}^{d} |w_{a,j} - w_{b,j}| \tag{2.13}$$

iv. The Dice Coefficient

Dice (1945) measured intersection between two sets scaled by size giving a value in the range 0 to 1.

$$Dice(a,b) = \frac{2|a \cap b|}{|a| + |b|} \tag{2.14}$$

v. Euclidean Distance

Euclidean distance is a standard metric for geometrical problems. It is the distance between two points and can be easily measured with a ruler in two or three-dimensional space. Euclidean distance is used in clustering problems. For example, K-means algorithm measured distance between text documents but large for vectors of different lengths. The Euclidean distance between $\vec{d_a}$ and $\vec{d_b}$ is large even though the distribution is very similar.

Given two documents d_a and d_b represented by term vectors $\vec{t_a}$ and $\vec{t_b}$ respectively, and weight (w), then Euclidean distance (*D*E) of the two documents is defined as:

$$D_E(\vec{d_a}, \vec{d_b}) = (\sum_{t=1}^{m} |w_{d_a} - w_{d_b}|^2)^{1/2} \tag{2.15}$$

Where the term set is $T = \{t_1,...t_m\}$ as mentioned above. The *tfidf* feature selection can be used in Euclidean term weights.

vi. Linear Kernel Function Similarity

Linear Kernel's measures of similarity is such that it calculates the dot product of two vectors $s(a,b) \succ s(a,c)$ if objects *a* and *b* are more similar than object a and c, then a kernel is positive. The function linear kernel is a polynomial kernel with a degree =1 and coefficient =0 (homogeneous). If a and b are column vectors, weight (w) and document (d) then linear kernel (k) is define as:

$$k(d_a, d_b) = \sum (w_a)^T w_b \qquad (2.16)$$

It does not consider the optimisation problem and the computation becomes increasingly expensive with increasing simple size.

vii. Radial Basis Function (RBF)

The RBF is a non linear measure and it is used to map the data onto infinite dimensions. It computes the vector between two vectors. The minus sign in 2.17 inverts the distance measure into a similarity score due to its exponential. The similarity ranges from 1 to 0. RBF is applied in many science and engineering fields. For document (*d*) a and b, γ is gama and weight (*w*). The kernel (k) is defined as:

$$k(d_a, d_b) = \exp(-\gamma \|w_a - w_b\|^2) \qquad (2.17)$$

where

$\|w_a - w_b\|^2$ is the square of the Euclidean distance $\sum_{t=1}^{i} \left| w_{d_a} - w_{d_b} \right|^2$) between two a and b vectors.

RBF has few basic functions that cannot fit the training data adequately due to limited flexibility. On the other hand, those with too many basic functions yield poor generalisation abilities because of the limited flexibility of the RBF and its ability to erroneously fit the noise in the training data.

viii. Cosine Similarity

Documents are represented as term vectors. The similarity of two documents corresponds to the correlation between the vectors. This is quantified as the cosine of the angle between vectors, that is, cosine similarity. Larsen and Aone (1999) presented cosine similarity as one of the most popular similarity measures applied to text documents in information retrieval applications and clustering. Similarity between a and b and Weight (w) is defined as:

Cosine Similarity (A,B) = $\frac{|a \cap b|}{\sqrt{|a|} \bullet \sqrt{|b|}}$

$$\text{Cosine } (W_a, W_b) = \frac{\sum_{j=1}^{n}(W_{aj} * W_{bj})}{\sqrt{\sum_{j=1}^{n} W_{aj}^2}\sqrt{\sum_{aj=1}^{n} W_{bj}^n}} \qquad (2.18)$$

When two same copies of document d for example, are combined to get a new pseudo-document d' , the cosine similarity between d and d' is 1. This means that these two documents are regarded to be the same as illustrated in the Figure 2.12.

The purpose of normalisation is to make similarity of each element in a vector to be in the same range so that individual element gets the same weight when measures are applied. Vectors are normalised by sizes.

$$\|\vec{x}\|_2 = \sqrt{\sum_i x_i^2} \qquad (2.19)$$

where $1 \leq i \leq n$

Given two documents d_a and d_b, the cosine similarity is:

$$SIM_C = (\vec{d_a}, \vec{d_b}) == \frac{\vec{d_a} \bullet \vec{d_b}}{|\vec{d_a}| * |\vec{d_b}|} \qquad (2.20)$$

Where $\vec{d}_a$ and $\vec{d}_a$ are n-dimensional vectors over the term set $T = \{t_{1,\ldots,} t_n\}$, each dimension represents a term with its weight in the document. Cosine similarity is non-negative and bounded between [0, 1].

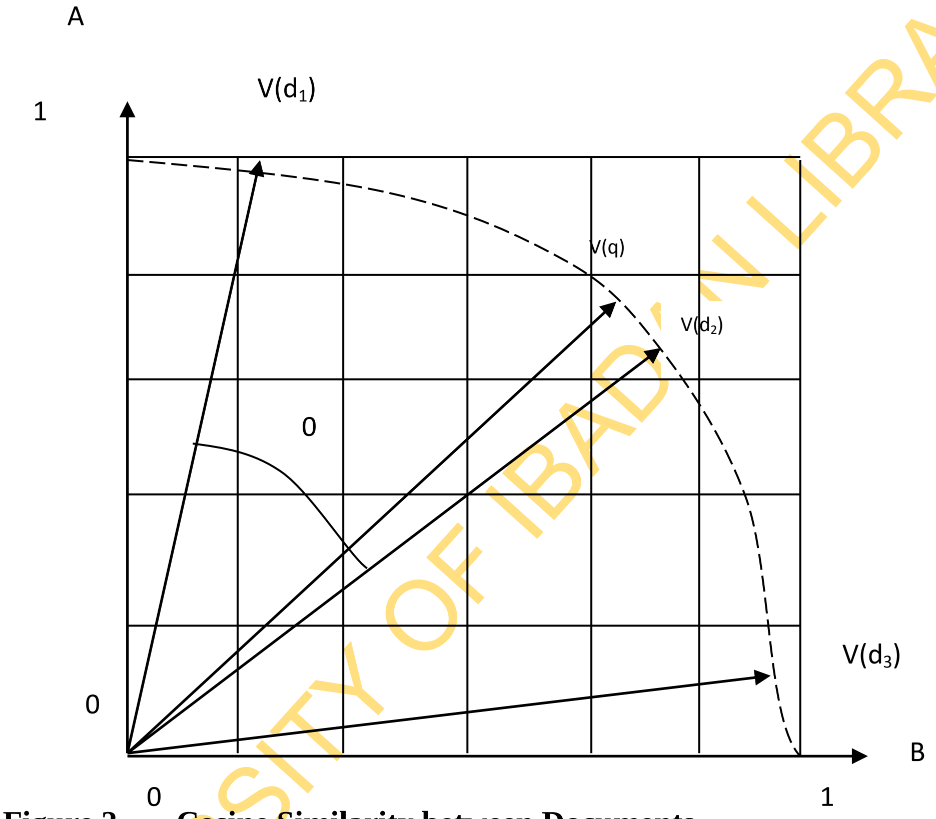


Figure 2. Cosine Similarity between Documents

The algorithm below describes computation of Cosine measure.

Algorithm for Computing Cosine Scores

```
float $scores[n] = 0$
float $length[n]$
for each query term $t$
do calculate $w_{t,d}$ and get list for $t$
    for each pair $(d, tf_{t,d})$ in list
    do $scores[d] += w_{t,d} * w_{t,d'}$
read the array length
for each $d$
do $scores[d] = scores[d] / length[d]$
return K components of $scores[]$
```

2.8. Theoretical Framework on Knowledge-Based Similarity

Finding similarity plays an important stage of text similarity. Similarity can be in two ways. These are lexical and semantic similarities. Lexical similarity can be done by different string term-based similarities (discussed in section 2.7) while semantic similarity is done by corpus-based and knowledge-based algorithms. However, various approaches have been used to quantify the similarity between concepts in ontology while still maintaining information contained.

Pedersen *et al.* (2004) developed software called “WordNet::Similarity” that measures the similarity of concepts using different measures that used dictionary definition. This programme is used to compute conceptual similarity of words. Turney (2006) measured semantic similarity between words or concepts based on features of concepts and this plays an important role in many research areas such as Artificial Intelligence (AI), Natural Language Processing (NLP), cognitive science and knowledge engineering.

Pirrò (2009) and Hirst and Budanitsky (2006) considered semantic networks as better choices for estimating semantic similarity than other lexical resources. However, some of the most popular semantic similarity methods in Figure 2.13 are implemented and evaluated using WordNet as the underlying reference ontology. Patwardhan *et al.*

(2003) used semantic similarity measures with WordNet to enrich ontology with information about its leaf-nodes for disambiguation. However, disambiguation provides a small ranked list of WordNet-senses for each leaf node in the ontology hierarchies. These WordNet-senses are good candidates for the description of node as a whole or in parts.

Based on the WordNet utilisation, semantic similarity or distance measures between two concepts or words in any application can be classified into four categories as illustrated Figure 2.13. The categories are:

1. Path length based measures
2. Information Content based measures
3. Feature based measures and
4. Hybrid measures.

2.8.1 Path Length Based Measures

The path length measures the similarity between two concepts as a function of the length of the path linking the concepts and the position of the concepts in the taxonomy. It uses link or edge as parameter to refer to the relationships between concept nodes. The path length can be categorised into:

i. *The Shortest Path Based Measure:* The measure only takes $len(c_1, c_2)$ into consideration. Knappe *et al.* (2002) assume that the $sim(c_1, c_2)$ depends on how close the two concepts are in the taxonomy and measures variant on the distance method. It is based on how observations of the behaviour of conceptual distance resemble that of a metric. Varelas *et al.* (2005) described the conceptual distance between two nodes and proportional to the number of edges separating the two nodes in the hierarchy.

For concept A and B in WordNetSimilarity, the following similarities are:

$$sim_{path}(c_A, c_B) = 2 * depth\_\max - len(c_A, c_B) \tag{2.21}$$

From 2.21, the similarity between two concepts (c_A, c_B) is the function of the shortest path $len(c_A, c_B)$ from c_A to c_B.

ii. *Wu & Palmer's Measure:* Wu and Palmer (1994) introduced a scaled measure. This similarity measure takes the position of concepts c_A and c_B in the taxonomy

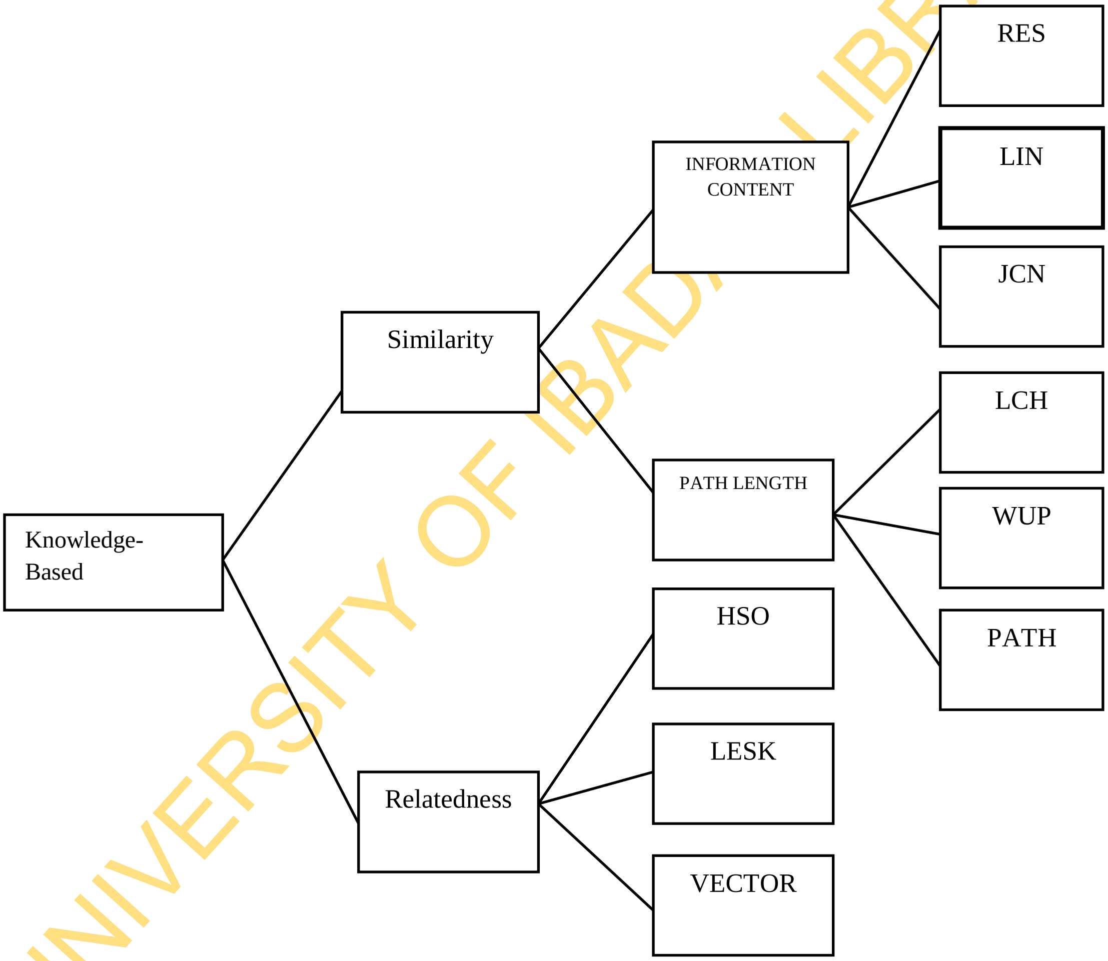


Figure 2.13: Knowledge-Based Similarity (adapted by Gomaa and Fahmy, 2013)

relatively to the position of the least common subsumer concept ($lcs(c_A, c_B)$) into account.

It assumes the similarity between two concepts as the function of path length and depth in path-based measures.

$$sim_{wp}(c_A, c_B) = \frac{2 * depth(lcs(c_A, c_B))}{len(c_A, c_B) + 2 * depth(lcs(c_A, c_B))} \tag{2.22}$$

From 2.22, the similarity between two concepts (c_A, c_B) is the function of the distance and the least common subsumer $lcs(c_A, c_B)$. It is not a similarity measure but a distance measure.

iii. *Leakcock & Chodorow's Measure:* Leakcock and Chodorow (1998) proposed the maximum depth of taxonomy and it has the following measure:

$$sim_{LC}(c_A, c_B) = -\log \frac{len(c_A, c_B)}{2 * deep\_\max} \tag{2.23}$$

From 2.23, the similarity between two concepts (c_A, c_B) is the function of the shortest $len(c_A, c_B)$ from c_A to c_B. The measure is based only on the positions of the concepts in the taxonomy but it assumes the links between concepts and represents its distances. All the paths have the same weight. However, it notes that the density of concepts throughout the taxonomy is not constant.

2.8.2 Information Content-Based Measure

Information Content (IC) assumes that each concept is associated with much information in WordNet. Resnik (1995) proposed an information-based statistic method which was based on the Information Content (IC) of each concept. The more common information two concepts share, the more similar the concepts are. This solved the problem to find a uniform link distance in path length based methods.

Saruladha *et al.* (2011) used information content to determine the common concepts and presented the common information content by finding the common features of the compared entity classes. This attempts to exploit the information contained to evaluate the similarity between the pairs of concepts. However, matching (term) similarity based on linguistics is considered as analysing entities in isolation while ignoring the relationships with other entities. It was defined as:

$$IC(c) = \frac{\log(depth(c))}{\log(deep\_\max)} * (1 - \frac{\log(\sum_{a \in hypo(c)} \frac{1}{depth(a)} + 1)}{\log(node\_\max)} \qquad (2.24a)$$

For a given concept c, a is a concept of the taxonomy, which satisfies $a \in hypo(c)$ If c is root, deep(root) is 1 and log(deep(c)) is 0. If c is a leaf, $hypo(c)$ is 0. Then

$$\sum_{a \in hypo(c)} \frac{1}{depth(a)} = 0$$

$$IC(c) = \frac{\log(depth(c))}{\log(deep\_\max)} \qquad (2.24b)$$

i. *Resnik's Measure:* Resnik (1995) proposed information content (IC) based similarity measure. It assumes two concepts where the similarity depends on the information content that is subsumed in the taxonomy. In Resnik's measure, taxonomy of noun concepts in information content is calculated using the noun frequencies of each concept.

$$sim_{\mathrm{Resnik}}(c_A, c_B) = -\log P^n(lcs(c_A, c_B)) = IC(lcs](c_A, c_B)) \qquad \mathbf{(}2.25)$$

From 2.25, the values only rely on concept pair's lowest subsumer in the taxonomy. Resnik similarity has the problem of concept pair with the same lcs resulting in the same similarity values.

ii. *Lin's Measure:* Lin (1998) proposed similarity measure based on information content and used both the amount of information needed to state the commonality between two concepts and the information needed to fully describe these terms/concepts.

$$sim_{Lin}(c_A, c_B) = \frac{2 * IC(lcs(c_A, c_B)}{IC(c_A) + IC(c_B)} \qquad (2.26)$$

From 2.26, the measure has taken the information content of compared concepts into account and the values of this measure vary between 1 and 0. The length or distance between each concept in taxonomy is not considered.

iii. *Jiang's Measure:* Jiang and Conrath (1997) calculated semantic distance derived from the edge-based notion of distance with the addition of the information content as a decision factor to obtain semantic similarity.

According to Pirrò (2009), Hirst and Budanitsky, (2006) and Jiang and Conrath it provided the best results when measuring semantic relatedness.

$$dis_{J\&C}(c_A, c_B) = IC(c_A) + IC(c_B) - 2IC(lcs(c_A, c_B) \qquad (2.27)$$

From 2.27, the measure has taken the IC of compared concepts into account and the value is semantic distance between two concepts not semantic similarity.

2.8.3 Feature-Based Measure

The feature-based measure is independent on the taxonomy and the subsumer of the concepts, although it attempts to exploit the properties of the ontology concepts to obtain the similarity values. This was based on the assumption that each concept is described by a set of words indicating its properties or features, such as definitions or glosses in WordNet. The more common characteristics two concepts have the less non-common characteristics and the more similar the concepts are.

Tversky (1977) argued that similarity is not symmetric and features between a subclass and its superclasses have a larger contribution to the similarity evaluation than those in the inverse direction. However, Tversky (1977) defined similarity as:

$$sim_{Tversky}(c_A, c_B) = \frac{|c_A \cap c_B|}{|c_A \cap c_B| + k|c_A / c_B| + (k-1)|c_B / c_A|} \qquad (2.28)$$

Where c_A, c_B correspond to description sets of concept c_A and c_B respectively, k is adjusted and $k \in [0,1]$.

From 2.28, the values of $sim_{Tversky}(c_A, c_B)$ vary from 0 to 1 and $sim_{Tversky}(c_A, c_B)$ increases with commonality and decreases with the difference between the two concepts.

2.8.4 Hybrid Measure

Rodriguez and Egenhofer (2003) presented hybrid measures that combined both the ideas of the methods and the relationship such as IS-A, part-of etc. in the taxonomy.

Dong *et al.* (2009) used similarity function based on three parts, synonyms sets, neighbourhoods and features. The similarity value of each part is assigned to a weight and summed together. Information content based measures and path based measures as parameter are commonly used. Mihalcea *et al.* (2006) used multiple similarity measures using corpus-based measures and the six others which were knowledge-based. These were evaluated separately. However, Zhou *et al.* (2008) proposed a measure expressed as:

$$sim_{zhou} = (c_A, c_B) = 1 - k\left(\frac{\log(len(c_A, c_B) + 1}{\log(2 * (deep\_\max - 1))}\right) - (i - k) * ((IC(c_{A)} + IC(c_B) - 2 * Ic(lcs(^{(c_A, c_B))}\!\big/_2 \qquad (2.29)$$

As shown in 2.29, both IC and path have been taken into consideration. Parameter k was adapted manually for good performance. If k=1, 2.29 is path-based; if k=0, 2.29 is IC-based measure. The measure is semantic relatedness not semantic similarity between concepts.

2.9 Evaluation of Similarity Metrics

In this section, machine learning techniques were used to determine the similarities metrics of each of the method and the integration of the methods.

2.9.1 Evaluation Using Machine Learning Techniques

Machine Learning (ML) has several applications but data mining is the most important aspect. Mistakes made during analyses or when trying to establish relationships between multiple features are prone to error. This makes it difficult to find solutions to certain problems. Machine learning can be successfully applied to these problems by improving the efficiency of systems and the designs of machines. Every instance in the dataset used by machine learning algorithms is represented using the same set of features. The features may be continuous, categorical or binary. There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends. These two forms are as follows:

i. *Classification:* Classification problems admit only discrete and unordered values as the output of instances but the models predict categorical class labels. The choice of the learning algorithm involves the use of critical step. The classifier maps from unlabelled instances to classes for routine use and the evaluation are based on prediction accuracy. There are at least three techniques that were used to calculate a classifier's accuracy. The training is split into two sets in which the two-thirds are used for training and the other one-third is for estimating performance. There is also a technique known as cross-validation in which each subset of the classifier is trained on the union of all the other subsets. The average of the error rate of each subset is therefore an estimate of the error rate of the classifier.

ii. *Prediction:* Prediction is at the heart of every scientific discipline and the study of generalisation from data is centred on machine learning, statistics and data mining. The models predicted continuous valued functions. Machine learning and statistical methods are used throughout the scientific world for handling the information overload that characterises the current digital age. Numeric prediction is interpreted as prediction of a continuous class.

Nitish *et al.* (2012) presented an approach that combines corpus-based semantic relatedness measure over the whole sentence with the knowledge-based semantic similarity scores. The scores as features were fed to machine learning models like linear regression and bagging models to obtain a single score given the degree of similarity between sentences.

2.9.2 Tool for Implementing Machine Learning Technique: WEKA

This is a tool used for implementing machine learning techniques. It is an open source of many data mining and machine learning algorithms including pre-processing on data classification, clustering and association rule extraction. It was created by researchers at the University of Waikato in New Zealand. It is Java-based and *WEKA* is an environment for comparing learning algorithms. Researchers can implement new data mining algorithms to add in WEKA. However, WEKA is the best-known open-source data mining software in which data can be imported from a file in various formats. The formats are as follows:

i. CSV: Comma Separated Values (text file)

ii. C4.5: A format used by a decision induction algorithm C4.5, requires two separated files: Name file: defines the names of the attributes and Date file: lists the records (samples)

iii. Binary

iv. Data can also be read from a URL or from an SQL database (using JDBC)

v. ARFF (Attribute Relation File Format) has two sections: the Header information defines attribute name, type and relations while the Data section lists the data records.

However, ARFF can be created by file using Notepad or Word. ARFF consists of two distinct sections:

i. *Header* section defines attribute name, type and relations, starts with a keyword.

@Relation<data-name>

@attribute <attribute-name><type> or {range}

ii. *Data* section lists the data records, starts with

@data <list of data instances>

iii. Any line that starts with % is the comments.

2.9.3 Predictive Analysis

The approaches and techniques used to conduct predictive analysis can broadly be grouped into regression techniques and machine learning techniques. Regression algorithms were used to build a model that makes numeric predictions based on numeric values. The algorithms such as linear regression, support vector machines for regression (SVMreg) can be used for the predictions (Smola and Schölkopf, 2004). Examples are as follows:

i. *Linear Regression:* This is a prediction technique used when the class and all attributes are numeric. It is one of the easiest techniques used and it is bounded by linearity. If data exhibits a linear dependency, the best-fitting straight lines are found, where “best” is interpreted as the least mean-squared difference. In linear regression, the class is expressed as a liner combination of the attributes, each of which has the following weight (w) for a set a{1…k}:

$$A = w_0 + w_1 a_1 + w_2 a_2 +,,,++ w_k a_k \qquad 2.30$$

The goal in linear regression is to choose the weights that will minimise the sum of the squares of the difference between the predicted class value and the actual class values in the dataset.

ii. *Neural Network:* This belongs to the group of feed-forward neutral networks. The configuration variations of Multilayer Perceptron networks as shown in Figure 2.14.

This was determined by the weight vector and it is necessary to adjust the weights of the network. The weight was adjusted by an iterative process. Small changes in the weight got the desired values by the process called training the net and was done by the training set (learning rule).

$$net_j = \sum_i w_{ij} x_{ij} = w_{0j} x_{0j} + w_{1j} x_{1j} \_ ... w_{nj} x_{nj} \qquad 2.31$$

iii. *Support Vector Machine Regression (SVMReg):* It is applied not only to classification problems but also to the regression. Still, it contains all the main features that characterise maximum margin algorithm. The capacity of the system can be controlled by parameters that do not depend on the dimensionality of feature space. It relies on defining the loss function that ignores errors. In SVM regression, the input ω mapped onto a m-dimensional feature space using some fixed (nonlinear) mapping and a linear model is constructed in feature space. Using mathematical notation, the linear model $f(x,\omega)$ is given by:

$$f(x,\omega) = \sum_{i=1}^{m} \omega_i g_i(x) + b \qquad 2.32$$

where $g_i(x) = i = 1,...m$ denotes a set of nonlinear transformations and b is the "bias" term.

iv. *Bagging (Bootstrap Aggregating):* This is an ensemble method that creates separate samples of the training dataset and creates a classifier for each sample. The results of these multiple classifiers are then combined. The trick is that each sample of the training dataset is different giving each classifier that is trained a subtly different focus and perspective on the problem.

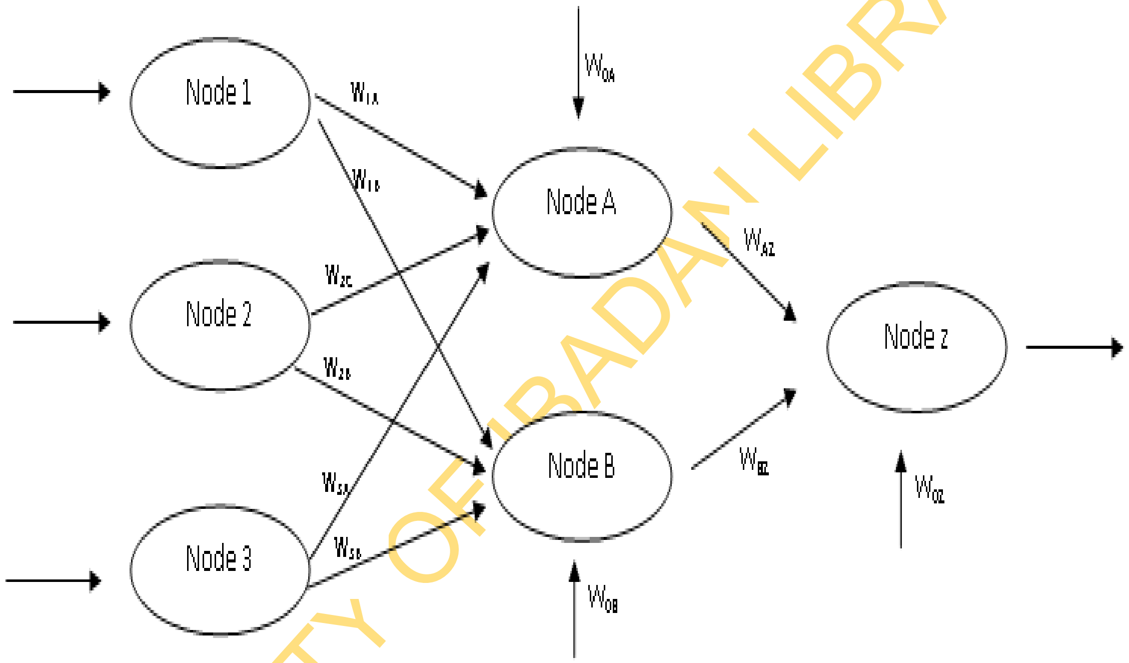


Figure 2.14: Neural Network Model

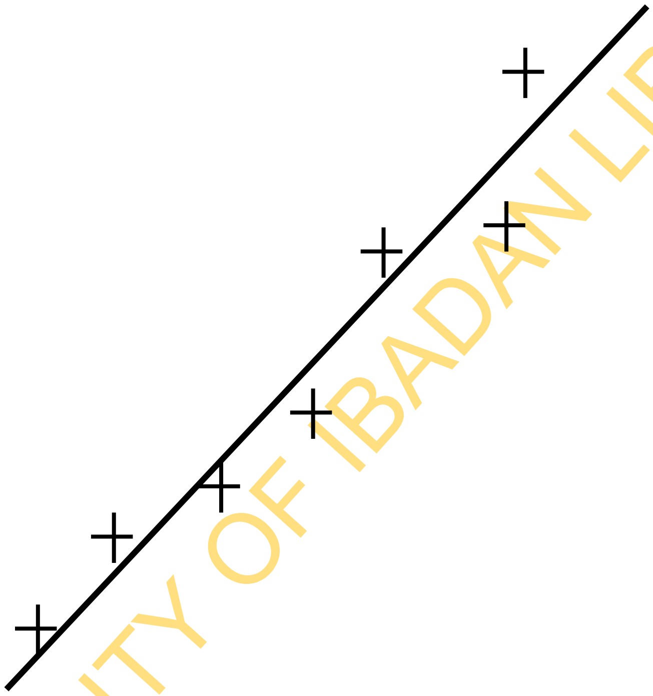


Figure 2.15: Support Vector Machine Regression

v. *M5P Model Tree:* The M5P combines a conventional decision tree with the possibility of linear regression functions at the nodes. First, a decision-tree induction algorithm is used to build a tree but instead of maximising the information gained at each inner node, each attribute at that node is tested by calculating the expected reduction in error. The attribute that is chosen for splitting maximises the expected error reduction at that node as depicted in Figure 2.16

2.9.4 Prediction Metrics

Cross-validation is one of the most useful techniques to evaluate different combinations of feature selection, dimensionality reduction, and learning algorithms. There are multiple categories of cross-validation and the most common one is k-fold cross-validation. In the K-fold cross-validation, the original training dataset is split into k different subsets called folds where 1 fold is retained as test set, and the other k-1 folds are used for training the model. In our research work K-fold = 10.

i. *Correlation Coefficient:* This refers to any of a broad class of statistical relationships involving dependence. Familiar examples of dependent phenomena include the correlation between the physical statures of parents and offspring. Correlations are useful because they can indicate a predictive relationship that can be exploited in practice.

ii. *Cross-Correlation* (Disambiguation): This is a measure of the similarity of two series as a function in relation to the other. The true value of interest can be denoted as θ and the value estimated using some algorithm denoted as $\hat{\theta}$.

Correlation is how much θ and $\hat{\theta}$ is related. It gives values between −1 and 1. Where 0 is no relation, 1 is very strong linear relation and −1 is an inverse linear relation (i.e. bigger values of θ indicate smaller values of $\hat{\theta}$ or vice versa).

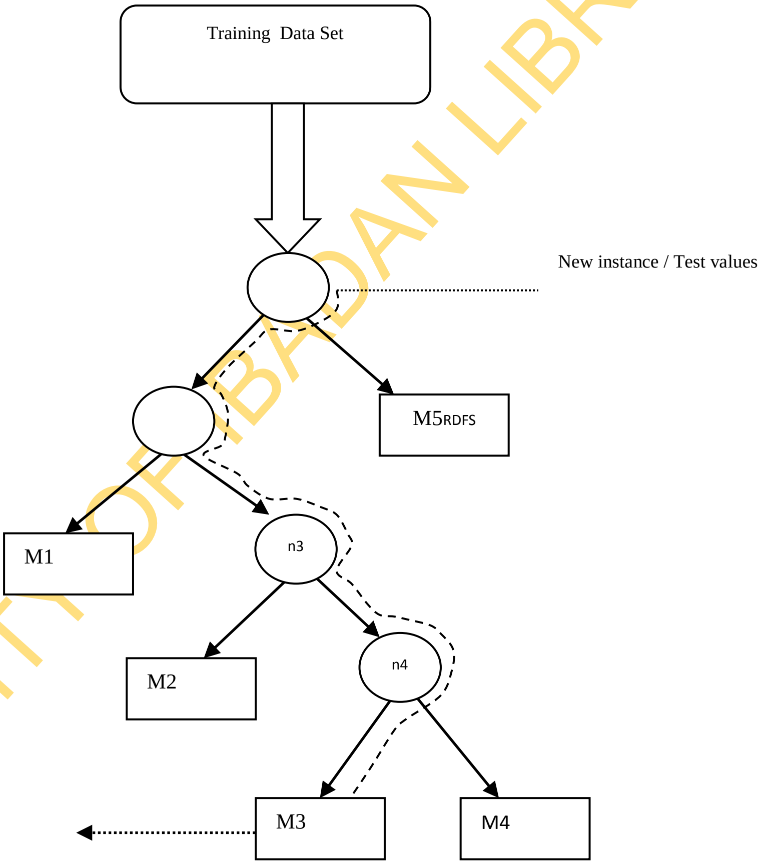


Figure 2.16: A M5P Model Tree

2.9.5 Predictive Errors Analysis

The weights are calculated from the training data with the following errors:

i. *Mean Absolute Error:* Mean Absolute Error (MAE) measures the average magnitude of the errors in a set of forecasts without considering direction. It measures accuracy for continuous variables. The MAE is the average of the absolute error, where is the prediction and the true values. The MAE is a linear score which means that all the individual differences are weighted equally in the average.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|\hat{\theta}_i - \theta_i\right| \qquad 2.33$$

where $\hat{\theta}$ is a mean value of θ

ii. *Root Mean Squared Error (RMSE):* The RMSE is a quadratic scoring rule which measures the average magnitude of the error. The difference between forecast and corresponding observed values is each squared and then averaged over the sample. Finally, the square root of the average is taken. Since the errors are squared before averaged, the RMSE gives a relatively high weight to large errors. This means the RMSE is most useful when large errors are particularly undesirable.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\hat{\theta}_i - \theta_i\right)^2} \qquad 2.34$$

where $\hat{\theta}$ is a mean value of θ

Relative Absolute Error (RAE): This is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. Relative error gives an indication of how good a measurement is relative to the size of that which is being measured. where $\hat{\theta}$ is a mean value of θ

$$RAE = \frac{\sum_{i=1}^{N}\left|\hat{\theta}_i - \theta_i\right|}{\sum_{i=1}^{N}\left|\bar{\theta}_i - \theta_i\right|} \qquad 2.35$$

iii. *Root Relative Squared Error (RRSE):* This predictor is just the average of the actual values. Thus, the relative squared error takes the total squared error and normalises it by dividing it by the total squared error of the simple predictor. By taking the square root of the relative squared error, one reduces the error to the same dimensions as the quantity being predicted.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}\left(\hat{\theta}_i - \theta_i\right)^2}{\sum_{i=1}^{N}\left(\bar{\theta}_i - \theta_i\right)^2}} \qquad 2.36$$

2.10 Related Work

The documents retrieved in the collection model are in response to a query arranged according to the relevance of the query. A single entry in the collection usually consists of details in form of: the title of the document, its URL and a short document called a snippet to calculate the similarity scores. Although such technologies are mostly used but users are still often faced with the daunting task of sifting through multiple pages of results, many of which are irrelevant. Roush (2004) indicated that almost 25% of Web searchers are unable to find useful results in the first set of URLs that are returned. This is due to the keywords based searches which have a tough time distinguishing between words that are spelled the same way but have different meanings. This often results in hits that are completely irrelevant to the query. Also, search engines cannot return hits keywords that mean the same but entered different words in the query. With the conceptual knowledge, search engines based on concepts can effectively handle the above problems where domain specific ontology based semantic search is used.

Ruban and Sam (2015) compared the performance of the traditional query processing methodology with the domain-independent ontology-based query expansion. The Google API was used to search the query and terms were added from the ontology to refine the query. The refined queries were further passed to the search interface. The experiment revealed that the queries that were refined using the domain independent ontology gave more accurate results than that of the query that was given directly to the Search API. So, it was concluded that performance of any search engine will increase by using ontology based query expansion. Tomassen (2009) presented method of indexing documents with ontology vocabulary. In this approach, the index terms derived from the ontology are adapted to the domain terminology with the addition of terms from domain of interest. The approaches are used to clear the query ambiguity in search result.

Matching terms from the document, terms from query in indexing techniques cause a lot of problems because the system did not consider the context (multiple meaning of a word). Therefore, indexers are structured to improve search results with context ontology that provides multiple meanings of a word. Context provides extra information to improve search result's relevance. A context semantic cluster is used to provide indexing of search engine.

Khan and Mustafa (2012) presented semantic search systems that expand search keywords using domain ontology to deal with semantic heterogeneity. Their studyfocused on matching the semantic similarity of individual keywords in a multiple-keywords query, but ignored the semantic relationships that exist among the keywords of the query. The proposed prototype systems matched patterns of keywords to capture the context of keywords in order to validate the system. The system was compared with existing systems for evaluation. The results demonstrate improvement in precision and recall of search.

Zamir and Etzioni (1999) proposed automatic organization of web documents and these problems can be tackled effectively. These approaches are usually supervised and still suffer from alliance on a predefined taxonomy of categories. Different techniques of similarity consisting of associating explicit semantics like Word Sense Disambiguation (WSD) and clustering based on suffix tree techniques are used instead. All these still depend on the way queries are being formulated.

Mahalakshmi (2015) explained the detailed description about text mining and its framework, based on challenge issues in web clustering. Based on similarity, different similarity measures such as string based, corpus based, knowledge based and hybrid based similarity were discussed. Clustering techniques required a precise definition of the closeness between a pair of objects in terms of either the pair wise similarity or distance measures.

Fischer (2013) developed the Semantically Enhanced Domain Specific Natural Language (SE-DSNL) approach which provides experts with the ability to specify how ontological knowledge can be mapped to linguistic information of any known language. The concept provides a flexible and generic meta-model that captured all the relevant information. A prototypical implementation was developed which takes the information of a SEDSNL into consideration in order to use it for parsing natural language text model. This was applied to a given input text and the result is a semantic interpretation of the input text which maps its lexical and syntactic elements to the ontology. The direct integration of semantic and linguistic information further allowed the use of the semantic information at runtime. The validity of the approach showed that it has been evaluated using two case studies that yield certain advantages that could be demonstrated by treating elaborate linguistic phenomena. This encouraged ontology to be mapped with natural language.

Different approaches to search results clustering have been presented in the thesis. Zamir and Etzioni, (1998), Stanislaw and Dawid, (2005), Chim and Deng (2008) and Chung *et al*. (2008) utilised suffix tree model based approach. A great advantage of STC is that phrases are used to provide concise, meaningful descriptions of groups and offer more semantic representations of the text present in the document**.** Unfortunately, due to the vector space model, it does not support incremental processing and is time consuming when applied to large numbers of snippets.

Zhu and Heinz (2008) improved the relevance of web search by recommending to the users personalised results with new web search system that is, Recommender Intelligent Browser (RIB). The RIB combined web snippet categorisation, clustering, and personalisation to find similarity. The problem with this system is semantic heterogeneity which was not solved. The dimensionality of feature would be high because the number of pages retrieved is not limited to a particular web page.

Using Snippets provides useful information regarding the local context of the query term; however, downloading large size documents can be avoided by using snippets. But the main drawback of using snippets is that only those snippets for the top-ranking results for a query can be processed efficiently. Finding similarity score with query and the document returned from the web has a lot of shortcomings. The use of web directories such as Open Directory Project (ODP) provides categorisation for the classification of web pages. Moreso, search results are organised by categories based on the query. The shortcomings are: it is static and needs to be updated manually to cover new pages. Secondly, it is unable to cover large portions of the web and lastly, web pages are classified based on common categories. This latter feature of web directories makes it difficult to distinguish between instances of the same kind when calculating the similarity score. Finding similarity in related domains cannot be based on single information sources as in some of the related works such as Nirgude *et al*. (2013) , Varelas *et al.* (2005), Nguyen and Al-Mubaid (2006), Bollegala *et al*. (2007) etc.

Nirgude *et al*. (2013) proposed Page Count and Snippets Method (PCSM) to estimate semantic similarity between any two words or entities based on page counts and text snippets retrieved from a web search engine. It uses five page count based concurrence measures and integrates with lexical patterns extracted from text snippets. A lexical pattern extraction algorithm was used to identify the semantic relations that exist between any query word pair. Similarity score of both methods are integrated by using Support Vector Machine (SVM) to get optimal results. The method performance is measured by using Pearson correlation value. The correlation value is 0.8960% which is higher than existing methods. This was based on lexical matching and the problem of semantic between related terms still remains unsolved.

Boubekeur and Azzoug (2013) presented automatic concept-based document indexing. It was based on two folds; one of which is the introduction of concept identification based on a domain word sense disambiguation framework that relies on the joint use of WordNet and WordNet domains. The other defines semantic weighting scheme that relies on concept centrality and on its latent importance in the document. It shows that concept-based indexing approach is more effective than the classical keyword-based indexing approach. But in the approach, WordNet domain is hard to classify in any particular domain and senses of synsets can appear in different contexts. The retrieval score for documents does not take into account semantic concepts weights. However, the retrieval system is not structured.

Soni *et al.* (2013) recommended query construction for information retrieval on ontologies, dynamic semantic network and a lexical chain which was formed by a catalogue for the provision of retrieved documents. Semantic relatedness metrics were used to achieve optimal evaluation of the metrics. The lexical analysis was used for extracting semantic association which led to co-occurrence of word in the document collection. This does not solve the problem of semantic mismatch and ambiguity.

Varelas *et al.* (2005) performed experiment on several semantic similarity methods by computing the conceptual similarity between natural language terms using WordNet. Semantic Similarity Retrieval Model (SSRM) was suggested and incorporated in conceptual similarity of retrieval mechanism. SSRM worked in conjunction with taxonomic ontology which is an application specific ontology. However, each term was represented by its tree hierarchy and is stored in XML repository but XML files are unstructured and obsolete for representation of concepts.

Nguyen and Al-Mubaid (2006) proposed a measure that used a new feature of CommonSpecificity (CSpec). This measure was derived from the information content (IC) of concept and IC of corpus. Semantic relatedness of concepts calculation was based on the information sources used which were based on WordNet and the measure adopted was a distance measure not a similarity measure and did not involve lexical resources like web collection.

Bollegala *et al* (2007) proposed semantic similarity measures that used the information available on the web to measure the similarity between words or entities. The method exploited page counts and text snippets returned by a web search engine. Moreover, the semantic similarity measure significantly improved the accuracy in a community mining task and disambiguation. This approach was only limited to information source on web collection of documents. Moreover, a clustering technique was not applied to the web documents. Therefore, increase in the dimensionality of the features retrieved and the similarity of the terms was based on only lexical matching not semantic similarity.

Islam and Inkpen (2008) presented a method called Semantic Text Similarity (STS). This method determined the similarity of two texts from combination of semantic and syntactic information. It considered two functions using string similarity, semantic word similarity and an optional function common-word order similarity. STS method achieved a very good Pearson correlation coefficient but the string similarity is lexical similarities which cannot actually solve the problem of ambiguity and mismatch.

Sabai (2013) suggested the use of Wikipedia category tree and spread activation strategy to compute semantic similarity. The Wikipedia was used as ontology to reduce the effort of experts required to build ontology. Spread activation strategy has produced excellent results for semantic related systems such as word sense disambiguation. The semantic similarity computes using ontologies and describing its documents. The system search is limited to a certain category and the information ranks semantic similarity according to the categories. However, it is not generic and also the spread activation is obsolete for disambiguation.

Prathvi and Ravishankar (2013) presented a method that finds similarity between two words which returned values $0 \leq x \leq 1$. The semantic similarity method makes use of page counts and text snippets retrieved by search engine (Google). Techniques such as pattern extraction from the snippets and pattern clustering were used to determine the similarity scores which help in finding various relationships between words. Traditional Jaccard, Overlap, Dice and Cosine Similarity measures were adopted using page counts. The page counts-based co-occurrence measures and pattern clusters are integrated using support vector machines to define semantic score for a word pair. Semantic Similarity scores found depended on the downloaded web pages which considered the global co-occurrences of two words on the web. However, the feature-based similarity measure was used and this does not signify the semantic similarity. By combining page counts-based co-occurrence measures and lexical pattern, a cluster is learned using support vector machine for optimal semantic similarity between two words. The synonymous and polysemous are not considered going by the lexical pattern extracted from the web and it was based on one information source.

Frederik and Roos (2014) presented the integration of word-sense disambiguation techniques into lexical similarity measures. The specific terms are weighted according to their origin within their respective ontologies. The document similarities between the concept document and sense documents were used to disambiguate the concept meanings. The weighting terms were observed according to the ontology origin which led to the highest performance. The method was only limited to an instance which can change if virtual documents change and it can be faced with disturbed concept names or descriptions.

Nitish *et al.* (2012) presented an approach that used two information sources to calculate the degree of similarity between two sentences. The approach combined corpus-based semantic relatedness measure over the sentence with knowledge-based semantic similarity. However, an Explicit Semantic Analysis (ESA) was used as the corpus-based measure with traditional vector space model as metric in the corpus while the knowledge based semantic similarity used Lin's measure. The similarity scores were fed as features to machine learning models to obtain a single score given the degree of similarity of the sentence. The models used were linear regression and Bagging models. Although the methods show significant improvement in calculating the semantic similarity *tf-idf* was used as a metric and the corpus-base have an issue of co-occurrence. Even the semantic similarity of knowledge only used information contained in the corpus without considering the taxonomy of each word used.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

The Conceptual Knowledge Model for improving Term Similarity in retrieval of web documents consists of three (3) phases and integration of the data from the phases into similarity correlation coefficient for determining the level of similarity of terms in retrieval of web documents. The phase one describes the extraction of two related domain ontologies for querying the search engines. This describes if two related terms *A* and *B* are used as query, would the similarity values be the same? The second phase describes how these queries are searched in the document collections information source. The third phase improves the second phase by using additional knowledge information source for searching the queries. Finally, integration of the data from the two sources is combined to determine the similarity of terms.

3.1.1 Description of the Ontology Extraction

Ontology describes the conceptualisation of a domain in terms of concepts, subconcepts and their relations. These are structured in form of tree like (IS-A hierarchy) on subsumption relationships between concepts. The two ontologies extraction are done from SWEET (Semantic Web for Earth and Environmental Terminology). (https://github.com/DataONEorg/onto-dataonejava/blob/master/SWEETontologies/humanResearch.owl) and (https://github.com/DataONEorg/onto-dataonejava/blob/master/SWEETontologies/reprSciMethodology.owl) as depicted in Figure 3.1

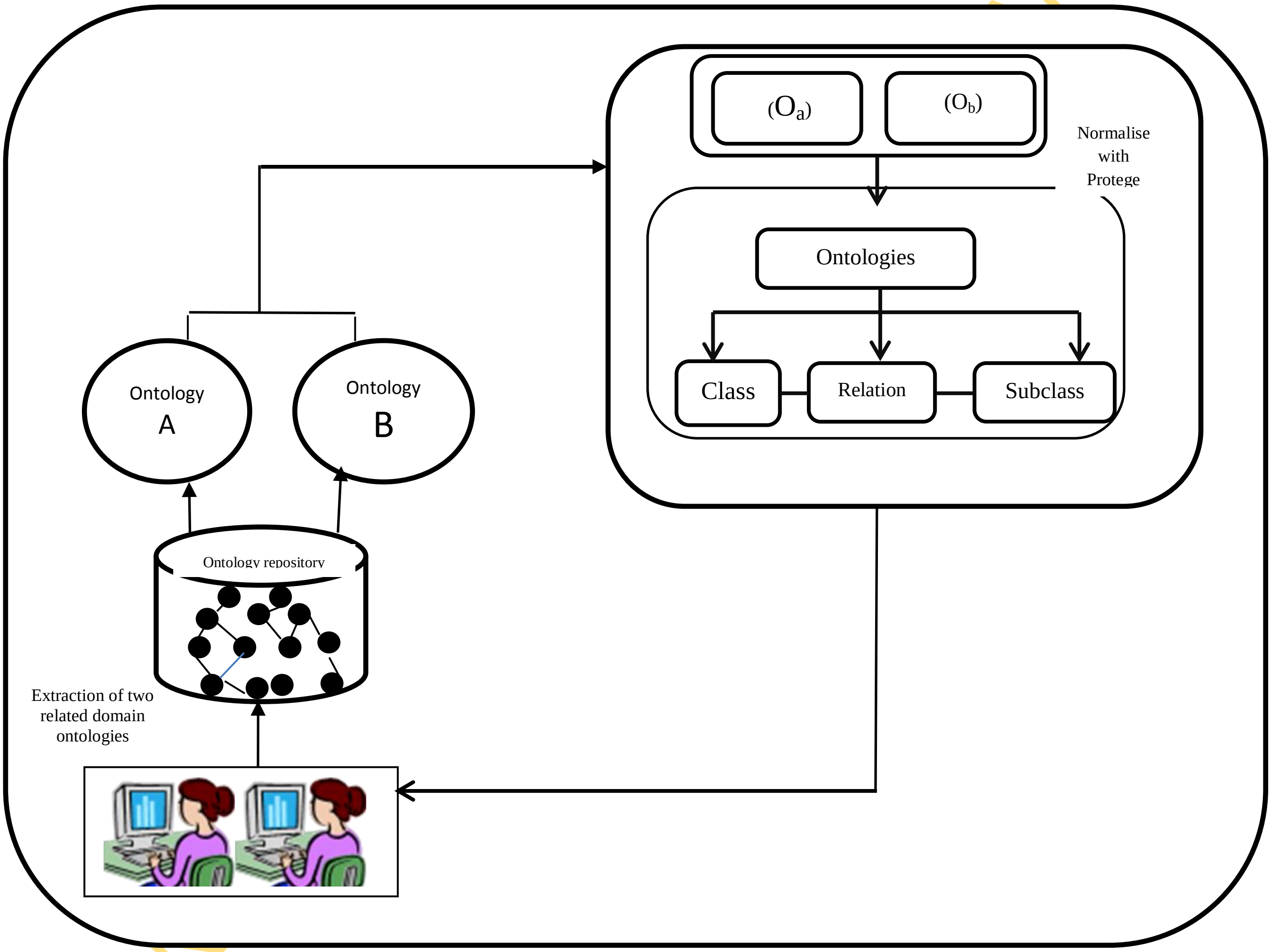


Figure 3.1: Extration of Related Ontologies

The Scope and Assumption of the Work

i. The two ontologies extracted from Internet expressed the relations or overlapping knowledge in a common domain.

ii. The ontologies are represented in a Web Ontology Language (OWL) and normalised with Protégé 4.2.

iii. The two ontologies are expressed in many-to-many relationships. (m:n).

The two (2) related ontologies extracted are in form of XML as depicted in Appendix III and IV. Figure 3.2 shows the graphical representation example of the basic constructs and mechanism of Web Ontology Language (OWL) and normalisation of the ontology with Protégé 4.2 in OWL to make it readable for the user. It describes the set of collection of entities (classes) as an instance derived from concepts. The classes consist of concepts and subconcepts where the property indicates the relationship of each concept. A focus on a particular concept shows how it relates to other concepts with its property.

These extractions are conducted on SciMethodology and humanResearch domains consisting of 17 and 26 concepts in OWL respectively. The seventeen (17) and twenty-six (26) concepts are combined in many-to-many relationships resulting into four hundred and forty two (442) concepts. The information for the two ontologies represents the total number of concepts in form of $c_a[m_a]:c_b[n_b]$. One concept from ontology *A* combined with several concepts in another ontology *B*, so the integration and combination of data are drawn. This relates how the data relates to the real world objects. This simplifies how data from two different information sources (Multiple Documents Sources and Knowledge source) are used for the conceptual knowledge model for the term similarity.

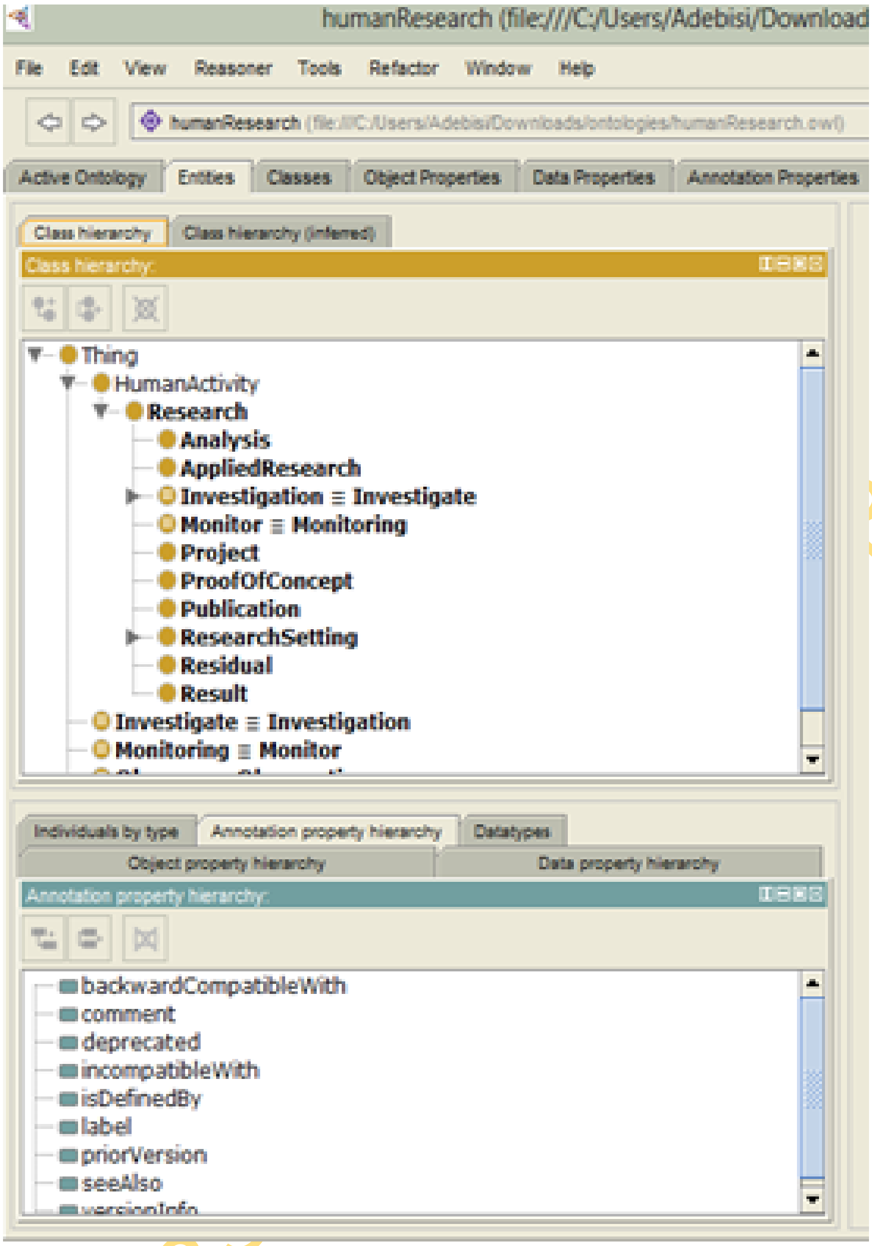


Figure 3.2: Screenshot of the Ontology Concepts Hierarchy

3.1.2 Model User' Query

To represent the semantically similar terms, user query is not sufficient for semantic information retrieval task. User' query is not consistent with vocabulary used in document collections (indexer); therefore, the vocabulary of the queries is controlled so that it agrees with the term used to describe a particular context. Therefore, each node of the ontologies extracted corresponds to concepts, subconcepts and their relations. This formed parent-child relationship with one another in form of hierarchical structure (IS-A relationship).

Each node of the ontology $n_i \in N$ corresponds to a unique concept $c_i \in C$ such that, each $n_i \equiv c_i$. Thus, the relationship $c_p \rightarrow c_c$ describes IS-A relationship (r) between a parent-child concept.

$$O_a = C_{a1} \xrightarrow{r_1} C_{a2} \xrightarrow{r_2} C_{a3} \xrightarrow{r_3} \cdots C_{an} \qquad 3.1$$

$$O_b = C_{b1} \xrightarrow{r_1} C_{b2} \xrightarrow{r_2} C_{b3} \xrightarrow{r_3} \cdots C_{bn} \qquad 3.2$$

Each concept of the ontology in 3.1 and 3.2 is used for querying the search engine. There exists query (*q*), an element of each ontology in A and B such that

$$q \in C_i \qquad \text{where } C_i \text{ is a set of } C_1, C_2, C_3 \cdots C_n \qquad 3.3$$

The concept that the words represent in the search query is used for the expansion of the query (George and Vicky, 2009). Tomassen (2009) presented method of indexing documents with ontology vocabulary of the index terms derived from the ontology to the domain terminology. This method was adapted with each concept from the domain used as a search term but takes additional multiple variable from domain of reference so that more access and search result are retrieved. For instance,

q_a: ***Experiment*** investigation in Research

q_b: ***Analysis*** in human research activity

The concepts and their relations are used as search terms in the Multiple Document Sources (MDSs) and the Knowledge sources. As the queries vocabulary are controlled, the web resource is also structured to suit the controlled queries. Therefore, clustering technique is adapted to structured web resources.

3.2 Adaptation of Suffix Tree Clustering for Structured Search Result

In the second phase of the Conceptual Knowledge Model, user' interaction with Multiple Document Sources (MDSs) needs to be optimised so that the documents retrieved provide users with relevance ranked list. The document and query are represented in a compatible manner.

A Carrot2 was downloaded from the web (http//project.carrot2.org/download java-api html) to structure the web documents and comes with a jar files and Javadocs. The Carrot2 was used because of its advantages: it avoids overlap of document with multiple topics, snippets tolerance and embedded with WordNet. This produced high quality clusters even when only it has access to the snippets returned by the search engine (Sridevi *et al.,* 2011). A Suffix Tree Clustering (STC) is adapted for clustering the web documents to show pattern that exists in the document returned with each URL in the result. Chung *et al*. (2008) utilised suffix tree model and provided concise, meaningful descriptions of groups, phrases base and offered more semantic representations of the document presented in the cluster**.** This was done for search queries in the two domains as depicted in Figure 3.3. This represents document-concept match of the knowledge repository that gives information about concepts and their relationships with other concepts. This will enable conceptual match between extracted concepts that are relevant to the document in the knowledge repositoryas an alternative to keyword match. This is used to tackle the problems of polysemy and synonym from documents so that mismatch and ambiguous terms are disambiguated.

The documents set are grouped into a subspace according to search terms which makes the search results easier, provides efficient and effective retrieval. Figure 3.4 shows a screenshot of Carrot2 tool benchmark that structured, grouped the documents into clusters and assigned scores to each of the cluster. Bollegala *et al.* (2011) used page counts and snippets as information sources provided by web search engines to generate semantic similarity results but without clustering. To avoid repetition of document set in the clusters, web pages are set to four (4) with 100 documents.

The implementation of the suffix tree was done in Java with the Carrot2 API and used Goggle API as the backend search engine to cluster the web for its simplicity.

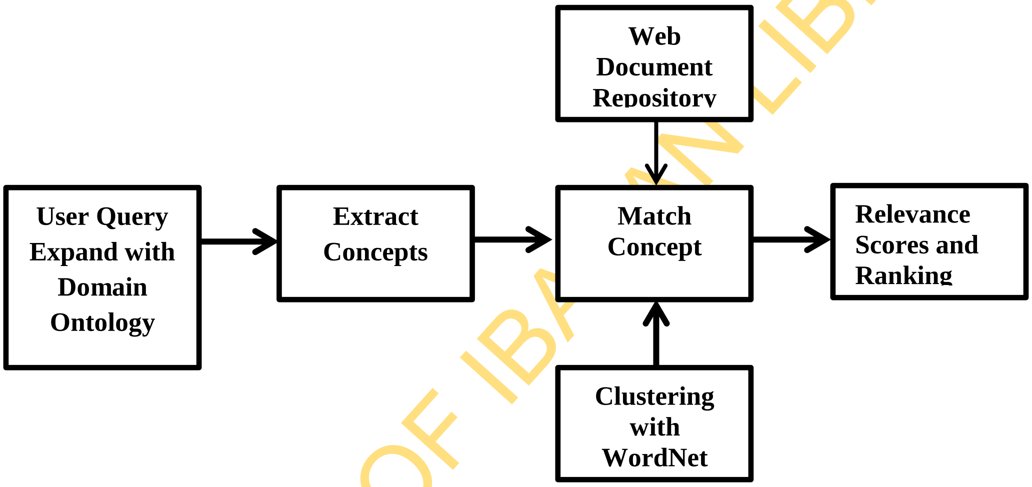
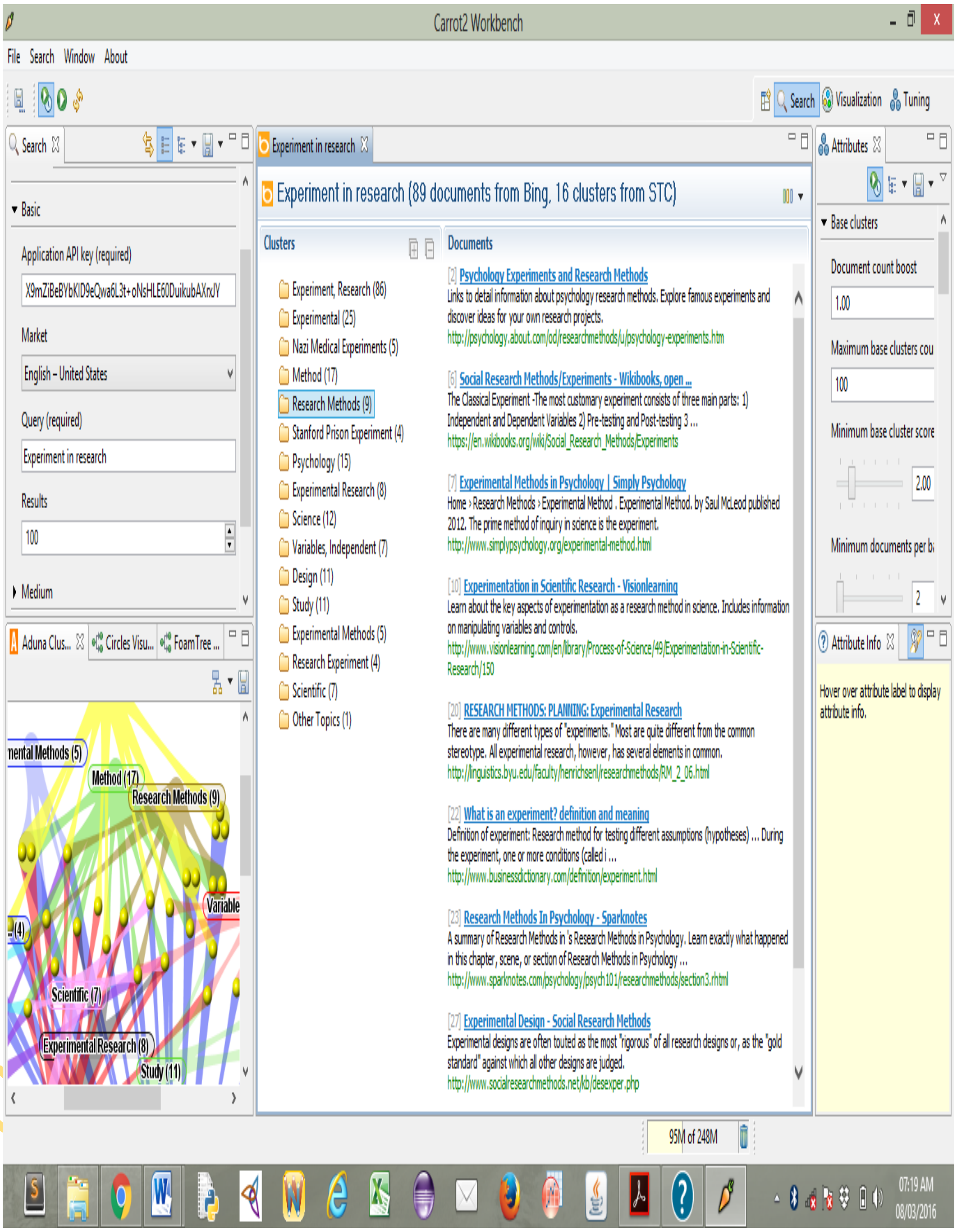


Figure 3.3: Conceptual Match in Multiple Document Sources



Application API key (required)

X9mZiBeBYbKID9eQwa6L3t+oNsHLE60DuikubAXrxJY

Market

English – United States

Query (required)

Experiment in research

Results

100

Document count boost

1.00

Maximum base clusters cou

100

Minimum base cluster score

2.00

Minimum documents per b

2

Experiment in research (89 documents from Bing, 16 clusters from STC)

[2] Psychology Experiments and Research Methods
Links to detail information about psychology research methods. Explore famous experiments and discover ideas for your own research projects.
http://psychology.about.com/od/researchmethods/u/psychology-experiments.htm

[6] Social Research Methods/Experiments - Wikibooks, open ...
The Classical Experiment -The most customary experiment consists of three main parts: 1) Independent and Dependent Variables 2) Pre-testing and Post-testing 3 ...
https://en.wikibooks.org/wiki/Social_Research_Methods/Experiments

[7] Experimental Methods in Psychology | Simply Psychology
Home › Research Methods › Experimental Method . Experimental Method. by Saul McLeod published 2012. The prime method of inquiry in science is the experiment.
http://www.simplypsychology.org/experimental-method.html

[10] Experimentation in Scientific Research - Visionlearning
Learn about the key aspects of experimentation as a research method in science. Includes information on manipulating variables and controls.
http://www.visionlearning.com/en/library/Process-of-Science/49/Experimentation-in-Scientific-Research/150

[20] RESEARCH METHODS: PLANNING: Experimental Research
There are many different types of "experiments." Most are quite different from the common stereotype. All experimental research, however, has several elements in common.
http://linguistics.byu.edu/faculty/henrichsenl/researchmethods/RM_2_06.html

[22] What is an experiment? definition and meaning
Definition of experiment: Research method for testing different assumptions (hypotheses) ... During the experiment, one or more conditions (called i ...
http://www.businessdictionary.com/definition/experiment.html

[23] Research Methods In Psychology - Sparknotes
A summary of Research Methods in 's Research Methods in Psychology. Learn exactly what happened in this chapter, scene, or section of Research Methods in Psychology ...
http://www.sparknotes.com/psychology/psych101/researchmethods/section3.rhtml

[27] Experimental Design - Social Research Methods
Experimental designs are often touted as the most "rigorous" of all research designs or, as the "gold standard" against which all other designs are judged.
http://www.socialresearchmethods.net/kb/desexper.php

Figure 3.4: A Screenshot of Carrot2 Benchmark

The project was created in Eclipse, imported the library into the project and ran the code from the Eclipse; this is depicted in Appendix V. A cluster that relates to the domain ontology is chosen from the set of clusters. 3.4 represents the sequence of selecting documents that matches the domain from the chosen cluster.

Let $D = c_{1,} c_2, c_3 \cdots c_m$ 3.4

where D represents documents (snippets) as a sequence of terms { $c_i \Rightarrow_{,} \; i = 1 \cdots m \}$

Let S be a suffix tree of n snippet and

For each n_m length is a tree that contain a root (R) node

$\sum n_m$ is the entities (concept, subconcept and its relation) as depicted in Figure 3.5

$D(S) =$ Set of entities n in snippet are elements present in cluster S ($C_i \in S$)

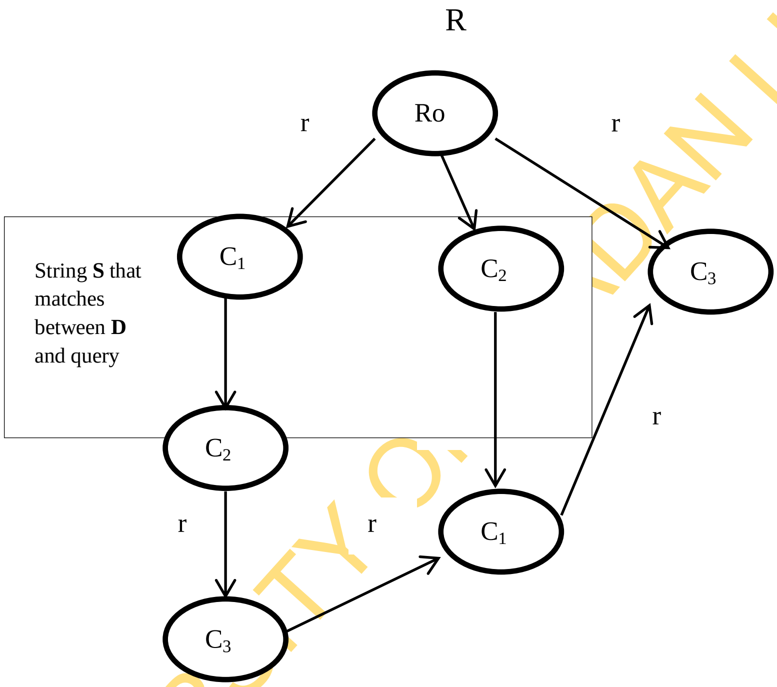
Domain Relevance Score $(DRS) = \sum_{i=1}^{3} \frac{1}{n_i} D(S) C_i$ 3.5

This was expressed further as:

$$\frac{1}{n_1} D(S) C_1 + \frac{1}{n_2} D(S) C_2 + \frac{1}{n_3} D(S) C_3 \qquad 3.6$$

In 3.5, the documents set in cluster D(S) that match the concepts, sub-concepts and their relations are checked as depicted in Figure 3.5. This helps to disambiguate the large document set in each domain; therefore, the documents that relate to the query terms are selected. This was limited to at least three (3) of the entities concepts as shown in 3.6 and Figure 3.6 shows the sequence of the selection of the entities in the documents set.

Two ways are distinguished in detecting concepts and their relation from documents. The domain ontologies are projected on the clustered documents and extraction of all words and multiword concepts (compound terms) from the ontology are identified in the documents retrieved. This enables reuse of web resource even if the query changes. Also, the reverse is followed by projecting the documents onto the ontology. For each word or multiword concept candidate formed by combining adjacent words in text phrases, the ontology is checked using those words just as it is in the query or the stem forms are checked. Java implementation of suffix tree clustering is shown in Figure 3.7 and the underlined candidate terms are related in the ontology.



Where C_1, C_2, C_3 are entities which contains the classes (concept and subconcept) and its relation (r) in the cluster (S)

Figure 3.5: A Suffix Tree Clustering in Multiple Document Sources

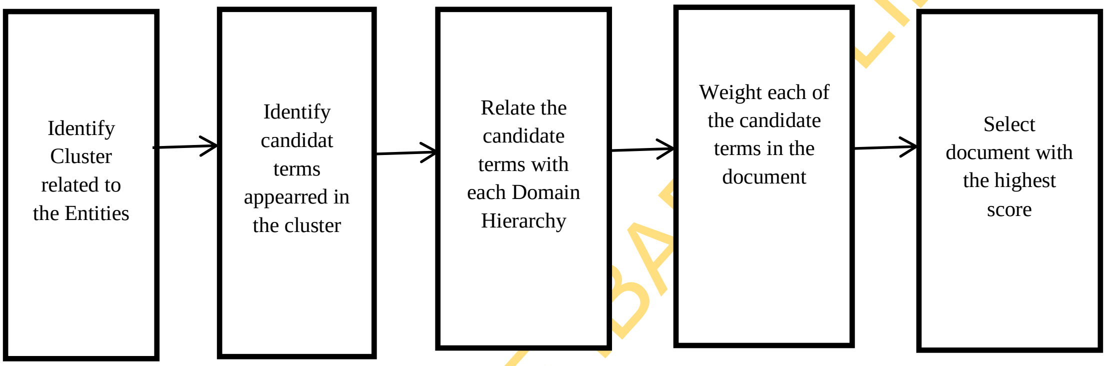
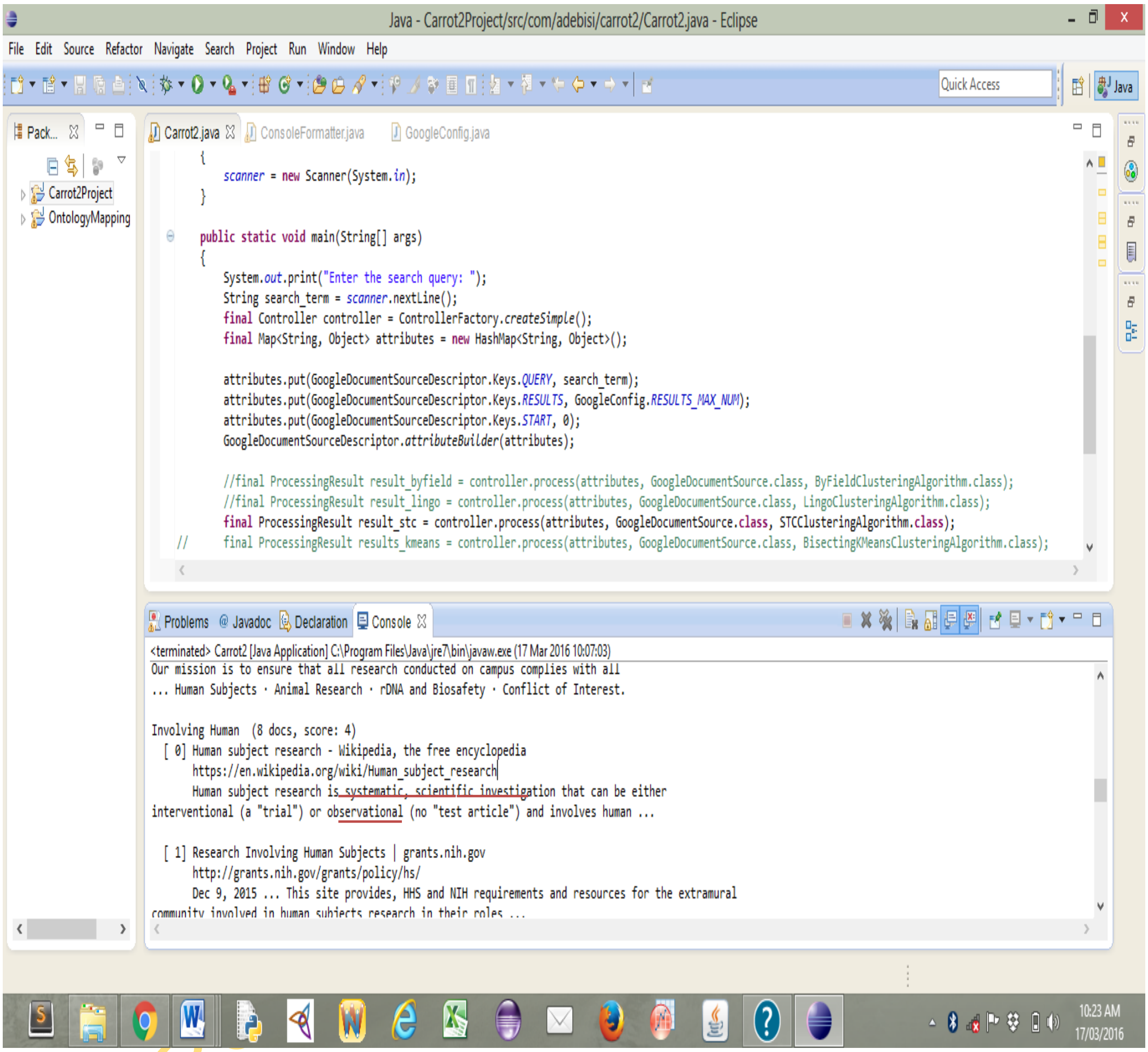

Identify Cluster related to the Entities
Identify candidat terms appearred in the cluster
Relate the candidate terms with each Domain Hierarchy
Weight each of the candidate terms in the document
Select document with the highest score

Figure 3.6: Selection of Cluster in relation to Domain Entity

.



```java
    {
        scanner = new Scanner(System.in);
    }

    public static void main(String[] args)
    {
        System.out.print("Enter the search query: ");
        String search_term = scanner.nextLine();
        final Controller controller = ControllerFactory.createSimple();
        final Map<String, Object> attributes = new HashMap<String, Object>();

        attributes.put(GoogleDocumentSourceDescriptor.Keys.QUERY, search_term);
        attributes.put(GoogleDocumentSourceDescriptor.Keys.RESULTS, GoogleConfig.RESULTS_MAX_NUM);
        attributes.put(GoogleDocumentSourceDescriptor.Keys.START, 0);
        GoogleDocumentSourceDescriptor.attributeBuilder(attributes);

        //final ProcessingResult result_byfield = controller.process(attributes, GoogleDocumentSource.class, ByFieldClusteringAlgorithm.class);
        //final ProcessingResult result_lingo = controller.process(attributes, GoogleDocumentSource.class, LingoClusteringAlgorithm.class);
        final ProcessingResult result_stc = controller.process(attributes, GoogleDocumentSource.class, STCClusteringAlgorithm.class);
//      final ProcessingResult results_kmeans = controller.process(attributes, GoogleDocumentSource.class, BisectingKMeansClusteringAlgorithm.class);
```

```
<terminated> Carrot2 [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (17 Mar 2016 10:07:03)
Our mission is to ensure that all research conducted on campus complies with all
... Human Subjects · Animal Research · rDNA and Biosafety · Conflict of Interest.

Involving Human  (8 docs, score: 4)
  [ 0] Human subject research - Wikipedia, the free encyclopedia
       https://en.wikipedia.org/wiki/Human_subject_research
       Human subject research is systematic, scientific investigation that can be either
interventional (a "trial") or observational (no "test article") and involves human ...

  [ 1] Research Involving Human Subjects | grants.nih.gov
       http://grants.nih.gov/grants/policy/hs/
       Dec 9, 2015 ... This site provides, HHS and NIH requirements and resources for the extramural
community involved in human subjects research in their roles ...
```

Figure 3.7: Java Implementation of STC and Selected Entities

The documents chosen (set of snippets) undergo preprocessing to remove unwanted words for further processing.

3.2.1 Dimensionality Reduction by Document Indexing

The documents retrieved from the two (2) domains are mapped and preprocessing is done using ScikitLearn (SKlearn) feature extraction module. This module extracts feature terms in a format supported by machine learning algorithm. The implementation was done in Python (see Appendix VI). The Pythron is used due to its level of robustness, ease of use; although the interface is Python but C-libraries are leveraged for performance. Python also has packages such as NumPy (use for matix computation), SciPy and finally and a learning curve almost like reading English.

Natural Language ToolKit (NLTK) is imported into the SKlearn to perform automatic document indexing as depicted in Figure 3.8. The document retrieved from two domains is mapped and preprocessing is performed. This involves the extraction of each word from the documents and every word is changed to its lowercase. The Treebank WordTokeniser in NLTK is employed to tokenise each word and split the documents into individual terms or sequences of words such as:

Input: word_tokenise (this research work is done by adebisi)

Output: [‘this’, ‘research’, ‘work’, ‘do’, ‘adebisi’]

Each word is tagged as it appeared in the tokenisation to its corresponding parts of speech which are based on the context of the document as in the example:

tagged_tokens=pos_tag (token)

pos_tag (word_tokenise (“ this’, ‘research’, ‘work’, ‘do’, ‘adebisi’ ”))

[(‘this’, ‘DT’), (‘research’, ‘NN’), ‘work’, ‘JJ’), (‘do’, ‘VB’), (‘adebisi’,’ NNP’)]

(where DT=Determiner, NN=Noun, JJ=Adjective, VB=Verb and NNP = Proper Noun).

The tagged words removed unwanted words that are stopwords such as: the, it, in, for etc., these words do not describe document’s content.

Suffixes words such as “es, ed, ing, ion” are removed and leave the word to a single term. Theystem words and reduce the number of unique vocabulary words or terms that need to be trailed. This speeds up computational operations.

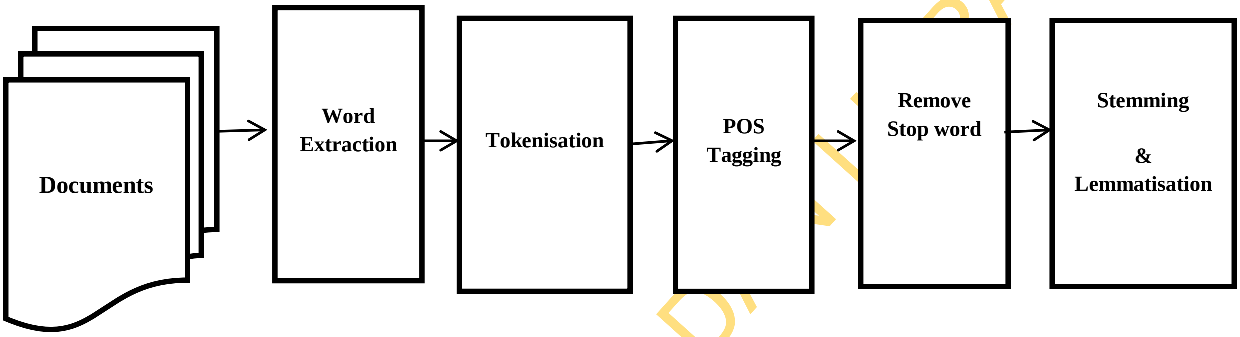


Figure 3.8: Document Indexing Process

Consequently, a WordNet package was imported to lemmatise words that are not in stem form but more into synonym replacement such as “feet and foot”. For instance:

Words (w) = [do, doing, did , done]

from nltk.stem import

***Input:** stemer.stem(w) for word w words*

***Output:** [' do',' do',' do',' do']*

In addition, the process reduced the total number of terms or words in the documents and hence reduced the size and complexity of the documents.

3.2.2 Dimensionality Reduction by Concept Weighting Model

The remaining words in the document underwent a count where countVectoriser is used to count the occurrence of word in each preprocessed document and returned as a sparse matrix. The following assumptions are made:

1. The normalisations of feature are based on snippets not on document.
2. The more concepts appeared in the snippets, the more characteristic concept to the snippets, four (4) for each query searched term are used.
3. The distance between concepts in an ontological set (hierarchies) is considered as one (1) for the concept weighting model; therefore one concept in the ontology is related to another concept in the hierarchy.
4. The probabilities of each concept in relation to all other concepts in the snippets are found.

Considering these assumptions, a concept weighting model is developed to normalise the sparse matrix of feature and determine the similarity.

$$wt_c = cf_{i.j} * \left(\left({df_c(j)} \big/ {N_c} \right) \right) \qquad 3.7$$

where,

wt$_c$= Concept Weighting

$cf_{i,j}$ = frequency of concept *i* in snippet *j*

$df_c(j)$ = snippets where concept *i* appear

N$_c$= number of all the concepts in the snippets set

The Concept Weighting CW_{aij} and CW_{bij} associated with *concepts* i^{th} are calculated. This finds the frequency of the concepts in the snippets that are assigned to the two domains. The model is used as the vectoriser to normalise and minimise the effects of common terms that have semantic meaning by interaction between the concepts and their relations that appear in the two domains. However, it reduced the dimension and assembled the matrix in form of Table 3.1. The n-dimensional matrix corresponds to a distinct term and each term has an associated weight.

The term and concept similarity measure are found on the vectoriser of mapped snippets in *A* and *B* as depicted in Figure 3.9.

How much does concept used in query A has to do with concept used in query B?

The similarities are calculated for *tf-idf* and Concept Weighting (CW) for Cosine, Euclidean and Gaussian Radial Basis Function (RBT). This finds the similarity between snippets for query *Ai* and *Bi*. These are usually values between 0 and 1, where 0 signifies low similarity and 1 signifies extremely high similarity. The algorithm below describes the process in the SciKitLearn.

Table 3.1: Concept by Snippet Matrix

$^{c}/_{d}$	D_1	D_2	• • •	D_3
C_1	$c_1 d_1$	$c_1 d_2$	• • •	$c_1 d_n$
C_2	$c_2 d_1$	$c_2 d_2$	• • •	$c_2 d_n$
•	▬	▬	• • •	▬
•	▬	▬		▬
•	▬	▬		▬
C_n	$c_n d_1$	$c_n d_2$	• • •	$c_n d_n$

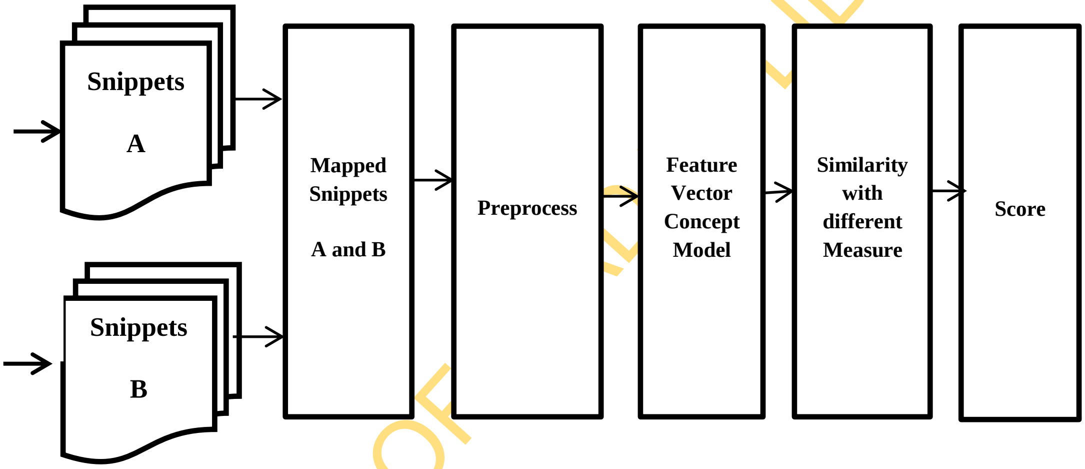

Snippets
A
Snippets
B
Mapped Snippets
A and B
Preprocess
Feature Vector Concept Model
Similarity with different Measure
Score

Figure 3.9: The Concept Similarity in MDSs

The concept similarity algorithm is presented as follows:

1. *Download ScikitLearn Tool*
2. *Install the Python Natural Language ToolKit (NLTK) library.*
3. *Install the WordNet corpora data for the Natuaral Language ToolKit.*
4. *Mapped snippets and each string was passed through NLTK to tokenise the snippets*
5. *A 'list comprehension' construct in NLTK library was set for the English stop-word.*
6. *Stem and lemmatise each of the tokens in the snippet using NLTK 'WordNetLemmatizer'.*
7. *Feed the filtered snippets tokens into the CountVectorizer*
8. *Get Feature_Names were called to get the list of extracted features from the snippets and are fed into the countVectorizer.*
9. *Apply normalisation to a sparse matrix of feature occurrence.*
10. *Calculate the similarity score with different similarity measure*

Figure 3.10 describes the incorporation of phase one into phase two of the Conceptual Knowledge from the extraction of ontologies to the query formulation, to the retrieved documents/snippets by the user.

The domain concept similarity scores generated in the MDSs are not enough to proffer similarity as accurate as possible. This requires additional source (Semantic Network) to adjust the similarity score to semantic level.

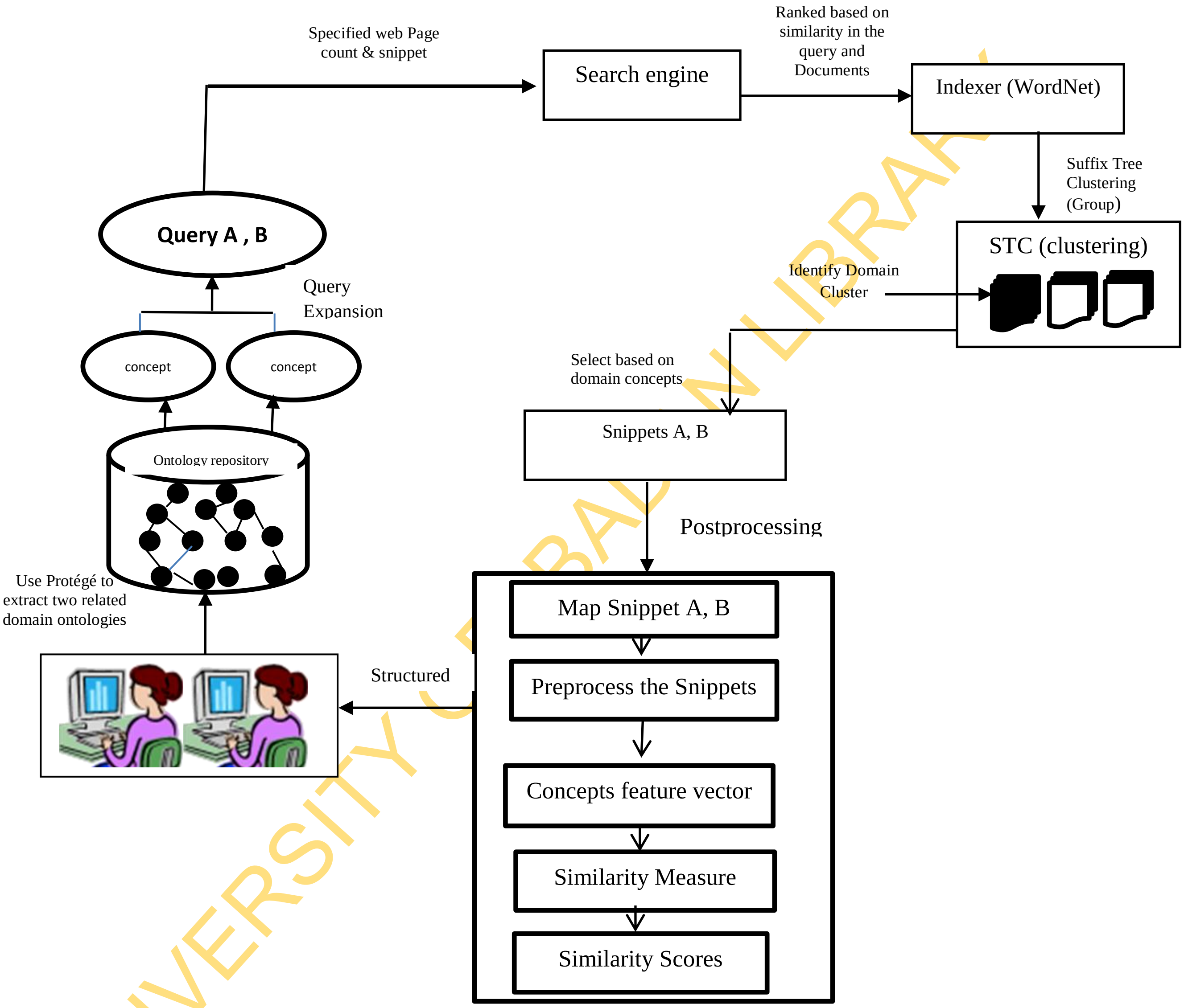


Figure 3.10: Conceptual Knowledge in MDSs

3. 3 Adjust Similarity Value with Semantic Network

The similarities in MDSs failed to deal with terms not covered by synonym dictionaries or are not able to cope with acronyms, abbreviations, buzzwords etc. But conceptual knowledge in semantic network source uses some kind of web intelligence to determine the degree of similarity between text expressions. The same concepts query *A* and *B* used in MDSs are also used in semantic similarity in WordNet lexicon to adjust the term similarity generated in MDSs. The WordNet relational dictionary (WordNetSimilarity) calculates the semantic similarity of different measures in the knowledge source. This WordNetSimilarity accessed information contained in concepts of a query and determined the similarity between query *A* and *B* but this required human intervention. Meng *et al.* (2014) used path and information content that is inversely proportional to length (CA,CB) but does not consider the position of the hierarchies of concepts A and B. Also, the Information Content (IC) similarity in WordNetSimilarity lacks the path from hierarchical structure of concepts.

The following assumptions are taken into consideration to determine semantic similarity of concepts in WordNetSimilarity for query *A* and *B:*

i. Concepts are stemmed.

ii. A noun was used as the part of speech with its hypernym.

iii. The compound word is treated as a single word and its related words are chosen in the WordNet hierarchy.

Figure 3.11 shows a senses for a particular synsets of concepts search. Hierarchy sense that relates to the documents chosen in MDSs is also chosen in WordNetSimilarity.

In 3.5, model is developed that describes the Information Content of LIn' measure with extension of PAth length to derive LIPA as shown:

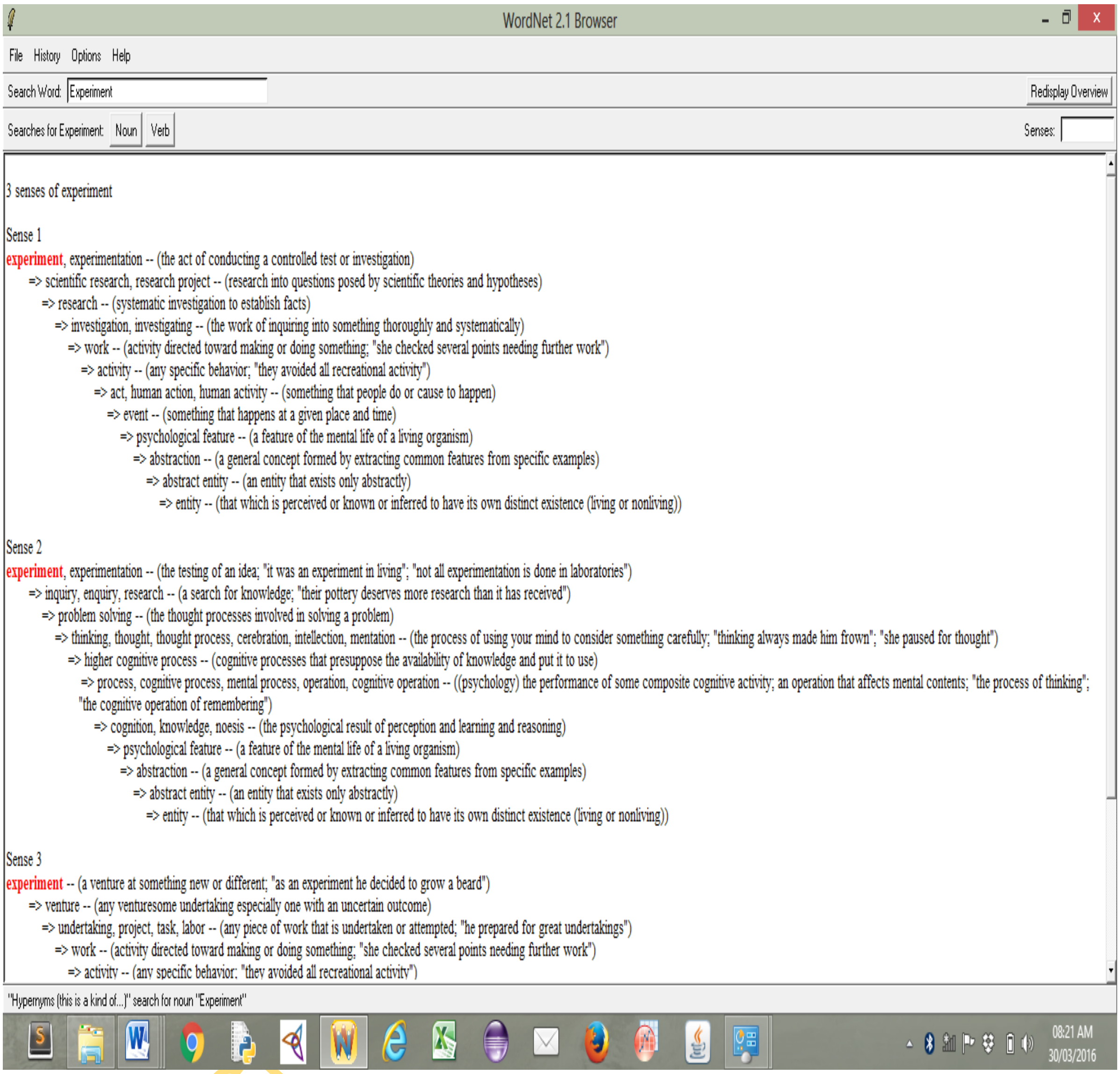
$$sim_{LIPA}(c_a, c_b) = \frac{2 * IC(lcs(c_a, c_b))}{IC(c_a) + IC(c_b)} * (1-k)^l \qquad 3.5$$

where

sim_{LIPA} = Similarity between concept *A* and *B*

lcs = *least common subsumer*

$$\begin{cases} l = 0,\ if\ C_a = C_b \\ otherwise\ l = max\ C_a\ or\ \ C_b \end{cases}$$



```
3 senses of experiment

Sense 1
experiment, experimentation -- (the act of conducting a controlled test or investigation)
    => scientific research, research project -- (research into questions posed by scientific theories and hypotheses)
      => research -- (systematic investigation to establish facts)
        => investigation, investigating -- (the work of inquiring into something thoroughly and systematically)
          => work -- (activity directed toward making or doing something; "she checked several points needing further work")
            => activity -- (any specific behavior; "they avoided all recreational activity")
              => act, human action, human activity -- (something that people do or cause to happen)
                => event -- (something that happens at a given place and time)
                  => psychological feature -- (a feature of the mental life of a living organism)
                    => abstraction -- (a general concept formed by extracting common features from specific examples)
                      => abstract entity -- (an entity that exists only abstractly)
                        => entity -- (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Sense 2
experiment, experimentation -- (the testing of an idea; "it was an experiment in living"; "not all experimentation is done in laboratories")
    => inquiry, enquiry, research -- (a search for knowledge; "their pottery deserves more research than it has received")
      => problem solving -- (the thought processes involved in solving a problem)
        => thinking, thought, thought process, cerebration, intellection, mentation -- (the process of using your mind to consider something carefully; "thinking always made him frown"; "she paused for thought")
          => higher cognitive process -- (cognitive processes that presuppose the availability of knowledge and put it to use)
            => process, cognitive process, mental process, operation, cognitive operation -- ((psychology) the performance of some composite cognitive activity; an operation that affects mental contents; "the process of thinking"; "the cognitive operation of remembering")
              => cognition, knowledge, noesis -- (the psychological result of perception and learning and reasoning)
                => psychological feature -- (a feature of the mental life of a living organism)
                  => abstraction -- (a general concept formed by extracting common features from specific examples)
                    => abstract entity -- (an entity that exists only abstractly)
                      => entity -- (that which is perceived or known or inferred to have its own distinct existence (living or nonliving))

Sense 3
experiment -- (a venture at something new or different; "as an experiment he decided to grow a beard")
    => venture -- (any venturesome undertaking especially one with an uncertain outcome)
      => undertaking, project, task, labor -- (any piece of work that is undertaken or attempted; "he prepared for great undertakings")
        => work -- (activity directed toward making or doing something; "she checked several points needing further work")
          => activity -- (any specific behavior; "they avoided all recreational activity")

"Hypernyms (this is a kind of...)" search for noun "Experiment"
```

Figure 3.11: A WordNet Hypernym Hierarchy with Senses

K is a parameter and $0 \leq K \leq 1$, which can be adapted manually to make the metric to get the best performance.

Threshold (K) value of 0.5 is assumed for two (2) concepts words to be similar or related in WordNet and

l to be the maximum length of taxonomy between two (2) concepts in WordNet.

A web interface is developed in Java to access information contained in the two domain concepts to determine the semantic similarity of query *A* and *B* in LIPA. Four other exixting WordNetSimilarity are calculated such as: LIN, JCN, WUP and PATH as depicted in Figure 3.12.

The process of WordNetSimilarity is as follows:

i. Java modules "*JDK 1.8*" is used

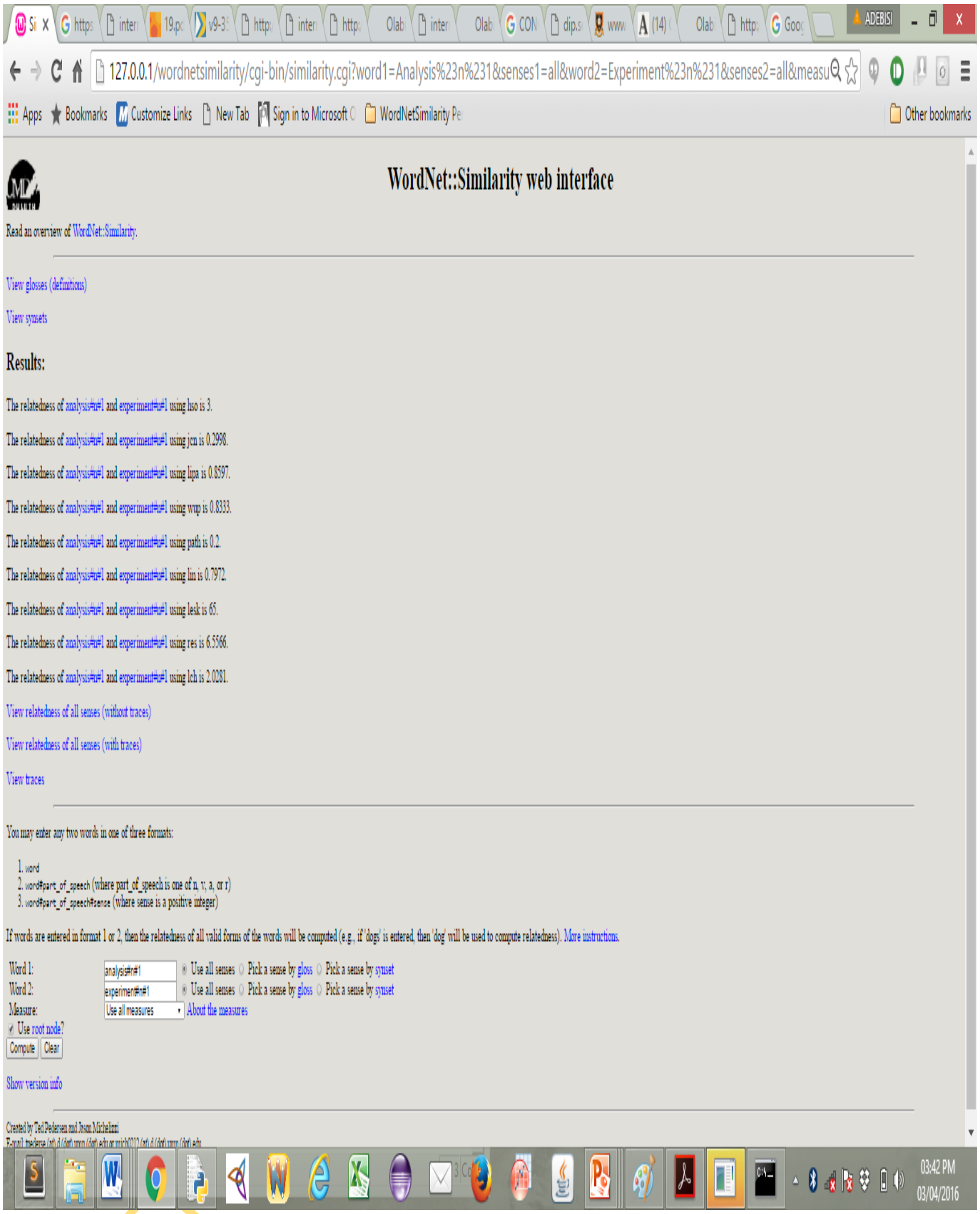
ii. It required the *WS4J* Package

iii. Installed Text Similarity

iv. *WordNet QueryData* and WordNet Similarity are extracted

v. Extend the LIn'measure (Information Content) with PAth length to form LIPA

Figure 3.13 describes how ontologies extracted are used in the knowledge sources in the third phase of the Conceptual Knowledge Model.



127.0.0.1/wordnetsimilarity/cgi-bin/similarity.cgi?word1=Analysis%23n%231&senses1=all&word2=Experiment%23n%231&senses2=all&measu

WordNet::Similarity web interface

Read an overview of WordNet::Similarity.

View glosses (definitions)

View synsets

Results:

The relatedness of analysis#n#1 and experiment#n#1 using hso is 3.

The relatedness of analysis#n#1 and experiment#n#1 using jcn is 0.2998.

The relatedness of analysis#n#1 and experiment#n#1 using lipa is 0.8597.

The relatedness of analysis#n#1 and experiment#n#1 using wup is 0.8333.

The relatedness of analysis#n#1 and experiment#n#1 using path is 0.2.

The relatedness of analysis#n#1 and experiment#n#1 using lin is 0.7972.

The relatedness of analysis#n#1 and experiment#n#1 using lesk is 65.

The relatedness of analysis#n#1 and experiment#n#1 using res is 6.5566.

The relatedness of analysis#n#1 and experiment#n#1 using lch is 2.0281.

View relatedness of all senses (without traces)

View relatedness of all senses (with traces)

View traces

You may enter any two words in one of three formats:

1. word
2. word#part_of_speech (where part_of_speech is one of n, v, a, or r)
3. word#part_of_speech#sense (where sense is a positive integer)

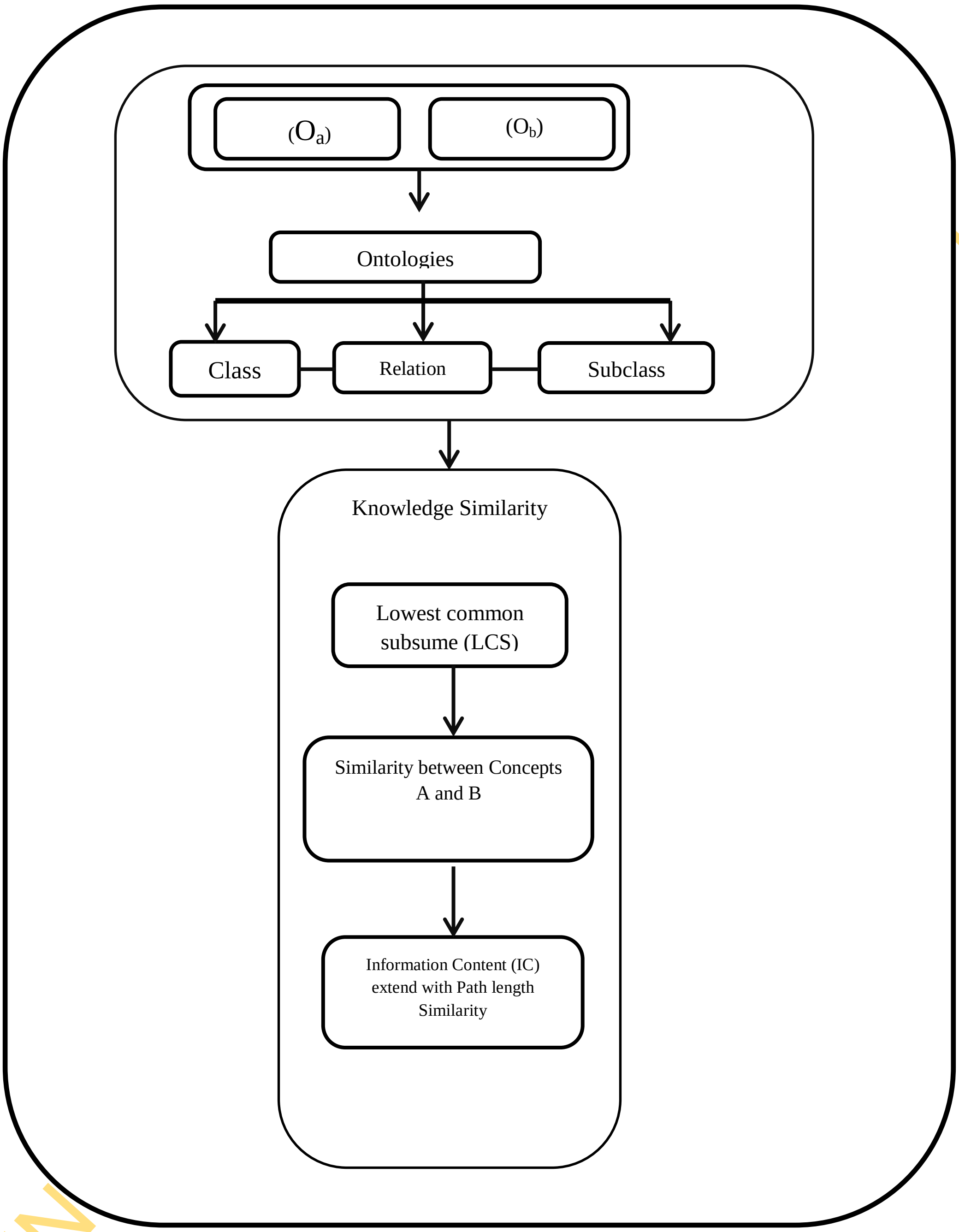
If words are entered in format 1 or 2, then the relatedness of all valid forms of the words will be computed (e.g., if 'dogs' is entered, then 'dog' will be used to compute relatedness). More instructions.

Word 1: analysis#n#1 ● Use all senses ○ Pick a sense by gloss ○ Pick a sense by synset
Word 2: experiment#n#1 ● Use all senses ○ Pick a sense by gloss ○ Pick a sense by synset
Measure: Use all measures About the measures
☑ Use root node?

Show version info

Created by Ted Pedersen and Jason Michelizzi
E-mail: tpederse (at) d (dot) umn (dot) edu or mich0212 (at) d (dot) umn (dot) edu

Figure 3.12: A Web Interface for WordNetSimilarity



(O_a)
(O_b)
Ontologies
Class
Relation
Subclass
Knowledge Similarity
Lowest common subsume (LCS)
Similarity between Concepts A and B
Information Content (IC) extend with Path length Similarity

Figure 3.13: Concept Similarity in Semantic Network

3.4 Integration of the MDSs and Knowledge Sources Data

The resources data generated from conceptual knowledge in MDSs and Semantic Network are combined using machine learning techniques. The techniques such as linear regression, Support Vector Machines for regression (SVMreg), neural network, M5P Model Tree and Bagging are used to conduct predictive analysis and determine the similarity correlation coefficient. The concept similarity amplified by a tuning parameter in WordNetSimilarity adjusts the similarity values generated. This was used to determine semantic similarity or relatedness of terms.

The Conceptual knowledge Model is presented in the algorithm below with representation in Figure 3.14

```
1.  get XML data of the two ontology a and b
2.  Use protégé to normalise the ontology to OWL
3.  Ontology a ,b $\in$ concept hierarchy ($C_1, C_2, C_3 \cdots C_n$)
4.  for $C_1, C_2, C_3 \cdots C_n$ in ontology a, b
5.      search each $C_1, C_2, C_3 \cdots C_n$ as search term
6.      cluster the web snippets
7.      retrieve textual data for the two related concepts a and b
        Select snippet relate to entities
8.  Merge snippet from a and b into a single snippets collection=
    SNIPPET_COLLECTION
    for each snippet in in SNIPPET_COLLECTION
        split snippet into tokens
        remove stop_words from tokens
        Merge tokens into TOKENS_COLLECTION
            Process TOKENS_COLLECTION
9.  Process into matrix with dimension (concept_count, unique token count)
    matrix_frequency
    Set total concept occurrences = count (token_collection) = term_occur
        for each concept in a,b
            concept in each a, b= $C_i$
        If term/token occurs in ontology
            Set concept frequency = cf
```

```
                    Compute concept weight =cf *(df(j)/no of concept in the snippet)
                     Set concept weight = WEIGHT
                        Set matrix value for matrix_frequency(Index(ontology),
Index(token)) =fq
10.  Create a matrix with dimensions (ontology_count, unique token count)
matrix_similarity
        for each ontology in step 1:
            compute token similarity from tokens_collection Sim
            Set matrix value for matrix_similarity (Index(ontology), Index(token)) = Sim
11.  Search Ontology a and b in WordNetSimilarity
     Let constant K = 0.5 for minimum Similarity of a and b
        for each $t_i$ in Ontology_a
        for each $t_i$ in Ontology_b:
            loop all $t_i$ in ontology_b
                 get Hypernyms of t1 from WordNet = HYPERNYMS_a
                         get Hypernyms of t2 from WordNet = HYPERNYMS_b
                   select Best-fit hypernym from HYPERNYMS _a = HYP _BEST_FIT a
                   select Best-fit hypernym from HYPERNYMS _b = HYP_BEST_FIT_b
        LCS_t1_t2 = Least Common Subsumer of ($c_a$, $c_b$) of (HYP_BEST_FIT_a,
        HYP_BEST_FIT_b)
                 L1 = Longest path length from $c_a$ to LCS_ $c_a$_$c_b$ in WordNet
                 L2 = Longest path length from $c_b$ to LCS_ $c_a$_$c_b$ in WordNet
                         Length = IF t1 == t2 THEN 0 ELSE L1 > L2 ? L1 : L2
                         IC = IC(LCS_t1_t2)
12. Compute SIM_$c_a$_$c_{b(LIPA)}$ = ((2 * IC) / (IC(t1) + IC(t2))) * (I - K) ^ Length
13. Evaluate step step 9 and 11
```

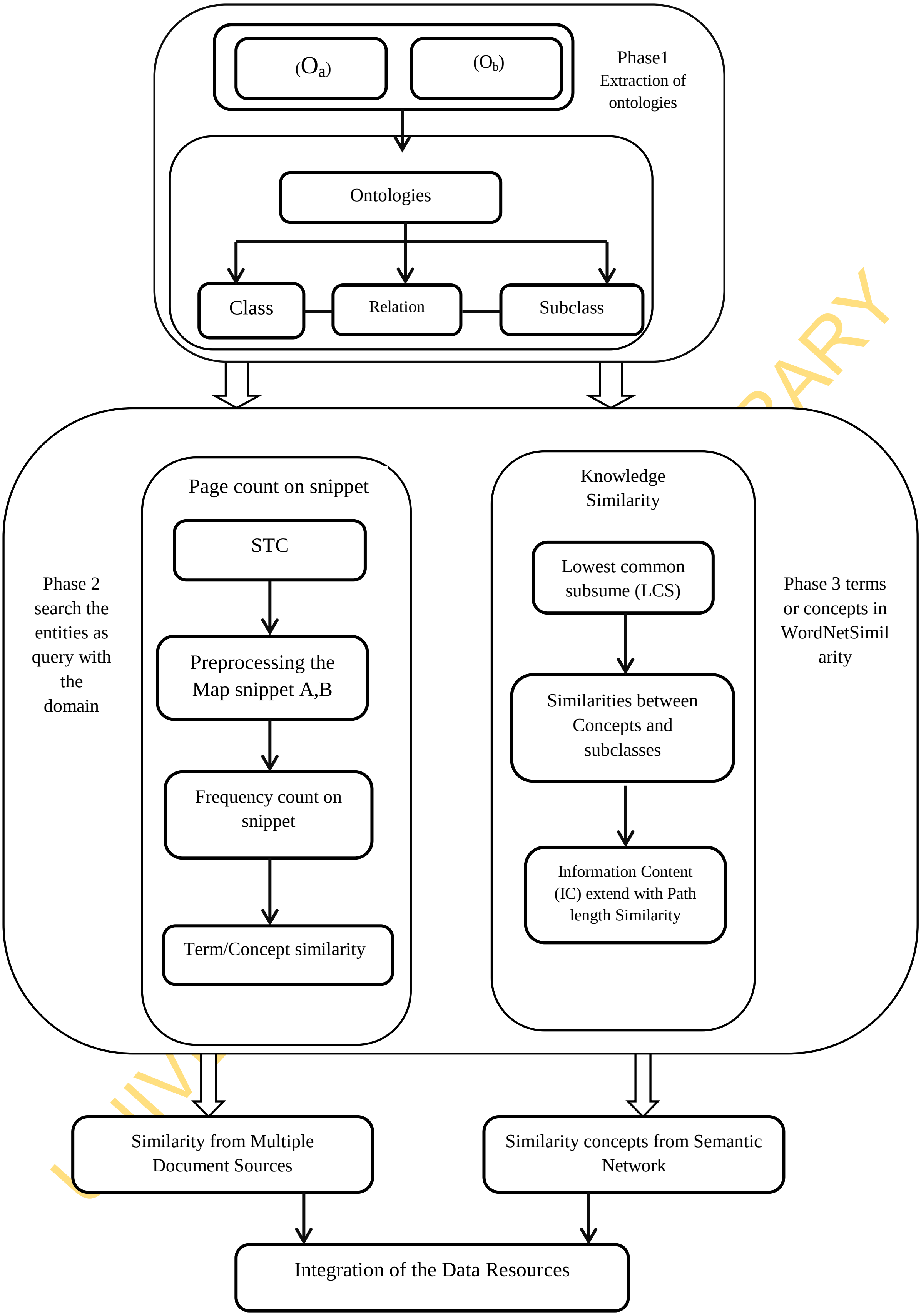


Figure 3.14: Conceptual Knowledge Model for Term Similarity

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction

This section provides the text analysis of the performance of different similarity scenarios in a quantitative way for the term similarity using Conceptual knowledge Model with each information source and the combination of the two information sources. This would verify that the approach is an applicable solution.

4.2 Dataset

The domain ontologies (the HumanResearch and SciMethodology) used for the term similarity contains 26 and 17 entities respectively. The ontology hierarchies reflect different conceptualisations of real world representation. Complete datasets are downloaded from SWEET and present in the data used for text analysis. Table 4.1 summarised the number of concepts, subconcepts and its relation to each ontology.

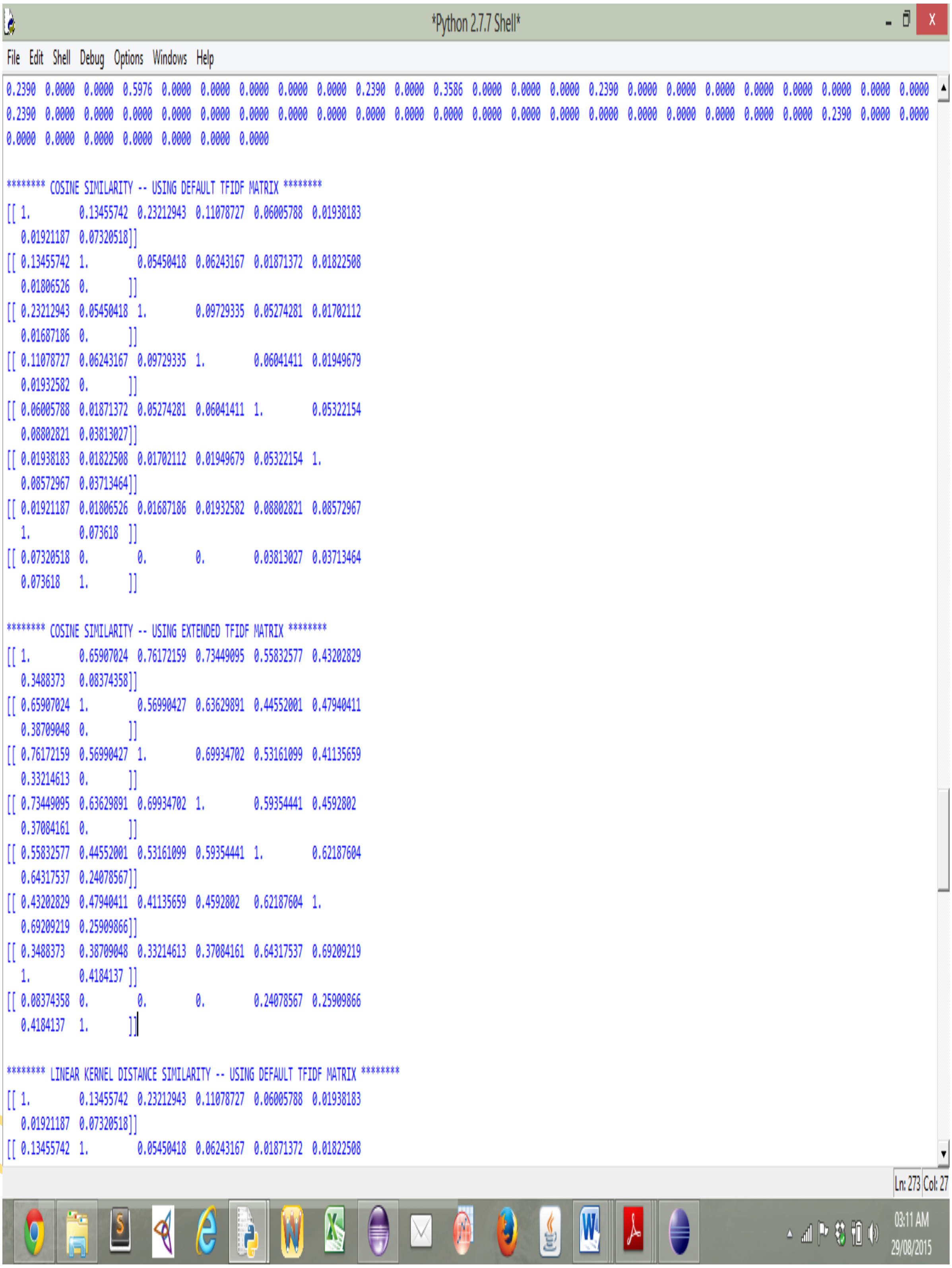
4.3 Feature Vector Normalisation on Term Similarity in MDSs

The term similarity in MDSs reflects the semantic implication of conceptualisation on the documents' analysis after filtered unwanted words. This depends on each term or concept of the ontology queries used. This is done in high level component so that it builds a dictionary of features and is transformed to feature vectors in form of matrices as depicted in Figure 4.1

Four documents (snippets) are chosen from each domain *A* and *B*. These generated 3,536 snippets (442x8 snippets) for each term and Concept Weighting (CW) respectively. These generated 21,216 datasets for six (6) different similarities but these

Table 4.1: The characteristics of the fraction of the domain ontologies

Ontologies	No of concept	No of sub-concept	No of relation	Max depth	Max depth of Concept
HumanResearch	5	2	19	3	10
SciMethodology	3	2	12	3	8



```
0.2390  0.0000  0.0000  0.5976  0.0000  0.0000  0.0000  0.0000  0.0000  0.2390  0.0000  0.3586  0.0000  0.0000  0.0000  0.2390  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000
0.2390  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.2390  0.0000  0.0000
0.0000  0.0000  0.0000  0.0000  0.0000  0.0000  0.0000

******** COSINE SIMILARITY -- USING DEFAULT TFIDF MATRIX ********
[[ 1.          0.13455742  0.23212943  0.11078727  0.06005788  0.01938183
   0.01921187  0.07320518]]
[[ 0.13455742  1.          0.05450418  0.06243167  0.01871372  0.01822508
   0.01806526  0.        ]]
[[ 0.23212943  0.05450418  1.          0.09729335  0.05274281  0.01702112
   0.01687186  0.        ]]
[[ 0.11078727  0.06243167  0.09729335  1.          0.06041411  0.01949679
   0.01932582  0.        ]]
[[ 0.06005788  0.01871372  0.05274281  0.06041411  1.          0.05322154
   0.08802821  0.03813027]]
[[ 0.01938183  0.01822508  0.01702112  0.01949679  0.05322154  1.
   0.08572967  0.03713464]]
[[ 0.01921187  0.01806526  0.01687186  0.01932582  0.08802821  0.08572967
   1.          0.073618  ]]
[[ 0.07320518  0.          0.          0.          0.03813027  0.03713464
   0.073618    1.        ]]

******** COSINE SIMILARITY -- USING EXTENDED TFIDF MATRIX ********
[[ 1.          0.65907024  0.76172159  0.73449095  0.55832577  0.43202829
   0.3488373   0.08374358]]
[[ 0.65907024  1.          0.56990427  0.63629891  0.44552001  0.47940411
   0.38709048  0.        ]]
[[ 0.76172159  0.56990427  1.          0.69934702  0.53161099  0.41135659
   0.33214613  0.        ]]
[[ 0.73449095  0.63629891  0.69934702  1.          0.59354441  0.4592802
   0.37084161  0.        ]]
[[ 0.55832577  0.44552001  0.53161099  0.59354441  1.          0.62187604
   0.64317537  0.24078567]]
[[ 0.43202829  0.47940411  0.41135659  0.4592802   0.62187604  1.
   0.69209219  0.25909866]]
[[ 0.3488373   0.38709048  0.33214613  0.37084161  0.64317537  0.69209219
   1.          0.4184137 ]]
[[ 0.08374358  0.          0.          0.          0.24078567  0.25909866
   0.4184137   1.        ]]

******** LINEAR KERNEL DISTANCE SIMILARITY -- USING DEFAULT TFIDF MATRIX ********
[[ 1.          0.13455742  0.23212943  0.11078727  0.06005788  0.01938183
   0.01921187  0.07320518]]
[[ 0.13455742  1.          0.05450418  0.06243167  0.01871372  0.01822508
```

Figure 4.1: Screenshot of the Feature Vector in different similarity measures

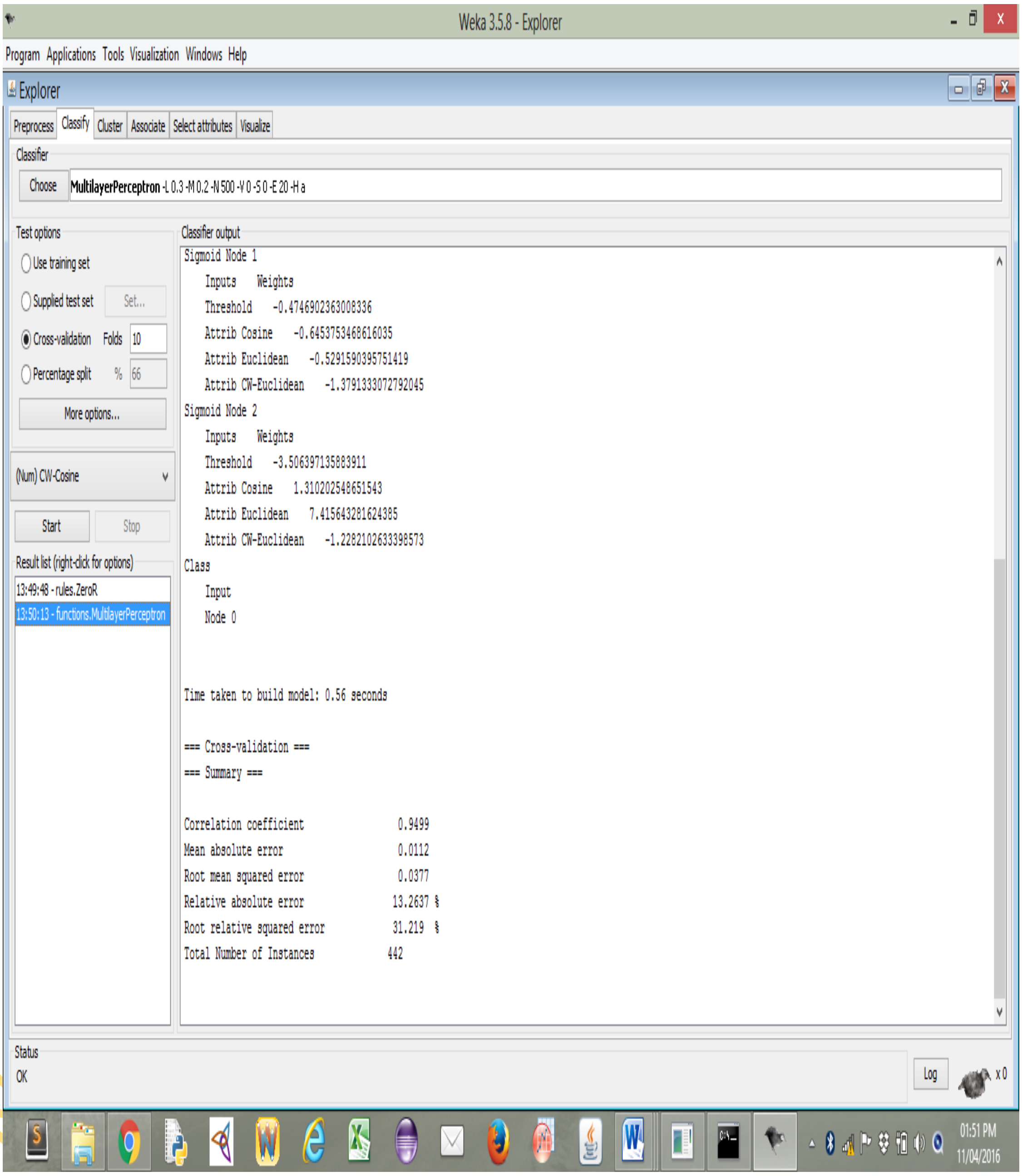
datasets cannot be used to determine accurate similarity of terms because of the dimension of the features; therefore, the mean values are taken for each similarity measure. This reduced the values to 2,652 datasets (442x6 similarities) and the similarities are Cosine and CW-Cosine, RBF and CW-RBF, and Euclidean and CW-Euclidean respectively as in Appendix I. Euclidean distance is converted to similarity *(I –Distance)* so that the datasets are on the same level of the similarity since distance is reciprocal of similarity. The RBT and CW-RBT generated scores that are approximately one (≈ 1) even for all zero and null snippets.

Due to the inconsistent values generated for RBT and CW-RBT, the scores yield poor generalisation abilities. Therefore, in the analysis of the similarity scores, four (4) measures are used using Machine Learning techniques: Support Vector Machine Regression (SVMReg), Neural Network (NN), Linear Regression (Linear Reg), M5P Tree and Bagging (Ensemble method).

4.4 Machine Learning Techniques on MDSs Scores

In this section, analyses are performed on the remaining four (4) similarity measures utilising the approach with the performance of established techniques using machine learning. Performance evaluation is one standard Term Frequency-Inverse Document Frequency (TF-IDF) weights as a baseline and the Concept Weighting (CW) estimation which are done by setting all parameters to $0 \leq S_X \leq 1$(where x are the similarity scores) using different similarity measures. Nirgude *et al*. (2013) used Support Vector Machine (SVM) to get optimal results. The method performance is measured by using Pearson correlation value. But different correlation measures are used to show the performance of the similarity in terms and concepts weighting.

The similarity scores are described using Attribute Relation File Format (ARFF) and represent the list of instances of the datasets with the set of attributes in continuous values. The Figure 4.2 shows screenshot of the analysis of MDSs dataset in WEKA using different machine learning techniques for similarity coefficient with the Mean Absolute Error (MAE).


Weka 3.5.8 - Explorer
Program Applications Tools Visualization Windows Help
Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Classifier
Choose
Test options
Set...
More options...
Start
Stop
Classifier output
Result list (right-click for options)
Status
Log
x 0
01:51 PM
11/04/2016

MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Num) CW-Cosine

13:49:48 - rules.ZeroR
13:50:13 - functions.MultilayerPerceptron

```
Sigmoid Node 1
    Inputs    Weights
    Threshold    -0.4746902363008336
    Attrib Cosine    -0.6453753468616035
    Attrib Euclidean    -0.5291590395751419
    Attrib CW-Euclidean    -1.3791333072792045
Sigmoid Node 2
    Inputs    Weights
    Threshold    -3.506397135883911
    Attrib Cosine    1.310202548651543
    Attrib Euclidean    7.415643281624385
    Attrib CW-Euclidean    -1.2282102633398573
Class
    Input
    Node 0


Time taken to build model: 0.56 seconds


=== Cross-validation ===
=== Summary ===


Correlation coefficient                  0.9499
Mean absolute error                      0.0112
Root mean squared error                  0.0377
Relative absolute error                 13.2637 %
Root relative squared error             31.219  %
Total Number of Instances              442
```

OK

Figure 4.2: Screenshot of Analysis of Machine Learning on MDSs Dataset

The Table 4.2 and Table 4.3 show the similarity coefficient for each of the similarity measures with its Mean Absolute Error (MAE). The result indicates the level of correlation coefficient of each similarity measures, while the bar chart is depicted in Figure 4.3.

Discussion on the Analysis of ML techniques on Document Similarity

Both Radial Basis Function (RBT) and CW-RBT generated 442x2=884 measures. This value is deducted from the 2,652 terms and concepts. This is due to inconsistency in values $(0.9 \leq x \leq 1)$ for null and zero snippets in the feature vectoriser.

The resulting dataset is 1,768 for Cosine, Euclidean, CW-Cosine and CW-Euclidean. Table 4.2 shows scores for similarity coefficient for Support Vector Machine Regression (SVMReg), Neural Network (NN), Linear Regression (Linear Reg), M5P Tree and Bagging. In order to select the best similarity coefficient, Neural Network (NN) and Bagging (ensemble method) are chosen. The values of NN are 0.881, 0.446, 0.949 and 0.964 for Cosine, Euclidean, CW-Cosine and CW-Euclidean with Mean Accuracy Error (MAE) of 0.058, 0.010, 0.014, and 0.008 respectively; while Bagging are 0.971, 0.415, 0.957 and 0.974 with Mean Accuracy Error (MAE) of 0.016, 0.008, 0.0137 and 0.008. It shows that the concept weighting method performs well than the existing term method. However, similarity attributes scores correlated well with the bagging technique except for Euclidean measure. The higher the coefficient of a similarity scores the better the method.

The similarity scores generated at this level cannot be used to determine the level of semantic terms or concepts similarity in MDSs. Therefore, a semantic network similarity is used to adjust the level of semantic similarity scores.

Table 4.2: Summary of Machine Learning on MDSs Similarity

Similarity	**SVMReg**	NN	**Linear Reg.**	**M5P Tree**	**Bagging**
Cosine	0.659	0.881	0.671	0.397	0.971
CW-Cosine	0.927	0.949	0.936	0.938	0.957
Euclidean	0.346	0.446	0.528	0.408	0.415
CW-Euclidean	0.930	0.964	0.929	0.957	0.974

Table 4.3: Mean Absolute Error on MDSs Similarity

Similarity	SVMReg	NN	Linear Reg.	M5P Tree	Bagging
Cosine	0.104	0.058	0.105	0.054	0.016
CW-Cosine	0.011	0.014	0.015	0.013	0.013
Euclidean	0.008	0.010	0.009	0.008	0.008
CW-Euclidean	0.008	0.008	0.011	0.007	0.008

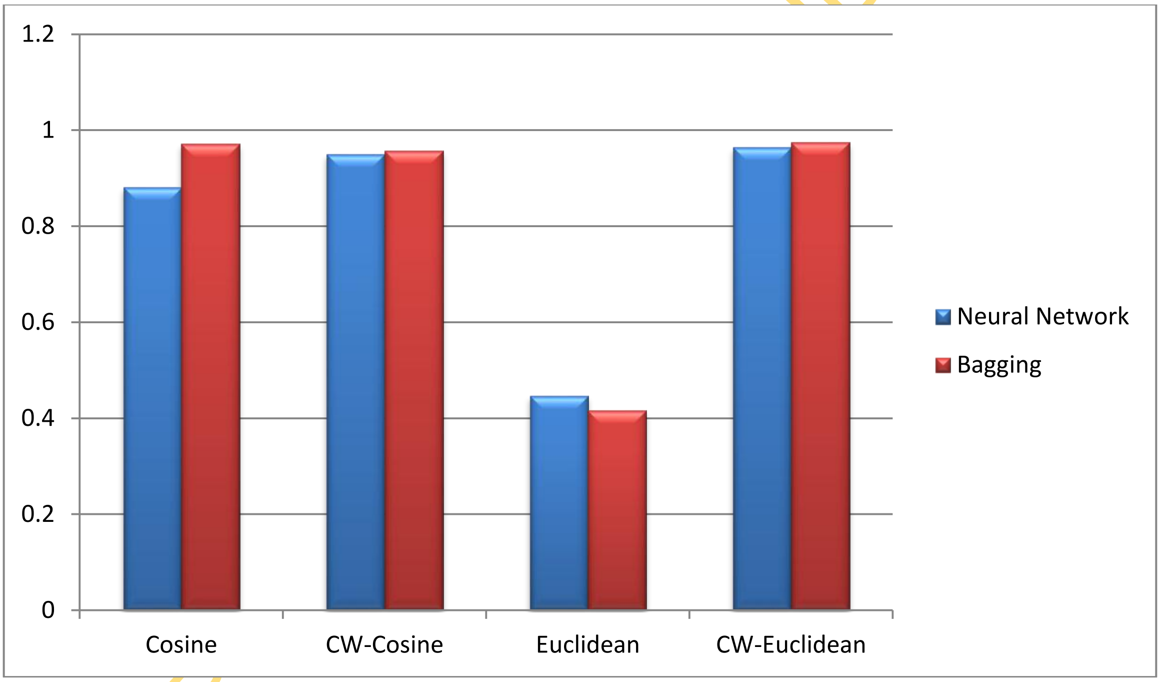


Figure 4.3: A Bar Chart for MDSs Similarity

4.5 Semantic Network on Term Similarity in Knowledge Source

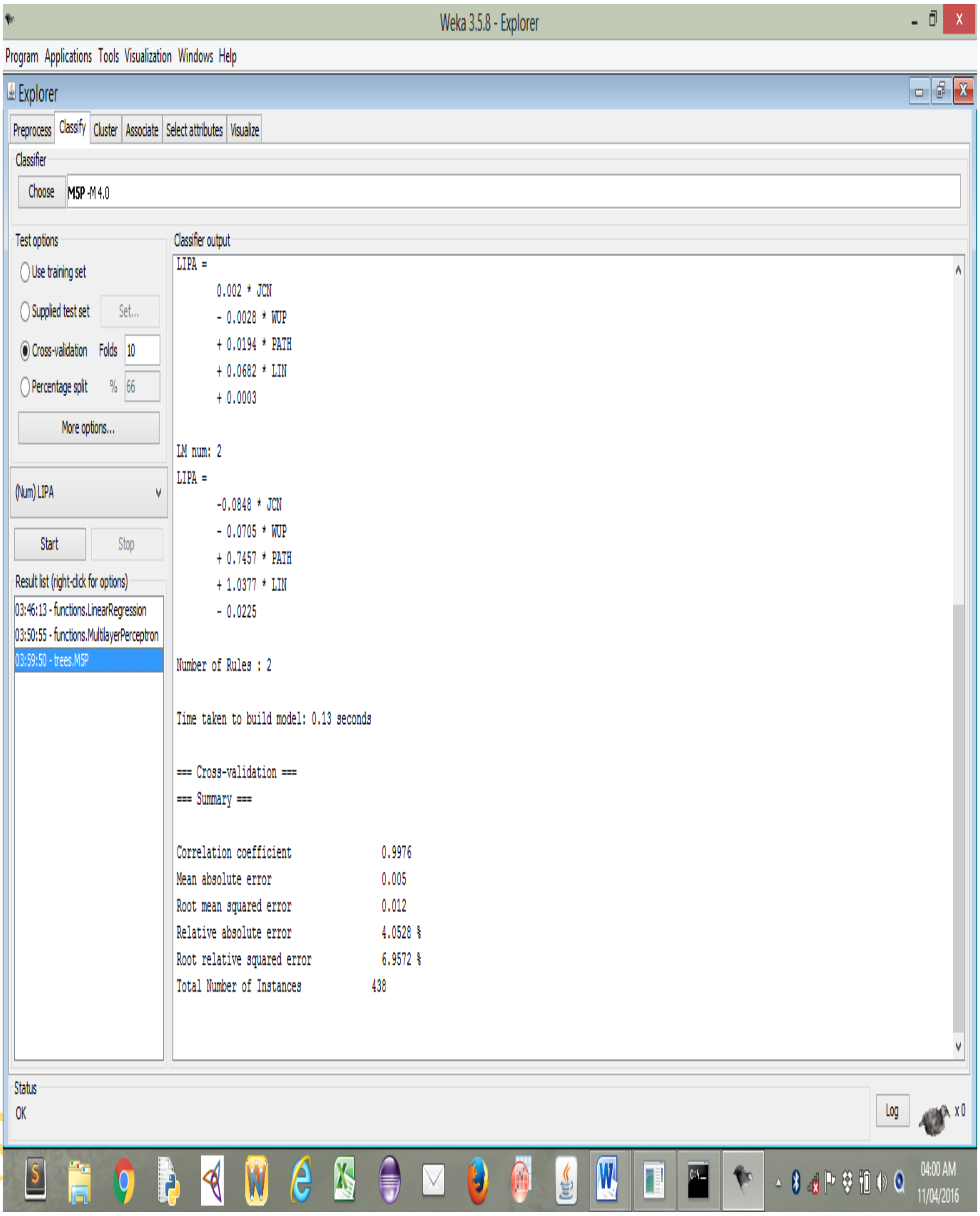
The term similarity in knowledge source reflects semantic meaning of terms in WordNetSimilarity. The datasets used in MDSs are also used after stemmed the words and the noun concepts of each term with its hypernym and are found for *A* and *B* in WordNetSimilarity.

Term similarity is found for four (4) existing knowledge similarities: Lin, JCN, WUP, and Path with a developed LIPA similarity. This generated 2,652 dataset for the five (5) similarity measures which are filtered for scores greater the 1. The resulting 2,190 (438x5) datasets are used for analysis and this determines the similarity scores ranging $0 \le S_{a,b} \le 1$. The performance analysis of WordNetSimilarity for Information Content (IC) similarity serves as a baseline for knowledge source and the developed LIPA. An Attribute Relation File Format (ARFF) is used to describe the list of instances of the dataset with the set of attributes in continuous values. Appendix II shows the scores for the terms/concepts *A* and *B* in WordnetSimilarity.

4.8 Machine Learning Techniques on WordNetSimilarity Scores

In the analysis of the similarity scores, similarity measures: Lin, Wu and Palmer (WUP), Jiang and Conrath (JCN), Path and developed LIPA are used in Machine Learning techniques to determine performance of the measures. The performance is measured using Pearson correlation value, but different correlation coefficient measures are used to show the performance of the similarity in WordNet.: Support Vector Machine Regression (SVMReg), Neural Network (NN), Linear Regression (Linear Reg), M5P Tree and Bagging (Ensemble method). The similarity scores are described using Attribute Relation File Format (ARFF) and represent the list of instances of the datasets with the set of attributes in continuous values. The Figure 4.4 shows screenshot of the analysis of WordNetSimilarity in WEKA using different machine learning techniques for correlation coefficient with the Mean Absolute Error (MAE).

The Table 4.4 and Table 4.5 show the summary of machine learning on each of the similarity measures (Support Vector Machine Regression (SVMReg), Neural Network (NN), Linear Regression, M5P tree and Bagging) with its respective Mean Absolute Error while the bar chart is shown in Figure 4.5. The result indicates the level of similarity coefficient (correlation) of each measure to other measures.



Classifier: M5P -M 4.0

○ Use training set
○ Supplied test set
● Cross-validation Folds 10
○ Percentage split % 66

(Num) LIPA

03:46:13 - functions.LinearRegression
03:50:55 - functions.MultilayerPerceptron
03:59:50 - trees.M5P

```
LIPA =
        0.002 * JCN
        - 0.0028 * WUP
        + 0.0194 * PATH
        + 0.0682 * LIN
        + 0.0003

LM num: 2
LIPA =
        -0.0848 * JCN
        - 0.0705 * WUP
        + 0.7457 * PATH
        + 1.0377 * LIN
        - 0.0225

Number of Rules : 2

Time taken to build model: 0.13 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9976
Mean absolute error                      0.005
Root mean squared error                  0.012
Relative absolute error                  4.0528 %
Root relative squared error              6.9572 %
Total Number of Instances              438
```

Status: OK

Figure 4.4: Screenshot of Analysis of ML on Semantic Network Similarity

Table 4.4: Summary of ML on Semantic Network Similarity

Similarity	SVM Reg	Neural Network	LINEAR Reg	M5P Tree	Bagging
LIPA	0.996	0.997	0.996	0.997	0.989
JCN	0.837	0.901	0.821	0.867	0.805
WUP	0.856	0.951	0.866	0.953	0.966
PATH	0.880	0.944	0.837	0.955	0.962
LIN	0.996	0.998	0.995	0.995	0.988

Table 4.5: Mean Absolute Error (MAE) on Semantic Networtk Similarity

Similarity	SVMReg	Neural Network	LINEAR Reg	M5P Tree	BAGGING
LIPA	0.006	0.006	0.007	0.005	0.005
JCN	0.013	0.014	0.017	0.009	0.011
WUP	0.066	0.042	0.071	0.0399	0.027
PATH	0.014	0.010	0.018	0.0079	0.005
LIN	0.005	0.005	0.006	0.0051	0.006

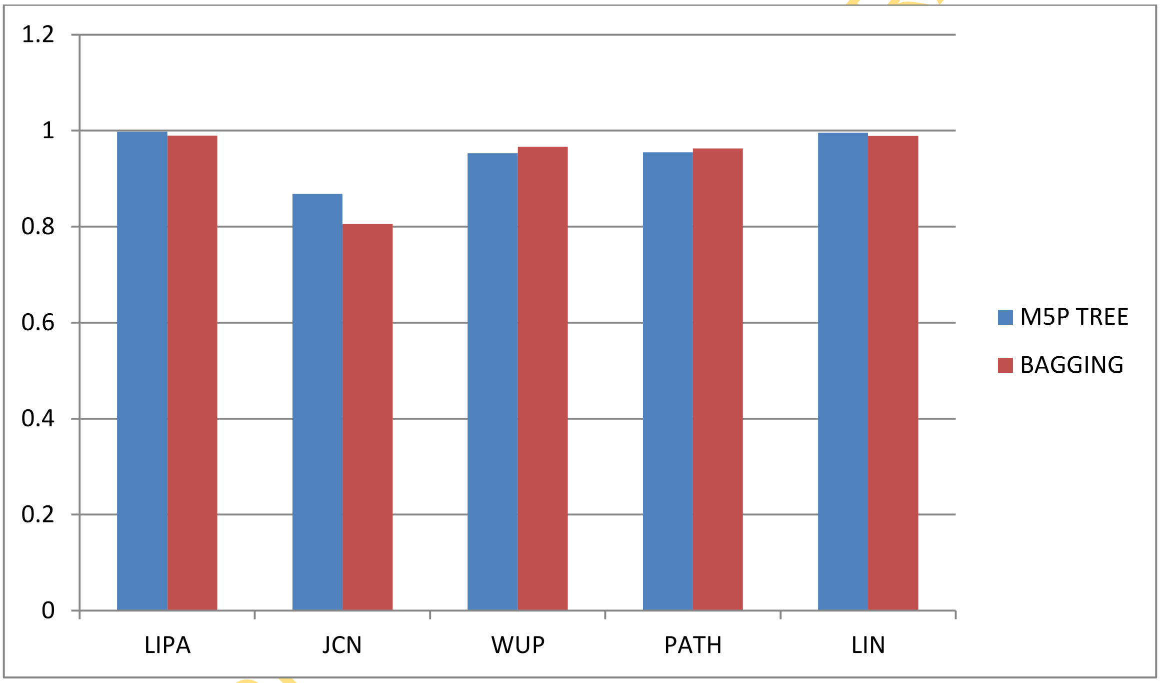


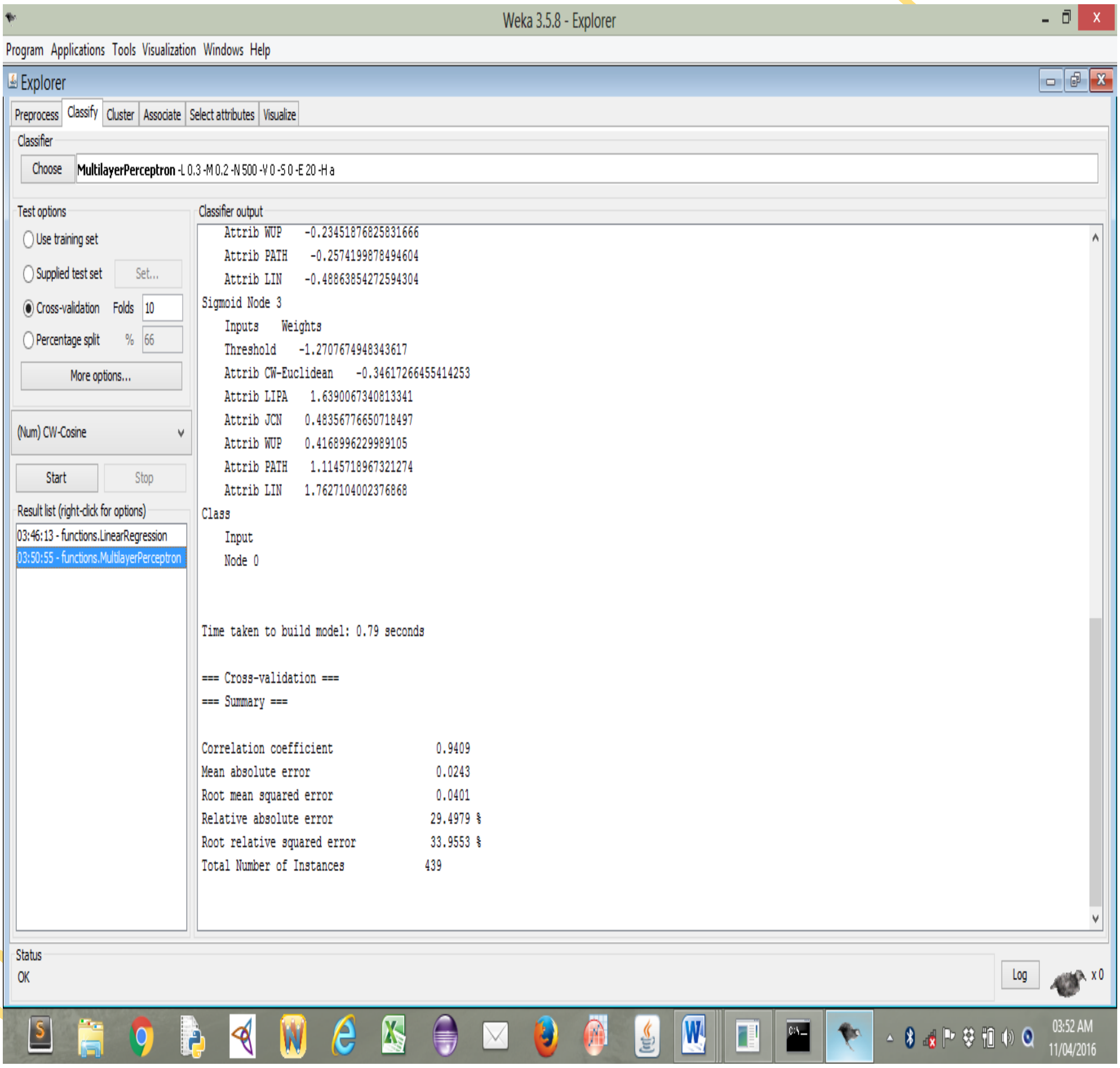
Figure 4.5: A Bar Chart for Semantic Network Similarity

Discussion on the Analysis of ML techniques on Knowledge Similarity

The resulting dataset is 2,190 (438x5) scores for JCN, WUP, Lin, Path and LIPA. The result generated supports the idea of Hirst and Budanitsky (2006) and Pirro (2009) that considered semantic network (WordNet) as a better choice for estimating semantic similarity than other lexical resources. In order to select the best performance evaluation generated for the result, M5P tree (single method) and Bagging (ensemble method) are chosen. The generated semantic similarities values are 0.868, 0.953, 0.955, 0.995 and 0.998 for JCN, WUP, PATH, LIN and LIPA respectively with Mean Accuracy Error (MAE) of 0.009, 0.040, 0.008, 0.005 and 0.005 respectively; while bagging are 0.8053, 0.966, 0.962, 0.988 and 0.988 respectively with Mean Accuracy Error (MAE) of 0.0117, 0.027, 0.005, 0.006 and 0.009. This indicates a better result compared to the term/concept similarity for all the measures. From the results, JCN has the lowest score from the semantic network because it is hybrid distance measure. However, the higher the correlation coefficient of a similarity measures the better the method. Moreover, the major reason why a path measure needs to be added to the information content is to reflect the taxonomy ideology to the semantic similarity and improve the measure. Therefore, the bar chart is shown in the Figure 4.5.

4.9 Conceptual Knowledge Model for Term Similarity in Information Sources

The Conceptual Knowledge presented the concept analysis of two information sources used in MDSs and knowledge source. Analysis for Conceptual Knowledge Model is on Concept Weighting (CW-Cosine and CW-Euclidean) in MDSS and all the five measures in knowledge similarity. Nitish et al. (2012) presented two structured information sources from latent semantic indexing and knowledge source for analysis of semantic similarity of concepts *A* and *B* using Linear Regression and Bagging. The scores for CW-Cosine, CW-Euclidean are combined with JCN, WUP, PATH, LIN and LIPA to form the Conceptual Knowledge semantic similarity of terms as shown in Figure 4.6. The performance is measured using Pearson correlation scores, different correlation coefficient measures are used to show the performance of the semantic similarity with machine learning techniques.: Support Vector Machine Regression (SVMReg), Neural Network (NN), Linear Regression (Linear Reg), M5P Tree and Bagging (Ensemble method). The similarity scores are described using Attribute Relation File Format (ARFF) and represent the list of instances of the datasets with the



Classifier

MultilayerPerceptron -L 0.3 -M 0.2 -N 500 -V 0 -S 0 -E 20 -H a

Test options

○ Use training set

○ Supplied test set Set...

● Cross-validation Folds 10

○ Percentage split % 66

(Num) CW-Cosine

Result list (right-click for options)

03:46:13 - functions.LinearRegression

03:50:55 - functions.MultilayerPerceptron

Classifier output

```
    Attrib WUP    -0.23451876825831666
    Attrib PATH    -0.2574199878494604
    Attrib LIN    -0.48863854272594304
Sigmoid Node 3
    Inputs    Weights
    Threshold    -1.2707674948343617
    Attrib CW-Euclidean    -0.34617266455414253
    Attrib LIPA    1.6390067340813341
    Attrib JCN    0.48356776650718497
    Attrib WUP    0.4168996229989105
    Attrib PATH    1.1145718967321274
    Attrib LIN    1.7627104002376868
Class
    Input
    Node 0


Time taken to build model: 0.79 seconds

=== Cross-validation ===
=== Summary ===

Correlation coefficient                  0.9409
Mean absolute error                      0.0243
Root mean squared error                  0.0401
Relative absolute error                 29.4979 %
Root relative squared error             33.9553 %
Total Number of Instances              439
```

Status

OK

Figure 4.6: Screenshot of Analysis of Conceptual Knowledge Similarity

set of attributes in continuous values. The Figure 4.6 shows screenshot of the analysis of Conceptual Knowledge in WEKA using different machine learning techniques for correlation coefficient with the Mean Absolute Error (MAE).

The Table 4.6 and Table 4.7 show the summary of machine learning on each of the similarity measures (Support Vector Machine Regression (SVMReg), Neural Network (NN), Linear Regression, M5P tree and Bagging) with its respective Mean Absolute Error while the bar chart is shown in Figure 4.7. The result indicates the level of similarity coefficient (correlation) of each measure to other measures.

Table 4.6: Summary of ML on Conceptual Knowledge Similarity

Similarity	SVMReg	NN	Linear Reg.	M5P Tree	Bagging
CW-Cosine	0.950	0.940	0.951	0.951	0.945
CW-Euclidean	0.950	0.943	0.943	0.955	0.955
LIPA	0.996	0.997	0.996	0.995	0.991
JCN	0.673	0.661	0.645	0.744	0.633
WUP	0.721	0.928	0.632	0.951	0.963
Path	0.722	0.923	0.688	0.688	0.962
Lin	0.996	0.995	0.996	0.994	0.993

Table 4.7: Mean Absolute Error on Conceptual Knowledge Similarity

Similarity	SVMReg	NN	Linear Reg	M5P Tree	Bagging
CW-Cosine	0.011	0.024	0.014	0.013	0.014
CW-Euclidean	0.009	0.015	0.013	0.013	0.013
LIPA	0.007	0.009	0.007	0.005	0.008
JCN	0.017	0.023	0.032	0.014	0.020
WUP	0.076	0.050	0.085	0.037	0.029
Path	0.016	0.013	0.021	0.021	0.006
Lin	0.006	0.008	0.007	0.005	0.006

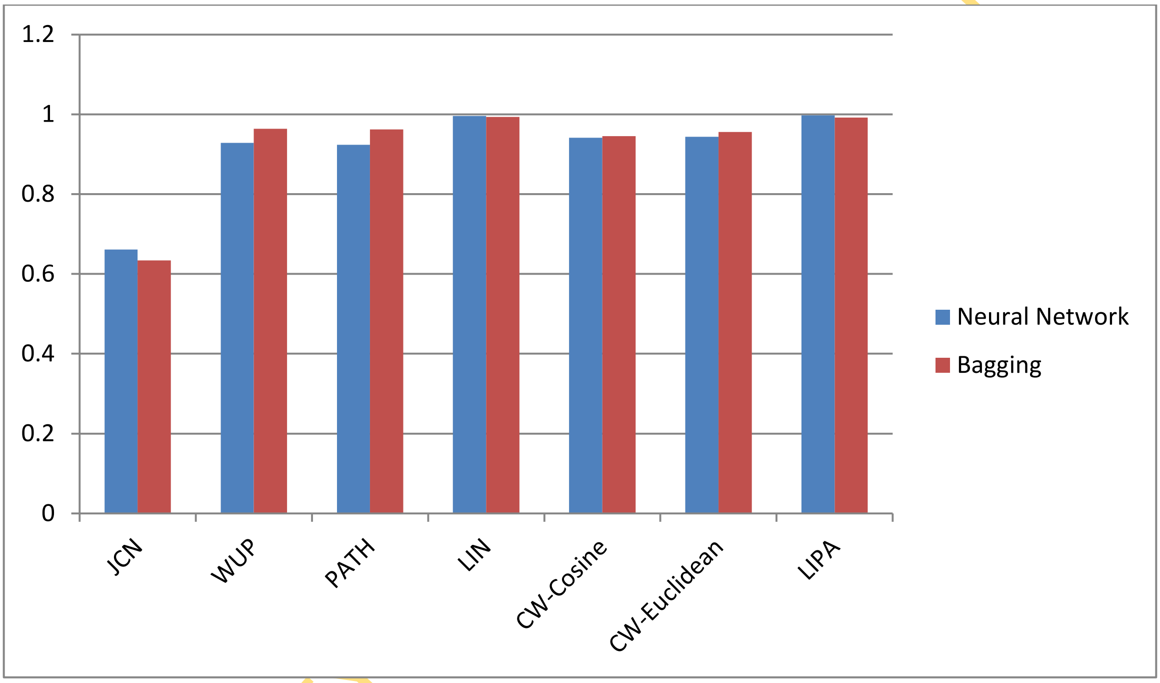


Figure 4.7: A Bar Chart for Conceptual Knowledge Similarity

Discussion on the Analysis of ML Techniques on Combined Similarity

The resulting datasets for JCN, WUP, PATH, LIN, CW-Cosine, CW-Euclidean and LIPA formed the Conceptual Knowledge Model for the two sources. The scores generated 3,094 dataset yielded after filtering the dataset for values greater the one (1). In order to select the best performance coefficient generated for the scores, neural network (single method) and Bagging (ensemble method) are chosen. The values are 0.661, 0.928, 0.923, 0.995, 0.940, 0.943 and 0.997 with MAE as follows: 0.022, 0.049, 0.011, 0.007, 0.024, 0.014 and 0.008 respectively for neural network. The Bagging values are 0.633, 0.963, 0.962, 0.993, 0.997, 0.945, 0.955 and 0.991 for JCN, WUP, PATH, LIN, CW-Cosine, CW-Euclidean and LIPA respectively. The Mean Accuracy Error (MAE) is: 0.019, 0.028, 0.006, 0.006, 0.012, 0.013 and 0.008 respectively. The higher the correlation of the similarity scores the better the method. It shows that JCN (hybrid) has semantic relation measures with low semantic similarity scores compared to other measures (JCN is a hybrid distance measure). The conceptual knowledge model generated better result for each similarity measures with low mean accuracy values. It is concluded that combined MDSs with semantic network predict better than single individual similarity measure.

CHAPTER FIVE

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

5.1 The Summary of the Research

Similarity is important in information retrieval of web documents. The search engine also plays an important role in finding relation among input keywords but fails in retrieving semantically related documents. The retrieval system depends on the similarity between source document and the query. Therefore, present to the user document that match the user keyword. However, the retrieval system have limited ability to exploit the conceptualisation involved in user needs and content meaning due to its ability to describe the relation among search terms. Hence, Term Similarity (TS) is based on lexical matching and reflects users' information need. But TS has not bridge the gap of vocabulary mismatch problem in retrieval system. Furthermore, the TS in retrieval of web documents are based on single information source and represent document in a linear feature vector. Therefore, similarity of term is considered without context or structure. An appropriate additional information knowledge source is required for mapping document to terms. Such knowledge resources are WordNet dictionary and domain ontology that contain data for computing similarity of terms in a more structured way. This knowledge is used to index and describe the context of documents. The idea is that high-level of semantic content information is accurately modelled using conceptual indexing. Therefore, related documents that do not share terms are still represented by nearby conceptual descriptors. This requires that concept of Semantic Web technology, in which user' request is understandable by the retrieval system and reduced the mismatch of terms. Ontology is used to represent knowledge that could be understood by machine and human.

The ontology structure (IS-A hierarchical concept) are used for query expansion, indexing and retrieval. Two ontologies are extracted from Internet inform of XML and normalised with Protégé 4.2 to make it readable. The concepts of the ontology are used to formulate queries and expand with terms from domain of reference. This help to structure the query and give a better meaning. Moreso, the web resources as well need to be structures. The Carrot2 API is used with Google API to query and structure the web documents. The implementation is done in Java and this was used to cluster the web documents and presented the result in form of clusters. The cluster related to the domain ontology structure is chosen from retrieved documents. The retrieved documents are preprocessed using Natural Language Toolkit (NLTK) to reduce dimensionality of features. Concept Weighting estimation is developed in Python and used on the preprocessed documents to determine similarity scores. However, the similarity scores on Multiple Document Sources is not enough to determine the semantic similarity of retrieved documents. Therefore, a WordNetSimilarity is employed on the concepts query with Information Content and Path length. This takes into consideration the hierarchical structure of the concepts and improves the sematic similarity level. Consequently, the similarity scores on Concept Weighting estimation and WordNetSimilarity are combines using Machine Learning Techniques such as: Linear Regression, Neural Network, Support Vector Machine Regression (SVMReg), M5P tree and Bagging. This determines the correlation coefficient of the similarity scores and the process improves the result generated from MDSs and WordNetSimilarity

Conceptual Knowledge Model contributes to the solution of problem of web documents retrieval by making use of the knowledge about ontological relations between concepts. It was observed that mapping semantic knowledge information source which extend the multiple document sources blends the idea of ontological similarity and indexing in retrieval of web documents.

5.2 Contribution to Knowledge

The two extracted domain ontologies concepts used for queries are represented in the same web ontology language (OWL) and English. The dimensionality of features of terms retrieved from the web is reduced by clustering, stemming, morphological analysis and WordNet lemmatisation. The approaches worked on many-to-many comparison of terms on the developed Concept Weighting estimation for the feature vector and extension of information content similarity with path length in WordNetSimilarity. Consequently, these approaches centred on structured retrieved document and semantic analysis of concepts respectively. Finally, machine learning techniques is used to combine the similarity scores generated from the two information sources to give better semantic similarity scores. Though the semantic similarity of terms or concepts in retrieval of web documents is (semi) automatic. But improved the semantic similarity of term retrieved from web documents.

5.3 Conclusion

In computing the retrieval of web documents, domain-query is incorporated into the term similarity scores. In practical building of the retrieval system, a WordNet clustering tool (CARROT2) was used to structure and cluster the web to achieve a better similarity of terms. This was done by matching the query term with the document index which reduced the word mismatch and ambiguity on web documents. It was observed that incorporating domain-query with suffix tree clustering leads to significant increase in performance of retrieval system.

However, the effects of different weighting approaches were investigated for the term and concept method. It was concluded that the concept weighting method performed better than term weighting method of different term similaries. The evaluated results suggested that the method was more effective than the existing benchmark method. The result of the TF-IDF weighting method in the similarity did not yield a measureable advantage. Consequently, in the knowledge model, the combination of information content with path length increases the performance of the concept similarity better than similarity of Information Content and Path length separately.

Furthermore, conceptual knowledge model improves retrieval of web documents with multiple document sources that are structured. This leads to reduction of false positives, increased performance of the retrieval system and the practical solution of semantic language mismatch and ambiguity with robust similarity between terms or concepts.

5.4 Recommendations

The Conceptual Knowledge Model (Concept Weighting in Multiple Document Sources and Semantic Network Similarity) would serve as the enabler into the semantic retrieval system which is an open problem. It is suggested that the methods should be implemented in the current web to enhance the semantic similarity of terms or concepts in the retrieval system. This will reduce language mismatch and ambiguity of term problems which would in turn enhance precision.

REFERENCES

Ahmed, M.S. and Amar, M. K. 2010. Semantic web search results clustering using lingo and wordNet. In: IJRRCS: Kohat University of Science and Technology (KUST) Pakistan, 1.2: 71–76.

Alipanah. N., Parveen, P., Menezes, S., Kha, L., Seida, S.B., and Thuraisingham, B. 2010. Ontology-driven query expansion methods to facilitate federated queries. In: Proceedings of 2010 IEEE International Conference on Service-Oriented Computing Applications (SOCA). IEEE, 1–8.

Arabshian, K., Danielsen, P., and Afroz, S. 2012. LexOnt: A semi-automatic ontology creation tool for programmable web. AAAI Technical report Spring Symposium Series in 2012, http://www.programmableweb.com, January, 2-8.

Arpírez, J.C, Corcho, O., Fernández-López, M., and Gómez-Pérez, A. 2001. WebODE: a Scalable Workbench for Ontological Engineering. First International Conference on Knowledge Capture (KCAP01).Victoria. Canada. October.

Avraham, S., Tung, C., Ilic K., Jaiswa,l P., Kellogg, E., McCouch, S., Pujar, A., Reiser, S. L, Rhee, Y., Sachs, M., Schaeffer, M., Stein L,, Stevens, P., Vincen, L., Zapata, F., and Ware, D. 2008. The plant ontology database: a community resource for plant structure and developmental stages controlled vocabulary and annotations, nucleic acids research. Oxford University Press, 449-454.

Baader, F., Calvanese, D., McGuinnes, D., Nardi, D., and Patel-Schneider, P. F. 2003. The description logic handbook: theory, implementation and applications. Reprinted 2004 in U.K at Cambridge University Press, 230-235.

Baeze-Yates, R. and Ribeiro-Neto, B. 1999. Modern Information Retrieval. Addison Wesley. Xxiv, 521-531.

Battista, A.D.L., Villanueva-Rosales, N. Palenychka, M. and Dumontier, M. 2007. SMART: A web-based, ontology-driven, semantic web query answering application. 33-62

Baziz, M., Boughanem, M., Aussenac-Gilles, N. 2004. The Use of Ontology for Semantic Representation of Documents. In The 2nd Semantic Web and Information Retrieval Workshop (SWIR), SIGIR 2004, Sheffield UK, 29. Yin Ding, Keith van Rijsbergen, Iad Ounis, Joemon Jose (Eds) July 38-45.

Belkin, N. J. and Croft, W. B. 1992. Information filtering and information retrieval: two sides of the same coin? Communications of the ACM 35.12: 29–38.

Bennacer, N., and Karoui, L. 2005. A framework for retrieving conceptual knowledge from Web pages. In Semantic Web Applications and Perspectives, Journal of Artificial Research (JAIR), University of Toronto, Toronto, Italy, 24:305-313.

Benz, D. and Hotho A. 2007. Ontology learning from folksonomies.Lernen, Wissen, Adaptation, (LWA’07) Workshop Proceedings, 109-112.

Berners-Lee, T., Hendler, J., Lassila, O. 2001. The Semantic Web. A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. In Scientific American, Mai http://www.scientificamerican.com/2001/0501issue/0501berners-lee.html. 285.5:34-43

Berry, M. W, Drmac, Z. Jessup, E.R.. 1999. Matrices, vector spaces and information retrieval, SIAM Rev, 41(2) 335-362.

Beyer, K., Goldstein J., Ramakrishnan, R. and Shaft, U. 1999. When is `nearest neighbour' meaningful. In Proceedings of ICDT-1999, 217-235.

Bhogal, J. Macfarlane, A. and Smith, P. 2007. A review of ontology based query expansion Information Processing and Management, ISSN 0306-4573.doi: 43.4: 866-886. http://dx.doi.org/10.1016/j.ipm.2006.09.003

Biatov, K; Khehler, J and Schneider, D. 2009. Semanting computing Audio clip content comparison using latent semantic indexing. 3rd International Conference SE'09. IEEE, 10:

Bloehdorn, S., and Hotho, A. 2004. Boosting for text classification with semantic features. In Proceedings of the 6th International Workshop on Knowledge Discovery on the Web (WebKDD) 149-166.

Bollegala D, Matsuo Y. and Ishizuka M. 2007. Measuring semantic similarity between words using web search engines. Proceeding of International Conference on World Wide Web, 757-766.

Bollegala, D., Matsuo, Y. and Ishizuka, M. 2011. A web search engine-based approach to measure semantic similarity between words", IEEE Transactions on Knowledge and Data Eng., 23.7: 977-982.

Bonino, D., Corno, F., Farinetti, L., and Bosca, A., 2004. Ontology driven semantic search. WSEAS Transaction on Information Science and Application 1.6: 1597–1605.

Boubekeur, F. and Azzoug, W. 2013. Concept-based indexing in text information retrieval International Journal of Computer Science and Information Technology (IJCSIT) DOI : 10.5121/ijcsit.2013.5110 , February 5.1: 119 – 136.

Bray, T. 2004. Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation, Boston, http://www.w3.org/TR/REC-xml.

Brickley, D. and Guha, R. V. 2000. Resource Description Framework (RDF) Schema Specification 1.0. Candidate recommendation, World Wide Web Consortium. http:// www.w3.org/2000/01.rdf-schema#

Buitelaar, P., Cimiano, P. and Magnini, B 2005. Ontology learning from text: An overview ontology learning from text: Methods, evaluation and applications.

Frontiers in Artificial Intelligence and Applications Series 123 Computer Science Semantic Web July, 123:180 .

Burkart, M. Kuhlen, T. Seeger, and Strauch D. 2004. Thesaurus. In: R.: Grundlagen der praktischen Information und Documentation. München, Saur Verlag.579-590.

Cardoso, J. 2007. The Semantic Web Vision: Where are we? IEEE Intelligent Systems, 22.5: 84-88.

Carpineto C.and Romano. G. 2012. A Survey of Automatic Query Expansion in Information Retrieval‖. ACM Comput. Surv.. DOI 10.1145/2071389.2071390 http://doi.acm.org/10.1145/2071389.2071390. January 44.1:1-50.

Carrot2 2009. Carrot2 an Open Source Search Results Clustering Engine. (http//project.carrot2.org/download java-api html).

Castells, P., Fernandez, M., and Vallet, D. 2007. An adaptation of the vector-space model for ontology-based information retrieval. IEEE Trans. on Knowl. and Data Eng., ISSN 1041-4347. 19.2:261–272. doi: http://dx.doi.org/10.1109/TKDE.2007.22.

Chatterjee, R. and Pushplata, M 2012. An analytical assessment on document clustering. In: I. J. Computer Network and Information Security, published online in MECS (http://www.mecs-press.org/). June 5: 63-71.

Chen, H., Lin, M., and Wei, Y. 2006. Novel association measures using web search with double checking, Proc. 21st International Conference on Computational Linguistics and 44th Ann. Meeting of the Assoc. for Computational Linguistics (COLING/ACL), 1009-1016.

Chim, H. and Deng, X. 2008: Efficient phrase-based document similarity for clustering, IEEE Transaction on Knowledge and Data Engineering, September, 20:, 1217-1229.

Choonghyun, H. and Choi, J. M. 2010. Effect of Latent Semantic Indexing for Clustering Clinical Documents., MD, PhD 978-0-7695-4147-1/10 Proceeding of the 2010 IEEE/ACIS Conference On Computer And Information Science, August, Japan. 561-566.

Chowdhury, G. and Chowdhury, S. 2002. Digital Libraries Research, major issues and trends, Journal of Documentation Facet publishing, 55:4, 408-448.

Christopher D. M. and Hinrich S. 2001. Foundations of Statistical natural language processing. MIT Press. Cambridge, Massachusetts. 529-574.

Chung, S. M., Holt, J. D, and Li, Y. 2008. Text document clustering based on frequent word meaning sequences, Data & Knowledge Engineering, 64.1: 381-404.

Cilibrasi, R.L. and Vitanyi, P.M.B. (2007). The Google Similarity Distance, IEEE Trans. Knowledge and Data Engineering, 19:3, 370-383.

Cimiano, P. and Volker, J. 2005. Text2Onto – A Framework for Ontology Learning and Data-driven Change Discovery Proceedings of the 10th International Conference on Applications of Natural Language to Information Systems (NLDB), 3513: 227-238.

Cimiano, P. Haase, P., Herold, M. Mantel, M. and Buitelaar, P. 2007. LexOnto: A Model for ontology lexicon for ontology-based NLP. In Proceedings of the OntoLex07 Workshop (held in conjunction with ISWC'07).

Cimiano, P.and Staab, S. 2004. Learning by Googling. SIGKDD Explor. Newsl., 6.2:24–33,

Ciorăscu, C., Ciorăscu, I. and Stoffel, K. 2003. knOWLer - Ontological Support for Information Retrieval Systems. In Proceedings of SIGIR 2003 Conference, Workshop on Semantic Web, Toronto, August, Canada.

Conesa, J., Storey, V.C. and Sugumaran, V. 2006. Using Semantic Knowledge to Improve Web Query Processing, In: NLDB 2006, Springer-Verlag Berlin, 106 – 117.

Croft, W. B., Bendersky, M.; and Metzler, D. 2010. Learning concept importance using a weighted dependence model. In WSDM, 31–40.

Daconta, M., Obrst, L. and Smith, K. 2003. The Semantic Web, Wiley Publishing.

Daqing, H. and Dan, W., 2010. Enhancing query translation with relevance feedback in translingual information retrieval. Information Processing and Management, 47.1: 1–17.

Davies, J., Fensel, D.and Harmelen, V. F 2003 OilEd: http://oiled.man.ac.uk/ Ontology building editor developed by Knowledge Media Institute, The Open University, further information available, from: http://apollo.open.ac.uk/.

Davulcu, H., Vadrevu, S. and Nagarajan, S. 2004. OntoMiner: Bootstrapping Ontologies From Overlapping Domain Specific Web Sites. In: Poster presentation at the 13th International World Wide Web Conference, New York May 17-22.

Dean, M., and Schreiber, G., 2003. OWL, Web Ontology Language Reference, W3C Candidate Recommendation, http://www.w3c.org/TR/owl-ref/

Debole, F. and Sebastiani, F.. 2003 Supervised term weighting for automated text categorization. In Proc. of SAC-03, 18th ACM Symposium on Applied Computing Melbourne, US, 784-788.

Deerweste, S., Dumais, S. T, Furnas, G. W., Landaue, T. K and Harshman, R. 1990. Indexing by Latent Semantic Analysis. Journal of the American Society of Information Science, 41.1: 391-407.

Deerwester. S., Dumais, S.T., Funas, G.W. Landaeur, T.K. and Harshman, R. 1990. Indexing by Latent Semantic Analysis. Journal of the society for information science 41.6

Dellschaft, K. 2005. Measuring the similarity of concept hierarchies and its influence on the evaluation of learning procedures. Master's thesis, UniversityKoblenz Landau, Campus Koblenz, Fachbereich 4 Information, Institute for Computer visualisation.

Dice, L. 1945. Measures of the amount of ecologic association between species.Ecology, 26.3:

Dong, H., Hussain, F. K. and Chang, E. 2009. A Hybrid Concept Similarity Measure Model for Ontology Environment, Lecture Notes in Computer Science on the Move to Meaningful Internet Systems: OTM 2009 Workshops, 1.1:58-72.

Egozi, O., Markovitch, S., and Gabrilovich, E. 2011. Concept-Based Information Retrieval Using Explicit Semantic Analysis‖. ACM Transactions on Information Systems. April 2.8: 1-34.

Eissen, M., Sven, D. S. and Potthast, M. 2005. The suffix tree document model revisited. *In* Proceedings of the 5th International Conference on Knowledge Management, 596-603.

Eugene, F. K. 1987. Taxicab Geometry , Dover. ISBN 0-486-25202-7.

Fang, W.D., Zhang, L., Wang, Y.X. and Dong, S.B. 2005. Toward a semantic search engine based on ontologies. In: Proceedings of 2005 International Conference on Machine Learning and Cybernetics, IEEE, 3:1913–1918.

Faure, D. 1999. Connaissances semantique acquises par Asium: Examples d'utilisations," In Journee du Reseau de science cognitives d'Ile-de France (RISC, ed), October 12:

Fellbaum, C., ed.1998. WordNet: An electronic lexical database, Language, Speech, and Communication. MIT Press, Cambridge, USA.

Fergerson, R.W., Knublauch, H., Noy, N.F. and Musen, M.A 2004. The protégé OWL Plugin: An Open Development Environment for Semantic Web Applications. In Mellraith, S. Pleousakis, D. Van Harmelen, F. eds.: Proceeding 3rd International Semantic Web Conference (ISWC 2004), Berlin Springer, Germany. 229-243

Fernández-López, M, Gómez-Pérez, A, Pazos, A, and Pazos, J. 1999. Building a Chemical Ontology Using Methontology and the Ontology Design Environment. IEEE Intelligent Systems & their applications. January/February 1999. 4.1: 37-46.

Ferrer-i-Cancho, R 2005. The structure of syntactic dependency networks: insights from recent advances in network theory. In: L. V., A. G. (eds.) Problems of quantitative linguistics, 60–75.

Fisher, D. H. 1987. Knowledge acquisition via incremental conceptual clustering. Machine Learning 2.2: 139–172.

Fortuna, B., Grobelnik, M., and Mladenic, D. 2006. Semi-automatic Data-driven Ontology Construction System. Proc.Of the 9th Int. multi-conference Information Society Springer. 309-318.

Frakes, W.B. and Baeza-Yates 1992. Stemming algorithms. Information Retrieval: Data Structures and Algorithms, eds. W.B. Frakes & R. Baeza-Yates, Prentice Hall: EnglewoodCliffs, US, 131–160.

Frederik, C. S. and Roos, N. 2014. Word-sense disambiguation for ontology mapping: concept Disambiguation using virtual documents and information Retrieval techniques. Springer-Verlag Berlin Heidelberg. J Data Semant, DOI 10.1007/s13740-014-0045-5, online September, 2014.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. Proceedings of the 20th International joint conference on artificial intelligence, 6–12.

Gale, WA, Church, K. W, and Yarowsky, D. 1992. A method for disambiguating word senses in a large corpus. Comput Humanit 26.5/6: 415–439.

George, T. and Vicky P.2009. A Generalized Vector Space Model for Text Retrieval Based on Semantic Relatedness.

Gil, Y., Motta, E., Benjamin V. R. and Musen M.ed. 2005. Proceeding of the 4th International Semantic Web Conference (ISWC) 2005.Proceedings of the 4th International Conference, LNCS 3729, Springer-Verlag. 548-562.

Gomaa, W H. and Fahmy, A. A. 2013. A survey of text similarity approaches. International Journal of Computer Applications (0975 – 8887) April 68.13: 13-18.

Gomez-Perez, A. and Manzano-Macho, D. 2003. OntoWeb Deliverable 1.5: A Survey of Ontology Learning Methods and Techniques. Universidad Politecnica de Madrid.18.4: 293-316.

Gonzalo, J., Verdejo, F., Chugur, I., Cigarrán, J. 1998. Indexing with WordNet synsets can improve text retrieval, in *Proc. the COLING/ACL '98 Workshop on Usage of WordNet for Natural Language Processing*,

Gruber, T.R. 1993. A translation approach to porTable ontologies. Knowledge Acquisition, 5.2: 199-220.

Guarino, N., Masolo, C., and Vetere, G. 1999. OntoSeek : content-based access to the web*". IEEE Intelligent Systems*, 14:70-80.

Guha, R., Dill, S., Eiron, N., Gibson, D., Gruhl, D., and Jhingran, A. 2003. A Case for automated large scale semantic annotation. (Elsevier, Ed.) Journal of Web Semantics 1.1: 115-132.

Hammouda, K.M. and Kamel, M.S. 2004. Efficient phrase-based document indexing for web document clustering, IEEE Transaction on Knowledge and Data Engineering, 16.10: 1279-1296.

Hartigan, J. and Wong, M. 1979. A k-means clustering algorithm. Applied Statistics, 28.1: 100–108.

Hazman, M., El-Beltagy, S. R., and Rafea, A. 2009. Ontology learning from domain specific web documents. In International Journal of Metadata, Semantics and Ontologies, 4.1-2: 24 – 33.

Hearst, M.B.1992. Automatic acquisition of hyponyms from large text corpora. In Proceeding of the 14th interval conference on computation linguistic Nanted, August France. 539-545

Hirst, G. and Budanitsky, A. 2006. Evaluating WordNet-based measures of semantic distance," Comput. Linguist., 32.1: 13-47.

Hoffman, H., Arnoldi, C., and Chuang, I. 2005. The Clinical Bioinformatics Ontology: A Curated Semantic Network Utilizing Refseq Information. In Proceedings of Pacific Symposium on Biocomputing 139-150.

Horrocks, I. and. Patel-Schneider, P. F. 2004. Reducing OWL Entailment to Description Logic Satisfiability.Journal of Web Semantics, 1-4.

Hotho, A., Berendt, B., and Stumme, G. 2002. Towards semantic web mining. In Proceedings of International Semantic Web Conference ISWC, 264– 278.

Hu, J., Fang. L. Cao, Y., Zeng, H. J., Li, H., and Yang, Q., 2008. Enhancing text clustering by leveraging Wikipedia semantics. In Proceeding of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval New York. 179-186.

Islam, A. and Inkpen, D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. ACM Trans. Knowl. Discov. Data July, 2.2: 1–25.

ISO. (2009). ISO-704 Terminology work: Principles and methods. 3rd ed. Geneva, International Organization for Standardization Switzerland:. 65:

Jain, A.K., Murty, M.N., Flynn, P.J. (1999): Data Clustering: A Review. ACM Computing Survey. 31: 264-323.

Jaiswa, l. P., Avraham, S., Tung, C., Ilic, K., Kellogg, E., McCouch, S., Pujar, A., Reiser-Seung, L, Rhee Y., Sachs M., Schaeffer, M., Stein L,, Stevens P., Vincen L., Zapata F., and Ware D. 2005. Plant Ontology (PO): a controlled vocabulary of plant structures and growth stages, Comparative and Functional Genomics. 388–397.

Jiang, J. J. and Conrath, D. W. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. Proceedings of International Conference on Research in Computational Linguistics, Taipei, Taiwan, August 22-24.

Jian-liang, X., Hong-yan, Y. W. and Jing, X. 2009. Development of domain ontology for E-learning course. International symposium of IT.on Medicine and Education (ITIME) 2009 IEEE. 1: 501-506.

Jing, F, Zhang, X and Tian-yang, D. 2008. Research of Plant Domain Knowledge Model Based on Ontology, In Proceedings of IEEE 3rd International Conference on Innovative Computing Information and Control. June, 108-112.

Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features. ECML 137-142.

Junfang, S. and Li, L. 2010. Web information extraction based on news domain ontology Theory. In Proceedings of IEEE 2nd Symposium on Web Society, 416-419.

Kalyanpur, A., Parsia, B., Sirin, E., Grau, B.C., and Hendler, J. 2005. Swoop: A Web Ontology Editing Browser", from: http://www.mindswap.org/2004/SWOOP/.Mind, July, 1-20.

Kamolvil, A. N. 2002. Property-based feature engineering and selection. Master's thesis, Department of Computer Sciences, University of Texas.

Kapoor, B. and Sharma, S. 2010. A comparative study ontology building tools for semantic web applications, International Journal July 1: 1-13.

Karoui, L., Aufaure, M., and Bennacer, N. 2004. Ontology Discovery from Web Pages: Application to Tourism. In ECML/PKDD 2004: Knowledge Discovery and Ontologies KDO-2004, September, 115-120.

Kaza, S. and Chen, H. 2008. Evaluating ontology mapping techniques: An experiment in public safety information sharing, Journal of Decision Support System., doi:10.1016/j.dss.2007.12.007. 45.4:714-728.

Kessler, B., Nunberg, G., and Schutze, H. 1997. Automatic detection of text genre. In Proceedings of the 30th ACL and EACL, 32–38.

Khan, L., and Luo, F. 2002. Ontology Construction for Information Selection In Proc. of 14th IEEE *International Conference on Tools with Artificial Intelligence*,. Washington DC, November 122-127.

Khan, S. and Marvon, F. 2006. Identifying relevant sources in query reformulation. In: Proceedings of the 8th International Conference on Information Integration and Web-based Applications & Services (IIWAS). Yogyakarta, Indonesia, 99–130.

Kifer, M., Lausen, G. and Wu, J. 1995. Logical foundations of object-oriented and frame-based languages. Journal of the ACM, 42.4: 741-843.

Kim, M. C. and Choi, K. S. 1999. A Comparison of Collocation-based Similarity Measures in Query Expansion. Information Processing and Management: an International Journal, 35.1: 9-30.

Klein, M. 2001. Combining and relating ontologies: an analysis of problems and solutions. In A. Gomez-Perez, Gruninger M., Stuckenschmid H. , and Uschold M., editors, Workshop on Ontologies and Information Sharing, IJCAI01, Seattle, USA, 4-5.

Knappe, R., Bulsko H. and Andreasen T. 2002. On Measuring Similarity for Conceptual Querying. Proceedings of the 5th International Conference on Flexible Query Answering Systems,October Copenhagen, Denmark, 27-29.

Kwantes, P. J. 2005. Using context to build semantics. Psychological Bulletin and Review, 12: 703-710.

Lancaster, F. W., and Warner, A. J. 1993, Information retrieval today. Arlingto, UA.: Information Resource Press.

Lancaster, F. W., Elliker, C. and Colonell, T. H. 1968. Subject analysis', Annual review of information science and technology , 24: 34-84.

Landauer, T. K. and Dumais, S. T. 1997. A solution to plato's problem: The latent semantic analysis theory of the aquisition, induction, and representation of knowledge. Psychological Review, 104, 211-240.

Larsen, B. and Aone, C.1999. Fast and effective text mining using linear-time document clustering. In Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining, (SIGKDD) ACM.

Lassila, O and Swick, R.1999. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation.http://www.w3.org/TR/REC-rdf-syntax/.

Leacock, C. and Chodorow, M. , 1998. Combining local context and WordNet similarity for word sense identification. In C. Fellbaum Ed., MIT Press 265–283.

Lee, C.Y. and Soo, V.W. 2005. Ontology-based information retrieval and extraction. In: Proceedings of 3rd International Conference on Information Technology: Research and Education (ITRE). IEEE, 265–269.

Lenat, D. B. 1995. CYC: A large-scale investment in knowledge infrastructure. Communications of the ACM 38.11: 33–38.

Leopold, E. and Kindermann, J. 2002. Text categorization with support vector machines. how to represent texts in input space? , Machine Learning 46: 423 - 444.

Lewis, D. D., and Ringuette, M. 1994. Comparison of two learning algorithms for text categorization. In Proceedings of the 3rd Annual Symposium on document analysis and information retrieval (SDAIR'94), 81–93.

Li, Y., Bandar, Z. A. and McLean, D. 2003. An approach for measuring semantic similarity between words using multiple information sources, IEEE Transactions on Knowledge and Data Engineering, July-August, 15. 4: 871 – 882.

Lin, D. 1998. An information-theoretic definition of similarity, Proceedings of the 15th International Conference on Machine Learning Madison, Wisconsin, USA July 24-27.

Lioma, C. and Ounis, I. 2008. A syntactically-based query reformulation technique for information retrieval", Information Processing and Management, 44.1: 143–162.

Madsen, R. E., Sigurdsson, S., Hansen, L. K. and Lansen, J. 2004. Pruning the Vocabulary for Better Context Recognition, 17th International Conference on Pattern Recognition, August, 2:127-138.

Maedche, A. and Staab, S. 2000. Ontology Learning for the Semantic Web.In IEEE Intelligent Systems, Special Issue on the Semantic Web, 16.2.

Maedche, A. and Volz, R. 2001. The ontology extraction maintenance framework Text-To-Onto. In Proceeding of the workshop on integrating data mining and knowledge management. 16:72-79.

Mahesh, K., Kud J. and Dixon, P 1999. Oracle at TREC8: A Lexical Approach, In proceeding of the 8th Text Retrieval Conference (TREC-8), NIST special publication 500.

Maki, W. S., McKinley, L. N. and Thompson, A. G. 2004. Semantic distance norms computed from an electronic dictionary (WordNet). Behavior research methods, instruments, and computers, 36: 421-431.

Manola, F. and Miller, E. (eds.) 2004. RDF (Resource Description Framework) Primer, W3C Recommendation, Boston ,:http://www.w3.org/TR/rdf-primer/

McCallum, A. and Nigam, K. 1998. A comparison of event models for naive Bayes text classification. Learning for text categorization, Technical Report WS-98-05 Menlo Park: The AAAI Press. 41–48.

Meng, L. Huang, R. Gu, J. 2014. Measuring Semantic Similarity of Word Pairs Using Path and Information Content. International Journal of Future Generation Communication and Networking 7.3:183-194

Mihalcea, R. 2006. Knowledge-based methods for word sense disambiguation, 107–131.

Mihalcea, R., Corley, C. and Strapparava, C. 2006. Corpus-based and knowledge based measures of text semantic similarity. In Proceedings of the 21st National Conference on Artificial Intelligence Menlo Park: AAAI Press, 775-780.

Miller, G. A. 1995. WordNet: a lexical database for English. Communications of the ACM, WordNet, http://wordNet.princeton.edu 38.11: 39–41.

Milne, D., Witten, I. H. and Nichols, D. M. 2007. A knowledge-based search engine powered by Wikipedia. In Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management New York, 445-454.

Montanes, E., Ferandez, J., Diaz, I., Combarro, E.F and Ranilla, J. 2003. Measures of rule quality for feature selection in text categorization, 5th international Symposium on Intelligent data analysis, Springer-Verlag Germeny. 28.10: 589-598.

Montoyo, A., Suárez, A., Rigau, G., and Palomar, M. 2005. Combining knowledge- and corpus-based word-sense-disambiguation methods. J Artif Intell Res 23.1: 299–330.

Navigli, R and Ponzetto, S 2010. Babelnet: Building a very large multilingual semantic network. In: Proceedings of the 48th annualmeeting of the association for computational linguistics, association for computational linguistics, 216–225.

Navigli, R. 2009. Word Sense Disambiguation: A Survey, ACM Comput. Surv., 41.2:1-69.

Navigli, R. and Velardi, R. 2004: Learning domain ontologies from document warehouses and dedicated web sites. Computational Linguistics 30.2:151-179.

Newman, M. E. J. and Girvan, M. 2004. Finding and evaluating community structure in networks. Phys. Rev. E., 69: 26-113.

Nirgude, V., Sharma, R. and Sedamkar, R. 2013. A Web Search Engine based approach to Measure the Semantic Similarity between Words using Page Count and Snippets Method (PCSM) International journal of advanced research in computer engineering & technology (IJARCET) ISSN: 2278 – 1323 July 2.7: 2252-2257.

Nitish, A., Kartik, A. and Paul, B. 2012. Pushing corpus based relatedness to similarity: shared task system description. First Joint Conference on Lexical and Computational Semantics (SEM), Montreal, Canada, Association for Computational Linguistics, June, 643–647.

Noy, N.F. and Musen, M.A. 2003. The PROMPT suite: Interactive tools for Ontology merging and mapping, Int. J. Human Comput. Stud, 59: 983–1024.

Oikonomakou, N. and Vazirgianni, M. 2005. A Review of Web Document Clustering Approaches, Data Mining and Knowledge Discovery Handbook, Springer US, 921-943.

Ontostudio. W. & Information, E. OntoStudio 3 Professional tool for knowledge architects, from:http://www.ontoprise.de/fileadmin/user_upload/Flyer_EN/Flyer_Onto Studio_en.pdf .

Osinski S.and Weiss D.. 2005. A Concept-Driven Algorithm for Clustering Search Results. 20.3:48– 54.

OWL Web Ontology Language, 2004.http://www.w3.org/TR/owl- features/

Ozcan, R. and Aslandogan, Y. A. 2005. Concept-based information access. In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05) IEEE Computer Society, Las Vegas, NV, 794–799.

Pan, J. Z., Taylor, S. and Thomas, E. 2009. Reducing ambiguity in tagging systems with folksonomy search expansion. In Proceedings *6th European Semantic Web Conference (ESWC),* 669-683.

Paralic, J. and Kostial, I. 2003. Ontology-based Information Retrieval, Information and Intelligent Systems, Croatia, 23-28.

Patwardhan, S., Banerjee, S. and Pedersen, T. 2003. Using measures of semantic relatedness for word sense disambiguation. Proceedings of 4th International Conference on Computational Linguistics and Intelligent Text Processing and Computational Linguistics, Mexico, February, 16-22.

Pease, A. and Niles, I. 2002. IEEE standard upper ontology: a progress report. Knowledge Engineering Review, Special Issue on Ontologies and Agents 17.1: 65–70.

Pedersen, T, 2006. Unsupervised corpus-based methods for word sense disambiguation. Agure E. and Edmonds P. (Ed.). Algorithm and application of text, speech and language technology, New York, USA. 33: 133-166.

Pedersen, T., Patwardhan, S. and Jason, M. 2004. WordNet:: Similarity - Measuring the Relatedness of Concepts. in the Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-04), San Jose, CA

Pedersen, T., Patwardhan, S., and Michelizz,i J. 2004. WordNet Similarity: Measuring the relatedness of concepts. In Proceedings of the 19th National Conference on Artificial Intelligence, New York, USA. 1024–1025.

Perich, F., Finin, T., and Joshi, A. 2004. SOUPA: Standard ontology for ubiquitous and pervasive applications, 2004. In Proceedings of the 1st Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 258 – 267.

Peter, K. 2009. Experiments on the difference between semantic similarity and relatedness. In Proceedings of the 17th Nordic Conference on Computational Linguistics - NODALIDA '09, Denmark, 81-88.

Pirrò, G. 2009. A semantic similarity metric combining features and intrinsic information content. Data and Knowledge Engineering, doi.10.1016j.datak. 68.11: 1289-1308.

Poole, J., and Campbell, J.A. 1995. A novel algorithm for matching conceptual and related graphs. In: Proceedings of the 3rd International Conference on Conceptual Structures. Applications, Implementation and Theory, Lecture Notes in Computer Science. Springer-Verlag, 954: 293–307.

Prathv, K. and Ravishankar, K. 2013. Measuring Semantic Similarity between Words using Page-Count and Pattern Clustering Methods" International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, 3.2: 31-34.

Protégé ontology editor developed by Stanford Medical Informatics, Stanford University School of Medicine, further information available, from: http://protege.stanford.edu/.

Qiu, Y. and Frei, H. P. 1993. Concept based query expansion. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, ACM Press, 160-169.

Quillian, M. R. 1968. Semantic Memory. in Marvin Minsky (ed.) *Semantic Information Processing* (Cambridge, MA: MIT Press): 227-270.

Resnik, P. 1995. Using information content to evaluate semantic similarity, Proceedings of the 14th International Joint Conference on Artificial Intelligence, Montréal Québec, Canada, August, 20-25.

Rinaldi, A.M. 2009. An ontology-driven approach for semantic information retrieval on the web. ACM Transactions on Internet Technology (TOIT) 9.3: 10-14.

Roberston S. 2004. Understanding inverse document frequency: on theoretical argument for IDF, Journal of Documentation, 60.5: 503-520.

Rodriguez M. A. and Egenhofer M. J. 2003. Determining Semantic Similarity among Entity Classes from Different Ontologies, IEEE Trans. on Knowledge and Data Engineering, 15.2:

Roush, W. 2004. Search beyond Google. Technology Review. http://www.technologyreview.com/articles/print_version/roush0304.asp., March 2004.

Rowley J. 1994. Aspects of a library systems methodology. Journal of information science, 20.1: 41-5.

Ruban, S. and Behin, S, S. 2015. An experimental analysis and implementation of ontology based query expansion, ARPN Journal of Engineering and Applied Sciences April 10.7: 3108-3111.

Saad, S., Salim N., Zainal, H. and Muda, Z. 2011. A process for building domain ontology: An experience in developing Solat ontology. In Proceedings of IEEE International Conference on Electrical Engineering and Informatics, 1-5.

Sabai, M. H. 2013. Semantic Information Retrieval based on Wikipedia Taxonomy International Journal of Computer Applications Technology and Research ISSN: 2319–8656 . 2.1: 77-80.

Sabou, M., Wroe, C., Goble, C., and Mishne, G. 2005. Learning Domain Ontologies for Web Service Descriptions: an Experiment in Bioinformatics. In Proceedings of the 14th International World Wide Web Conference (WWW2005), ACM Chiba, Japan. 190-198.

Sahami, M. and Heilman, T.D. 2006. A web-based kernel function for measuring the similarity of short text snippets. In Proceedings of the 15th World Wide Web Conference, ACM, 377-386.

Salton, G. and Buckley, C. 1988. Term-weighting approaches in automatic text retrieval," Inf. Process. Manage., 24.5: 513–523.

Salton, G. and McGill, M. 1983. Information Retrieval -Grundlegendes for Information swissenschaftler. New York, McGraw-Hill.xv+448, 17:4.

Salton, G. and Yang, C. S. 1973. On the Specification of Term Values in Automatic Indexing, Journal of Documentation, 29.4: 351-372.

Salvador, S. and Miguel-Angel, S. 2009. Using an AGROVOC-based ontology for the description of learning resources on organic agriculture, Metadata and Semantics, Springer, 481-492.

Sanchez, D. and Moreno, A. 2004. Creating ontologies from Web documents. In Recent Advances in Artificial Intelligence Research and Development.IOS Press, 113.1: 11-18.

Saruladha, K Aghila, G. and Sathiya, B. 2011. A comparative analysis of ontology and schema matching system. International Journal of Computer Applications 34.8: 14-21.

Schickel-Zuber, V. and Falting, B. 2007. OSS: A semantic similarity function based on hierarchical ontologies. In Proceeding of the 20th international joint conference on artificial intelligence, Morgan Kaufmann Publisher Inc. 551-556.

Schütze, H. and Pedersen, J.O. 1995. Information retrieval based on word senses. In: Proceedings of the 4th annual symposium on document analysis and information retrieval, 2.1:45-65.

Schwartz, C. 2001. Sorting out the web: Approaches to Subject Access, Westport, Ablex Publishing.

Seo, C. and Ozden. B. 2004. Ontology-based File Naming Through Hierarchical Conceptual Clustering. In Technical Report, University of Southern California.

Shabo, A. 2006. Revolutionary impact of XML on biomedical information interoperability. IBM Systems, January,45.2:361

Shamsfard, M. and Barforoush, A. A. 2003. The state of the art in ontology learning. A framework for comparison. The Knowledge Engineering Review. 18.4:293- 316.

Sieg, A., Mobasher, B., and Burke, R. 2007. Representing context in web search with ontological user profiles. In: Proceedings of the Sixth International Conference on Modeling and Using Context, Roskilde, Denmark. LNAI 4635, 439-452.

Smola, A. J. and Schölkopf, B. 2004. A tutorial on support vector regression. Statistics and Computing, 14.3:199-222.

Snoussi, L. M. and Nie J.Y. 2002. Towards an ontology-based web data extraction. The AI-2002Workshop on Business Agents and the Semantic Web (BASeWEB), 26-33.

Soni, A., Sunhare. H. and Patel S. 2013. Semantic retrieval technique based on domain ontology. International Journal of Innovative Research in Science, Engineering and Technology July 2.7: 3187-3192.

Sparck, J. K. and Willet P. 1997. Overall Information.In: Sparck Jones, K and Willet, P. (eds). Reading in Information Retrieval, Morgan Kaufmann Pub. Inc. San Francisco Calfonia, USA 47-60

Sparck, J. K., Walker, S and Robertson, S E. 2000. A probabilistic model of information retrieval: Development and comparative experiments. Parts 1 and 2, Information Processing and Management, 36.6: 779-808 and 809-840.

Sridevi, K. Umarani, R. and Selvi, V. 2011. An Analysis of Web Document Clustering Algorithms. International Journal of Science and Technology. December, 1.6: 275-282.

Stanislaw, O. and Dawid, W. 2005. A concept-driven algorithm for clustering search results. In: Intelligent Systems, IEEE , 20. 3: 48-54.

Steinbach, M., Karypis, G., and Kumar, V. 2000. A comparison of document clustering techniques. In Proceeding of the 6th ACM SIGKDD world text mining conference Boston USA. 22.8:885-905.

Sure, Y., Erdmann, M., Angele, J., Staab, S., Studer, R., and Wenke, D. 2002. OntoEdit: Collaborative ontology development for the semantic web. In International Semantic Web Conference 2002 (ISWC 2002), Sardinia, Italy. 221-235.

Talita, P., Yeo, A., and Kulathuramaiyer, N. 2010. Challenges in Building Domain Ontology for Minority Languages. In Proceedings of International Conference on Computer Applications and Industrial Electronics, 574 – 578.

Tomassen, S. L. and Strasunskas, D. 2009. Construction of Ontology Based Semantic-Linguistic Feature Vectors for Searching: The Process and Effect. In Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - IEEE Computer Society, Washington, 3: 133-138.

Tomassen, S.L. and Strasunskas, D. 2009. Relating ontology and Web terminologies by feature vectors: unsupervised construction and experimental validation. In: Kotsis, G., Taniar, D., Pardede, E. & Khalil, I. (eds.) Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services, ACM, 86-93.

Turney, P.D 2006. Similarity of semantic relations. Computing Linguistic, 32.3:379-416.

Tversky A. 1977. Features of Similarity, Psycological Review, 84.4:

Uhlir, J., Machkova, M., and TerezaCermanova, P. 2003. Creation of architectural ontology: user's experience. Proceedings. 14th International workshop on database and expert systems applications, Sept. 65 – 69.

Ullrich, M., Maier, A.and Angele, P. D. J 2003. Taxonomie, Thesaurus, Topic Map, Ontologie - einVergleich. Ontoprise Whitepaper Series. http://www.ullri.ch/download/Ontologien/ttto13.pdf.1.4:

Vallet, D., Fernandez, M. and Castells, P. 2005. An ontology-based information retrieval model. In The Semantic Web: Research and Applications, ESWC, 455–470.

van Hage, W. R. Katrenko, S. and Schreiber, G. 2005. A method to combine linguistic ontology-mapping techniques. In Proceeding 4th International Semantic Web Conference (ISWC), volume 3729 of Lecture notes in computer science, Galway (IE), 732–744.

Varelas, G., Voutsak,i E., Raftopoulou, P., Petrakis, E. G. and Milios, E. E. 2005. Semantic similarity methods in WordNet and their application to information retrieval on the web. Proceedings of the 7th annual ACM international workshop on Web information and data management, November, Bremen, Germany.

Visser, P.R.S., Jones, D.M., Bench-Capon, T.J.m., Shave, M.J.R. 1997. An analysis of ontology mismatches: heterogeneity versus interoperability, AAAI. Spring Symposium on Ontological Engineering, Stanford, USA.

Voorhees, E. M. 1994. Query expansion using lexical-semantic relations. In Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Springer, 61–69.

Waitelonis, J. and Sack, H. 2009. Augmenting video search with Linked Open Data. In Proceedings *I*-SEMANTICS, Graz, Austria, 550-558.

Wang, Y. and Wang, X. J 2005. A New Approach to feature selection in Text Classification, Proceedings of 4th International Conference on Machine Learning and Cybernetics, IEEE, 6: 3814-3819.

Wang, Z., Sun, X., Zhang, D. and Li, X. 2006. An optimal svm-based text classification algorithm fifth international conference on machine learning and cybernetics, Dalian, 13-16.

Wu, Z. and M. Palmer, M. 1994. Verb semantics and lexical selection, Proceedings of 32nd annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, June 27-30.

Xin, L. 1990. Document retrieval: A structural approach', Information Processing and Management, 26.2: 209-218.

Xu, J and Croft, W. B. 2000. Improving the effectiveness of information retrieval with local context analysis. ACM Trans. Inf. Syst., ISSN 1046-8188. doi:http://doi.acm.org/10.1145/333135.333138. 18.1:79–112.

Yang, C. and Wu, S. 2011. A WordNet based information retrieval on the semantic web. In: Networked Computing and Advanced Information Management (NCM), 7th International Conference IEEE. 324–328.

Yang, Y. 1999. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1.1/2: 67–88.

Yang, Y. and Pederson, J. O. 1997. A comparative study on feature selection in text categorization. ICML, 97. 412-42.

Zamir, O. and Etzioni, O. 1999. Document Clustering: A Feasibility Demonstration.Proceedings of the 19th International ACM SIGIRConference on Research and Development of Information Retrieval, 46-54.

Zhang, F., Srihari, R. K. Z. and Rao, A. B 2000. Intelligent indexing and semantic retrieval of multimodal documents, Information Retrieval, 2:245-275.

Zhang, G., Zhou, Z. T., and Wang, B. 2006. Research on topic based distributed information retrieval. International Journal of Technology Computer Engineering, 32.12: 80-82.

Zhao, Y. and Karypis, G. 2005. Hierarchical clustering algorithms for document datasets. Data Mining and Knowledge Discovery, 10:141–168.

Zhou, Z., Wang, Y. and Gu, J. 2008. New model of semantic similarity measuring in WordNet, Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering, Xiamen, China, 17-19.

Zhu, D. and Heinz, D. 2008. Improving web search by categorisation, clustering and personalization, Tang, Ch. and Ling, Ch.X. and Zhou, X. and Cercone, N.J. and Li, X. (ed), 4th International conference on advanced data mining and applications, Chengdu, China: Springer October, 659-666.

Zouaq, A., Gasevic, D. and Hatala, M. 2011. Ontologising concept maps using graph theory. In Proceedings of the ACM 26th Symposium On Applied Computing, Semantic Web and Applications, 1687-1692.

Appendix I: The Mean Values of MDSs on Similarity Measures

S/N	Concept1	Concept2	Cosine	CW-Cosine	RBF	CW-RBT	Euclidean	CW-Euclidean
1	HumanActivity	ExperimentActivity	0.022	0.246	0.976	0.981	0.016	0.198
2	Research	ExperimentActivity	0.014	0.025	0.971	0.971	0.01	0.018
3	Analysis	ExperimentActivity	0.035	0.371	0.974	0.983	0.025	0.308
4	Investigation	ExperimentActivity	0.028	0.338	0.978	0.985	0.021	0.273
5	Monitoring	ExperimentActivity	0.014	0.016	0.976	0.976	0.01	0.012
6	Project	ExperimentActivity	0.033	0.133	0.971	0.974	0.024	0.1
7	ProofOfConcept	ExperimentActivity	0.017	0.039	0.974	0.975	0.013	0.028
8	ResearchSetting	ExperimentActivity	0.018	0.091	0.974	0.976	0.117	0.175
9	Residual	ExperimentActivity	0.032	0.165	0.975	0.978	0.023	0.134
10	Result	ExperimentActivity	0.004	0.006	0.977	0.977	0.003	0.005
11	Representation	ExperimentActivity	0.015	0.028	0.973	0.973	0.011	0.02
12	Variable	ExperimentActivity	0.013	0.032	0.972	0.975	0.01	0.024
13	Assesment	ExperimentActivity	0.03	0.067	0.973	0.974	0.021	0.05
14	Evidence	ExperimentActivity	0.018	0.026	0.976	0.976	0.014	0.02
15	Experiment	ExperimentActivity	0.043	0.239	0.97	0.976	0.031	0.191
16	ExperimentActivity	ExperimentActivity	0.045	0.527	0.975	0.988	0.032	0.445
17	Campaign	ExperimentActivity	0.04	0.659	0.975	0.991	0.029	0.601
18	Correction	ExperimentActivity	0.042	0.284	0.975	0.982	0.031	0.228
19	Difference	ExperimentActivity	0.052	0.345	0.973	0.982	0.041	0.277
20	Hypothesis	ExperimentActivity	0.064	0.29	0.976	0.982	0.047	0.233
21	Publication	ExperimentActivity	0.016	0.041	0.974	0.975	0.012	0.03
22	Realization	ExperimentActivity	0.051	0.485	0.975	0.987	0.037	0.413
23	Sample	ExperimentActivity	0.067	0.375	0.972	0.981	0.049	0.309
24	Validation	ExperimentActivity	0.033	0.263	0.978	0.984	0.024	0.208
25	Proof	ExperimentActivity	0.026	0.202	0.981	0.985	0.019	0.101
26	Observation	ExperimentActivity	0.021	0.056	0.974	0.975	0.015	0.041
27	HumanActivity	Methodology	0.009	0.04	0.975	0.976	0.007	0.031
28	Research	Methodology	0.015	0.042	0.969	0.97	0.011	0.032
29	Analysis	Methodology	0.007	0.012	0.973	0.974	0.005	0.008
30	Investigation	Methodology	0.028	0.127	0.977	0.979	0.02	0.098
31	Monitoring	Methodology	0.008	0.009	0.975	0.975	0.006	0.006
32	Project	Methodology	0.004	0.004	0.971	0.971	0.003	0.003
33	ProofOfConcept	Methodology	0.005	0.009	0.974	0.974	0.004	0.007
34	ResearchSetting	Methodology	0.011	0.053	0.973	0.975	0.111	0.144
35	Residual	Methodology	0.008	0.01	0.974	0.974	0.006	0.006
36	Result	Methodology	0.016	0.019	0.975	0.976	0.011	0.014
37	Representation	Methodology	0.013	0.014	0.971	0.971	0.01	0.01
38	Variable	Methodology	0.013	0.093	0.971	0.973	0.01	0.073
39	Assesment	Methodology	0.013	0.013	0.973	0.973	0.009	0.009
40	Evidence	Methodology	0.005	0.007	0.976	0.976	0.004	0.005
41	Experiment	Methodology	0.021	0.123	0.969	0.973	0.015	0.096
42	ExperimentActivity	Methodology	0.029	0.1	0.974	0.976	0.022	0.077
43	Campaign	Methodology	0.01	0.013	0.973	0.973	0.007	0.01
44	Correction	Methodology	0.035	0.137	0.974	0.977	0.026	0.104
45	Difference	Methodology	0.009	0.008	0.973	0.973	0.006	0.006
46	Hypothesis	Methodology	0.068	0.289	0.975	0.981	0.05	0.239
47	Publication	Methodology	0.023	0.066	0.973	0.974	0.016	0.05
48	Realization	Methodology	0.029	0.103	0.974	0.976	0.02	0.074
49	Sample	Methodology	0.047	0.122	0.97	0.972	0.034	0.087
50	Validation	Methodology	0.013	0.089	0.978	0.98	0.01	0.068

51	Proof	Methodology	0.029	0.082	0.98	0.981	0.021	0.064
52	Observation	Methodology	0.025	0.122	0.973	0.975	0.018	0.094
53	HumanActivity	ResearchSci	0.038	0.326	0.975	0.983	0.025	0.262
54	ResearchHuman	ResearchSci	0.032	0.454	0.972	0.984	0.023	0.373
55	Analysis	ResearchSci	0.033	0.269	0.974	0.981	0.024	0.21
56	Investigation	ResearchSci	0.04	0.507	0.978	0.989	0.029	0.429
57	Monitoring	ResearchSci	0.032	0.472	0.976	0.987	0.023	0.393
58	Project	ResearchSci	0.029	0.498	0.973	0.986	0.021	0.417
59	ProofOfConcept	ResearchSci	0.023	0.094	0.974	0.976	0.017	0.073
60	ResearchSetting	ResearchSci	0.057	0.391	0.974	0.985	0.145	0.436
61	Residual	ResearchSci	0.027	0.288	0.975	0.982	0.02	0.229
62	Result	ResearchSci	0.036	0.528	0.977	0.989	0.026	0.449
63	Representation	ResearchSci	0.044	0.484	0.972	0.985	0.032	0.404
64	Variable	ResearchSci	0.035	0.266	0.972	0.979	0.025	0.27
65	Assesment	ResearchSci	0.029	0.371	0.975	0.984	0.021	0.309
66	Evidence	ResearchSci	0.034	0.352	0.977	0.984	0.024	0.291
67	Experiment	ResearchSci	0.067	0.552	0.971	0.986	0.048	0.477
68	ExperimentActivity	ResearchSci	0.057	0.53	0.975	0.988	0.041	0.449
69	Campaign	ResearchSci	0.028	0.186	0.975	0.979	0.021	0.145
70	Correction	ResearchSci	0.035	0.221	0.975	0.98	0.025	0.164
71	Difference	ResearchSci	0.038	0.34	0.974	0.982	0.028	0.279
72	Hypothesis	ResearchSci	0.069	0.38	0.977	0.985	0.338	0.43
73	Publication	ResearchSci	0.049	0.488	0.974	0.986	0.036	0.409
74	Realization	ResearchSci	0.035	0.075	0.975	0.976	0.026	0.058
75	Sample	ResearchSci	0.07	0.482	0.972	0.984	0.051	0.402
76	Validation	ResearchSci	0.042	0.375	0.978	0.986	0.03	0.307
77	Proof	ResearchSci	0.063	0.354	0.98	0.987	0.046	0.28
78	Observation	ResearchSci	0.045	0.798	0.974	0.979	0.033	0.217
79	HumanActivity	RetrievalAproach	0.024	0.105	0.979	0.981	0.018	0.079
80	Research	RetrievalAproach	0.021	0.129	0.976	0.979	0.078	0.097
81	Analysis	RetrievalAproach	0.017	0.085	0.979	0.98	0.012	0.065
82	Investigation	RetrievalAproach	0.02	0.17	0.981	0.984	0.014	0.131
83	Monitoring	RetrievalAproach	0.021	0.079	0.98	0.981	0.015	0.064
84	Project	RetrievalAproach	0.03	0.096	0.977	0.978	0.022	0.072
85	ProofOfConcept	RetrievalAproach	0.009	0.024	0.979	0.979	0.006	0.017
86	ResearchSetting	RetrievalAproach	0.024	0.07	0.979	0.98	0.121	0.156
87	Residual	RetrievalAproach	0.022	0.057	0.979	0.98	0.016	0.042
88	Result	RetrievalAproach	0.013	0.078	0.981	0.982	0.01	0.059
89	Representation	RetrievalAproach	0.038	0.095	0.977	0.978	0.039	0.072
90	Variable	RetrievalAproach	0.03	0.03	0.977	0.978	0.021	0.05
91	Assesment	RetrievalAproach	0.014	0.048	0.979	0.979	0.01	0.036
92	Evidence	RetrievalAproach	0.014	0.054	0.98	0.981	0.01	0.037
93	Experiment	RetrievalAproach	0.041	0.123	0.976	0.978	0.03	0.088
94	ExperimentActivity	RetrievalAproach	0.053	0.263	0.979	0.984	0.038	0.208
95	Campaign	RetrievalAproach	0.022	0.153	0.979	0.982	0.016	0.117
96	Correction	RetrievalAproach	0.012	0.059	0.98	0.981	0.009	0.045
97	Difference	RetrievalAproach	0.028	0.135	0.978	0.981	0.02	0.104
98	Hypothesis	RetrievalAproach	0.034	0.211	0.98	0.984	0.025	0.165
99	Publication	RetrievalAproach	0.033	0.164	0.978	0.981	0.042	0.125
100	Realization	RetrievalAproach	0.025	0.105	0.979	0.981	0.018	0.079
101	Sample	RetrievalAproach	0.045	0.213	0.977	0.981	0.028	0.167
102	Validation	RetrievalAproach	0.019	0.07	0.981	0.982	0.014	0.014
103	Proof	RetrievalAproach	0.032	0.141	0.983	0.986	0.023	0.117
104	Observation	RetrievalAproach	0.049	0.129	0.978	0.979	0.036	0.097

105	HumanActivity	CarbonDating	0.015	0.022	0.974	0.974	0.011	0.016
106	ResearcHuman	CarbonDating	0.005	0.004	0.969	0.969	0.004	0.003
107	Analysis	CarbonDating	0.012	0.012	0.973	0.973	0.009	0.009
108	Investigation	CarbonDating	0.048	0.047	0.975	0.975	0.035	0.035
109	Monitoring	CarbonDating	0.009	0.014	0.974	0.975	0.007	0.011
110	Project	CarbonDating	0.005	0.004	0.97	0.97	0.004	0.003
111	ProofOfConcept	CarbonDating	0.008	0.011	0.973	0.973	0.006	0.008
112	ResearchSetting	CarbonDating	0.007	0.022	0.973	0.973	0.108	0.119
113	Residual	CarbonDating	0.004	0.006	0.974	0.974	0.003	0.004
114	Result	CarbonDating	0.02	0.028	0.975	0.976	0.015	0.019
115	Representation	CarbonDating	0.009	0.009	0.971	0.971	0.007	0.007
116	Variable	CarbonDating	0.016	0.071	0.97	0.972	0.012	0.054
117	Assesment	CarbonDating	0.004	0.004	0.973	0.973	0.003	0.003
118	Evidence	CarbonDating	0.012	0.011	0.975	0.975	0.007	0.053
119	Experiment	CarbonDating	0.016	0.052	0.969	0.97	0.012	0.053
120	ExperimentActivity	CarbonDating	0.01	0.01	0.974	0.974	0.007	0.012
121	Campaign	CarbonDating	0.008	0.01	0.973	0.973	0.006	0.007
122	Correction	CarbonDating	0.012	0.021	0.974	0.975	0.009	0.015
123	Difference	CarbonDating	0.007	0.021	0.972	0.972	0.005	0.015
124	Hypothesis	CarbonDating	0.022	0.134	0.975	0.978	0.016	0.102
125	Publication	CarbonDating	0.013	0.012	0.972	0.972	0.008	0.009
126	Realization	CarbonDating	0.015	0.061	0.974	0.975	0.011	0.045
127	Sample	CarbonDating	0.021	0.044	0.97	0.971	0.016	0.041
128	Validation	CarbonDating	0.018	0.068	0.977	0.977	0.013	0.051
129	Proof	CarbonDating	0.017	0.016	0.98	0.98	0.012	0.012
130	Observation	CarbonDating	0.022	0.072	0.972	0.974	0.016	0.054
131	HumanActivity	IsotopeAnalysis	0.015	0.087	0.979	0.98	0.011	0.066
132	Research	IsotopeAnalysis	0.015	0.082	0.976	0.977	0.011	0.062
133	Analysis	IsotopeAnalysis	0.03	0.294	0.978	0.984	0.022	0.233
134	Investigation	IsotopeAnalysis	0.021	0.139	0.98	0.983	0.019	0.107
135	Monitoring	IsotopeAnalysis	0.014	0.066	0.979	0.98	0.01	0.049
136	Project	IsotopeAnalysis	0.015	0.053	0.976	0.977	0.011	0.04
137	ProofOfConcept	IsotopeAnalysis	0.005	0.017	0.979	0.979	0.004	0.012
138	ResearchSetting	IsotopeAnalysis	0.009	0.082	0.979	0.981	0.11	0.165
139	Residual	IsotopeAnalysis	0.003	0.005	0.979	0.979	0.002	0.003
140	Result	IsotopeAnalysis	0.009	0.033	0.98	0.981	0.007	0.025
141	Representation	IsotopeAnalysis	0.014	0.064	0.977	0.978	0.01	0.049
142	Variable	IsotopeAnalysis	0.018	0.018	0.977	0.978	0.013	0.059
143	Assesment	IsotopeAnalysis	0.003	0.005	0.978	0.978	0.002	0.004
144	Evidence	IsotopeAnalysis	0.003	0.005	0.98	0.98	0.003	0.004
145	Experiment	IsotopeAnalysis	0.006	0.019	0.976	0.976	0.005	0.014
146	ExperimentActivity	IsotopeAnalysis	0.02	0.081	0.978	0.98	0.014	0.06
147	Campaign	IsotopeAnalysis	0.009	0.019	0.978	0.978	0.006	0.014
148	Correction	IsotopeAnalysis	0.025	0.062	0.979	0.98	0.018	0.047
149	Difference	IsotopeAnalysis	0.01	0.044	0.977	0.978	0.016	0.007
150	Hypothesis	IsotopeAnalysis	0.027	0.147	0.98	0.982	0.02	0.113
151	Publication	IsotopeAnalysis	0.018	0.075	0.978	0.979	0.013	0.056
152	Realization	IsotopeAnalysis	0.013	0.025	0.979	0.979	0.01	0.019
153	Sample	IsotopeAnalysis	0.017	0.047	0.976	0.977	0.012	0.034
154	Validation	IsotopeAnalysis	0.012	0.036	0.981	0.981	0.009	0.027
155	Proof	IsotopeAnalysis	0.025	0.133	0.983	0.985	0.018	0.101
156	Observation	IsotopeAnalysis	0.013	0.024	0.978	0.978	0.009	0.017
157	HumanActivity	Photometry	0.01	0.014	0.975	0.975	0.007	0.01
158	Research	Photometry	0.015	0.063	0.971	0.972	0.011	0.047

159	Analysis	Photometry	0.011	0.024	0.975	0.975	0.008	0.018
160	Investigation	Photometry	0.016	0.03	0.978	0.978	0.011	0.022
161	Monitoring	Photometry	0.006	0.014	0.976	0.976	0.004	0.011
162	Project	Photometry	0.02	0.046	0.972	0.973	0.014	0.034
163	ProofOfConcept	Photometry	0.007	0.028	0.975	0.976	0.005	0.021
164	ResearchSetting	Photometry	0.008	0.013	0.975	0.975	0.11	0.113
165	Residual	Photometry	0.011	0.026	0.975	0.976	0.008	0.02
166	Result	Photometry	0.014	0.026	0.977	0.977	0.01	0.019
167	Representation	Photometry	0.009	0.025	0.973	0.974	0.006	0.019
168	Variable	Photometry	0.009	0.008	0.973	0.973	0.007	0.006
169	Assesment	Photometry	0.008	0.089	0.975	0.977	0.006	0.07
170	Evidence	Photometry	0.012	0.017	0.976	0.977	0.008	0.012
171	Experiment	Photometry	0.015	0.021	0.971	0.971	0.011	0.015
172	ExperimentActivity	Photometry	0.018	0.026	0.975	0.975	0.013	0.019
173	Campaign	Photometry	0.008	0.012	0.975	0.975	0.006	0.009
174	Correction	Photometry	0.013	0.035	0.976	0.976	0.01	0.026
175	Difference	Photometry	0.02	0.054	0.973	0.974	0.015	0.04
176	Hypothesis	Photometry	0.025	0.1	0.976	0.978	0.018	0.076
177	Publication	Photometry	0.015	0.028	0.974	0.975	0.011	0.021
178	Realization	Photometry	0.004	0.006	0.976	0.976	0.003	0.004
179	Sample	Photometry	0.022	0.022	0.972	0.972	0.016	0.026
180	Validation	Photometry	0.008	0.013	0.979	0.979	0.006	0.01
181	Proof	Photometry	0.015	0.032	0.981	0.981	0.011	0.024
182	Observation	Photometry	0.027	0.133	0.974	0.977	0.02	0.103
183	HumanActivity	RadioactiveDating	0.004	0.004	0.975	0.975	0.003	0.003
184	Research	RadioactiveDating	0	0	0.971	0.971	0	0
185	Analysis	RadioactiveDating	0.008	0.007	0.974	0.974	0.006	0.005
186	Investigation	RadioactiveDating	0.014	0.014	0.977	0.977	0.01	0.01
187	Monitoring	RadioactiveDating	0	0	0.976	0.976	0	0
188	Project	RadioactiveDating	0.004	0.003	0.972	0.972	0.003	0.002
189	ProofOfConcept	RadioactiveDating	0.011	0.014	0.974	0.975	0.008	0.01
190	ResearchSetting	RadioactiveDating	0.004	0.003	0.974	0.974	0.106	0.106
191	Residual	RadioactiveDating	0.011	0.01	0.974	0.974	0.008	0.007
192	Result	RadioactiveDating	0.014	0.019	0.977	0.977	0.01	0.014
193	Representation	RadioactiveDating	0.005	0.005	0.973	0.973	0.004	0.004
194	Variable	RadioactiveDating	0.012	0.036	0.972	0.973	0.009	0.027
195	Assesment	RadioactiveDating	0.006	0.016	0.974	0.975	0.005	0.012
196	Evidence	RadioactiveDating	0.004	0.005	0.976	0.977	0.003	0.004
197	Experiment	RadioactiveDating	0.012	0.036	0.971	0.972	0.009	0.027
198	ExperimentActivity	RadioactiveDating	0.008	0.005	0.975	0.975	0.006	0.004
199	Campaign	RadioactiveDating	0.004	0.005	0.975	0.975	0.003	0.003
200	Correction	RadioactiveDating	0.009	0.011	0.976	0.976	0.007	0.008
201	Difference	RadioactiveDating	0.011	0.02	0.973	0.974	0.008	0.015
202	Hypothesis	RadioactiveDating	0.019	0.07	0.976	0.978	0.014	0.052
203	Publication	RadioactiveDating	0.013	0.012	0.974	0.974	0.009	0.009
204	Realization	RadioactiveDating	0.01	0.029	0.975	0.976	0.007	0.021
205	Sample	RadioactiveDating	0.015	0.028	0.972	0.972	0.011	0.021
206	Validation	RadioactiveDating	0.014	0.036	0.978	0.979	0.01	0.026
207	Proof	RadioactiveDating	0.008	0.009	0.981	0.981	0.006	0.007
208	Observation	RadioactiveDating	0.026	0.054	0.973	0.974	0.019	0.04
209	HumanActivity	Spectroscopy	0.012	0.026	0.975	0.975	0.009	0.019
210	Research	Spectroscopy	0.005	0.004	0.971	0.971	0.004	0.004
211	Analysis	Spectroscopy	0.016	0.034	0.975	0.975	0.011	0.025
212	Investigation	Spectroscopy	0.023	0.068	0.978	0.979	0.016	0.051

213	Monitoring	Spectroscopy	0.004	0.004	0.976	0.976	0.003	0.003
214	Project	Spectroscopy	0.01	0.032	0.972	0.973	0.007	0.024
215	ProofOfConcept	Spectroscopy	0.01	0.028	0.975	0.975	0.007	0.021
216	ResearchSetting	Spectroscopy	0	0	0.975	0.974	-0.522	0.104
217	Residual	Spectroscopy	0.006	0.016	0.975	0.975	0.004	0.012
218	Result	Spectroscopy	0.028	0.107	0.977	0.979	0.02	0.082
219	Representation	Spectroscopy	0.007	0.019	0.973	0.973	0.005	0.014
220	Variable	Spectroscopy	0.013	0.026	0.972	0.973	0.009	0.019
221	Assesment	Spectroscopy	0.008	0.013	0.974	0.974	0.006	0.01
222	Evidence	Spectroscopy	0.012	0.034	0.976	0.977	0.009	0.025
223	Experiment	Spectroscopy	0.013	0.032	0.971	0.972	0.01	0.024
224	ExperimentActivity	Spectroscopy	0.014	0.029	0.975	0.975	0.011	0.022
225	Campaign	Spectroscopy	0.009	0.026	0.975	0.975	0.006	0.02
226	Correction	Spectroscopy	0.008	0.02	0.976	0.976	0.006	0.014
227	Difference	Spectroscopy	0.018	0.078	0.974	0.975	0.013	0.061
228	Hypothesis	Spectroscopy	0.02	0.049	0.977	0.977	0.015	0.036
229	Publication	Spectroscopy	0.005	0.005	0.974	0.974	0.004	0.003
230	Realization	Spectroscopy	0.004	0.005	0.976	0.976	0.003	0.003
231	Sample	Spectroscopy	0.026	0.066	0.972	0.973	0.019	0.05
232	Validation	Spectroscopy	0.011	0.032	0.979	0.979	0.008	0.024
233	Proof	Spectroscopy	0.005	0.006	0.981	0.981	0.004	0.004
234	Observation	Spectroscopy	0.036	0.119	0.974	0.976	0.026	0.091
235	HumanActivity	ResearchExploration	0.024	0.212	0.978	0.982	0.017	0.167
236	Research	ResearchExploration	0.023	0.328	0.974	0.982	0.017	0.266
237	Analysis	ResearchExploration	0.029	0.205	0.977	0.981	0.021	0.16
238	Investigation	ResearchExploration	0.022	0.326	0.98	0.986	0.016	0.264
239	Monitoring	ResearchExploration	0.025	0.367	0.978	0.986	0.018	0.307
240	Project	ResearchExploration	0.031	0.385	0.975	0.984	0.022	0.322
241	ProofOfConcept	ResearchExploration	0.02	0.081	0.977	0.978	0.015	0.062
242	ResearchSetting	ResearchExploration	0.036	0.307	0.977	0.985	0.13	0.365
243	Residual	ResearchExploration	0.017	0.217	0.977	0.982	0.012	0.172
244	Result	ResearchExploration	0.021	0.406	0.979	0.988	0.015	0.344
245	Representation	ResearchExploration	0.033	0.389	0.975	0.985	0.024	0.325
246	Variable	ResearchExploration	0.028	0.2	0.975	0.979	0.02	0.154
247	Assesment	ResearchExploration	0.031	0.289	0.977	0.983	0.023	0.241
248	Evidence	ResearchExploration	0.016	0.226	0.979	0.983	0.012	0.18
249	Experiment	ResearchExploration	0.04	0.423	0.975	0.985	0.029	0.364
250	ExperimentActivity	ResearchExploration	0.036	0.368	0.977	0.985	0.026	0.31
251	Campaign	ResearchExploration	0.017	0.125	0.977	0.98	0.012	0.097
252	Correction	ResearchExploration	0.021	0.083	0.978	0.979	0.015	0.062
253	Difference	ResearchExploration	0.017	0.236	0.976	0.982	0.013	0.19
254	Hypothesis	ResearchExploration	0.029	0.15	0.979	0.981	0.021	0.117
255	Publication	ResearchExploration	0.028	0.353	0.976	0.984	0.02	0.289
256	Realization	ResearchExploration	0.012	0.027	0.978	0.978	0.009	0.02
257	Sample	ResearchExploration	0.029	0.347	0.975	0.983	0.021	0.285
258	Validation	ResearchExploration	0.033	0.295	0.98	0.986	0.024	0.242
259	Proof	ResearchExploration	0.036	0.216	0.982	0.986	0.027	0.17
260	Observation	ResearchExploration	0.03	0.164	0.977	0.98	0.022	0.131
261	HumanActivity	Engineering	0.025	0.083	0.972	0.974	0.018	0.064
262	Research	Engineering	0.046	0.14	0.965	0.969	0.034	0.11
263	Analysis	Engineering	0.021	0.058	0.97	0.971	0.015	0.044
264	Investigation	Engineering	0.035	0.167	0.975	0.978	0.025	0.133
265	Monitoring	Engineering	0.035	0.124	0.973	0.975	0.026	0.104
266	Project	Engineering	0.054	0.133	0.967	0.969	0.04	0.11

267	ProofOfConcept	Engineering	0.026	0.03	0.97	0.97	0.019	0.022
268	ResearchSetting	Engineering	0.046	0.164	0.97	0.975	0.138	0.235
269	Residual	Engineering	0.03	0.074	0.971	0.972	0.022	0.057
270	Result	Engineering	0.031	0.165	0.974	0.978	0.023	0.136
271	Representation	Engineering	0.053	0.17	0.968	0.972	0.038	0.137
272	Variable	Engineering	0.036	0.092	0.967	0.971	0.026	0.113
273	Assesment	Engineering	0.034	0.092	0.971	0.973	0.025	0.077
274	Evidence	Engineering	0.029	0.077	0.973	0.974	0.021	0.06
275	Experiment	Engineering	0.059	0.225	0.966	0.972	0.044	0.191
276	ExperimentActivity	Engineering	0.052	0.176	0.971	0.975	0.039	0.144
277	Campaign	Engineering	0.032	0.068	0.97	0.971	0.023	0.051
278	Correction	Engineering	0.025	0.049	0.972	0.972	0.019	0.036
279	Difference	Engineering	0.03	0.105	0.969	0.971	0.022	0.081
280	Hypothesis	Engineering	0.051	0.257	0.973	0.979	0.037	0.202
281	Publication	Engineering	0.043	0.145	0.97	0.973	0.032	0.116
282	Realization	Engineering	0.028	0.101	0.971	0.974	0.021	0.076
283	Sample	Engineering	0.046	0.172	0.968	0.972	0.034	0.139
284	Validation	Engineering	0.051	0.183	0.976	0.979	0.038	0.147
285	Proof	Engineering	0.032	0.124	0.978	0.981	0.023	0.093
286	Observation	Engineering	0.044	0.108	0.97	0.972	0.033	0.084
287	HumanActivity	Imaging	0.008	0.081	0.976	0.978	0.006	0.063
288	Research	Imaging	0.026	0.157	0.971	0.975	0.019	0.122
289	Analysis	Imaging	0.021	0.106	0.975	0.977	0.015	0.082
290	Investigation	Imaging	0.012	0.078	0.978	0.98	0.009	0.06
291	Monitoring	Imaging	0.015	0.061	0.976	0.978	0.011	0.046
292	Project	Imaging	0.032	0.122	0.972	0.975	0.023	0.092
293	ProofOfConcept	Imaging	0.036	0.112	0.974	0.976	0.026	0.087
294	ResearchSetting	Imaging	0.013	0.058	0.977	0.978	0.113	0.149
295	Residual	Imaging	0.006	0.039	0.976	0.976	0.005	0.029
296	Result	Imaging	0.006	0.016	0.978	0.978	0.005	0.012
297	Representation	Imaging	0.008	0.044	0.974	0.975	0.006	0.034
298	Variable	Imaging	0.005	0.007	0.973	0.973	0.004	0.005
299	Assesment	Imaging	0.005	0.011	0.975	0.975	0.004	0.008
300	Evidence	Imaging	0.01	0.02	0.977	0.977	0.008	0.015
301	Experiment	Imaging	0.016	0.026	0.972	0.972	0.012	0.019
302	ExperimentActivity	Imaging	0.018	0.048	0.975	0.976	0.013	0.036
303	Campaign	Imaging	0.004	0.006	0.975	0.975	0.003	0.004
304	Correction	Imaging	0.013	0.052	0.976	0.977	0.009	0.039
305	Difference	Imaging	0.009	0.016	0.974	0.974	0.006	0.012
306	Hypothesis	Imaging	0.01	0.023	0.977	0.977	0.008	0.017
307	Publication	Imaging	0.01	0.013	0.975	0.975	0.007	0.01
308	Realization	Imaging	0.012	0.018	0.976	0.976	0.009	0.013
309	Sample	Imaging	0.011	0.021	0.973	0.973	0.008	0.016
310	Validation	Imaging	0.005	0.006	0.979	0.979	0.003	0.004
311	Proof	Imaging	0.004	0.008	0.981	0.981	0.003	0.006
312	Observation	Imaging	0.024	0.068	0.974	0.975	0.017	0.052
313	HumanActivity	Optics	0.014	0.044	0.98	0.981	0.011	0.032
314	Research	Optics	0.013	0.061	0.977	0.978	0.009	0.046
315	Analysis	Optics	0.033	0.097	0.979	0.981	0.024	0.073
316	Investigation	Optics	0.017	0.071	0.982	0.983	0.013	0.052
317	Monitoring	Optics	0.015	0.074	0.981	0.982	0.011	0.056
318	Project	Optics	0.021	0.094	0.978	0.979	0.016	0.073
319	ProofOfConcept	Optics	0.024	0.086	0.979	0.981	0.017	0.065
320	ResearchSetting	Optics	0.024	0.086	0.98	0.981	0.121	0.168

321	Residual	Optics	0.012	0.073	0.98	0.981	0.009	0.055
322	Result	Optics	0.026	0.112	0.981	0.983	0.019	0.087
323	Representation	Optics	0.032	0.125	0.978	0.98	0.024	0.095
324	Variable	Optics	0.033	0.056	0.977	0.977	0.024	0.041
325	Assesment	Optics	0.011	0.043	0.979	0.98	0.008	0.033
326	Evidence	Optics	0.008	0.04	0.981	0.981	0.006	0.03
327	Experiment	Optics	0.016	0.079	0.977	0.979	0.011	0.06
328	ExperimentActivity	Optics	0.025	0.091	0.98	0.981	0.018	0.07
329	Campaign	Optics	0.02	0.039	0.979	0.98	0.014	0.028
330	Correction	Optics	0.019	0.074	0.98	0.982	0.014	0.056
331	Difference	Optics	0.014	0.07	0.979	0.98	0.01	0.053
332	Hypothesis	Optics	0.022	0.039	0.981	0.981	0.016	0.029
333	Publication	Optics	0.021	0.069	0.979	0.98	0.015	0.052
334	Realization	Optics	0.005	0.008	0.98	0.98	0.004	0.006
335	Sample	Optics	0.038	0.131	0.978	0.98	0.028	0.102
336	Validation	Optics	0.021	0.067	0.982	0.983	0.015	0.05
337	Proof	Optics	0.021	0.048	0.984	0.984	0.016	0.035
338	Observation	Optics	0.032	0.107	0.979	0.98	0.023	0.081
339	HumanActivity	Photography	0.022	0.084	0.974	0.976	0.016	0.063
340	Research	Photography	0.023	0.11	0.969	0.972	0.016	0.084
341	Analysis	Photography	0.033	0.104	0.973	0.975	0.024	0.081
342	Investigation	Photography	0.025	0.177	0.977	0.981	0.018	0.136
343	Monitoring	Photography	0.018	0.093	0.975	0.977	0.013	0.073
344	Project	Photography	0.014	0.091	0.971	0.974	0.01	0.072
345	ProofOfConcept	Photography	0.011	0.053	0.974	0.975	0.008	0.041
346	ResearchSetting	Photography	0.03	0.087	0.973	0.975	0.126	0.172
347	Residual	Photography	0.018	0.082	0.974	0.976	0.013	0.062
348	Result	Photography	0.026	0.122	0.976	0.978	0.019	0.096
349	Representation	Photography	0.026	0.128	0.971	0.974	0.019	0.101
350	Variable	Photography	0.023	0.082	0.971	0.973	0.017	0.061
351	Assesment	Photography	0.013	0.063	0.974	0.975	0.009	0.049
352	Evidence	Photography	0.019	0.059	0.975	0.976	0.014	0.045
353	Experiment	Photography	0.034	0.185	0.97	0.975	0.025	0.145
354	ExperimentActivity	Photography	0.032	0.174	0.974	0.978	0.023	0.135
355	Campaign	Photography	0.012	0.027	0.974	0.974	0.008	0.02
356	Correction	Photography	0.033	0.143	0.975	0.978	0.024	0.111
357	Difference	Photography	0.032	0.092	0.972	0.974	0.023	0.07
358	Hypothesis	Photography	0.052	0.243	0.976	0.981	0.038	0.197
359	Publication	Photography	0.018	0.123	0.974	0.976	0.013	0.094
360	Realization	Photography	0.014	0.067	0.975	0.976	0.01	0.051
361	Sample	Photography	0.044	0.157	0.971	0.974	0.032	0.119
362	Validation	Photography	0.024	0.108	0.978	0.98	0.017	0.081
363	Proof	Photography	0.035	0.134	0.98	0.982	0.025	0.106
364	Observation	Photography	0.069	0.194	0.973	0.976	0.051	0.157
365	HumanActivity	RemoteSensing	0.028	0.087	0.975	0.976	0.02	0.066
366	Research	RemoteSensing	0.023	0.091	0.97	0.972	0.017	0.069
367	Analysis	RemoteSensing	0.028	0.06	0.973	0.974	0.02	0.044
368	Investigation	RemoteSensing	0.043	0.177	0.977	0.98	0.031	0.139
369	Monitoring	RemoteSensing	0.019	0.086	0.975	0.977	0.014	0.066
370	Project	RemoteSensing	0.025	0.083	0.971	0.973	0.018	0.063
371	ProofOfConcept	RemoteSensing	0.015	0.024	0.974	0.974	0.011	0.018
372	ResearchSetting	RemoteSensing	0.024	0.087	0.974	0.976	0.121	0.169
373	Residual	RemoteSensing	0.014	0.041	0.974	0.975	0.01	0.03
374	Result	RemoteSensing	0.033	0.139	0.977	0.979	0.024	0.109

375	Representation	RemoteSensing	0.018	0.079	0.972	0.974	0.013	0.060
376	Variable	RemoteSensing	0.028	0.056	0.971	0.972	0.02	0.042
377	Assesment	RemoteSensing	0.013	0.048	0.974	0.975	0.009	0.037
378	Evidence	RemoteSensing	0.020	0.066	0.976	0.977	0.015	0.05
379	Experiment	RemoteSensing	0.036	0.117	0.970	0.973	0.026	0.091
380	ExperimentActivity	RemoteSensing	0.046	0.158	0.974	0.977	0.034	0.122
381	Campaign	RemoteSensing	0.028	0.055	0.973	0.974	0.02	0.041
382	Correction	RemoteSensing	0.018	0.069	0.975	0.977	0.013	0.052
383	Difference	RemoteSensing	0.042	0.118	0.973	0.975	0.032	0.092
384	Hypothesis	RemoteSensing	0.047	0.176	0.976	0.979	0.034	0.139
385	Publication	RemoteSensing	0.037	0.112	0.973	0.975	0.027	0.086
386	Realization	RemoteSensing	0.004	0.005	0.975	0.975	0.003	0.004
387	Sample	RemoteSensing	0.054	0.13	0.971	0.973	0.04	0.1
388	Validation	RemoteSensing	0.034	0.091	0.977	0.979	0.024	0.068
389	proof	RemoteSensing	0.031	0.131	0.981	0.983	0.022	0.099
390	Observation	RemoteSensing	0.069	0.209	0.973	0.977	0.051	0.164
391	HumanActivity	Tomography	0.007	0.009	0.978	0.978	0.005	0.007
392	Research	Tomography	0.005	0.005	0.974	0.974	0.003	0.004
393	Analysis	Tomography	0.017	0.028	0.977	0.977	0.012	0.021
394	Investigation	Tomography	0.021	0.033	0.979	0.979	0.015	0.024
395	Monitoring	Tomography	0	0	0.978	0.978	0	0
396	Project	Tomography	0.006	0.012	0.975	0.975	0.004	0.009
397	ProofOfConcept	Tomography	0.012	0.031	0.977	0.977	0.009	0.023
398	ResearchSetting	Tomography	0.003	0.035	0.977	0.974	0.106	0.106
399	Residual	Tomography	0.009	0.021	0.977	0.977	0.006	0.015
400	Result	Tomography	0.038	0.09	0.979	0.98	0.028	0.067
401	Representation	Tomography	0.01	0.023	0.975	0.976	0.007	0.017
402	Variable	Tomography	0.014	0.049	0.975	0.976	0.01	0.037
403	Assesment	Tomography	0	0	0.977	0.977	0	0
404	Evidence	Tomography	0.014	0.022	0.978	0.979	0.01	0.016
405	Experiment	Tomography	0.014	0.049	0.974	0.975	0.01	0.037
406	ExperimentActivity	Tomography	0.015	0.02	0.977	0.977	0.011	0.015
407	Campaign	Tomography	0.007	0.01	0.977	0.977	0.005	0.008
408	Correction	Tomography	0.004	0.016	0.978	0.979	0.003	0.012
409	Difference	Tomography	0.019	0.045	0.976	0.977	0.014	0.034
410	Hypothesis	Tomography	0.014	0.083	0.979	0.98	0.01	0.063
411	Publication	Tomography	0.016	0.051	0.976	0.977	0.012	0.038
412	Realization	Tomography	0.006	0.033	0.978	0.979	0.004	0.025
413	Sample	Tomography	0.019	0.049	0.975	0.976	0.014	0.036
414	Validation	Tomography	0.018	0.055	0.98	0.981	0.013	0.041
415	Proof	Tomography	0.013	0.046	0.982	0.983	0.009	0.034
416	Observation	Tomography	0.031	0.09	0.976	0.978	0.022	0.068
417	HumanActivity	XrayDiffraction	0.013	0.026	0.974	0.975	0.009	0.02
418	Research	XrayDiffraction	0	0	0.97	0.97	0	0
419	Analysis	XrayDiffraction	0.024	0.053	0.973	0.974	0.018	0.039
420	Investigation	XrayDiffraction	0.02	0.041	0.977	0.977	0.015	0.03
421	Monitoring	XrayDiffraction	0.01	0.018	0.975	0.975	0.007	0.014
422	Project	XrayDiffraction	0.006	0.012	0.971	0.971	0.005	0.009
423	ProofOfConcept	XrayDiffraction	0.011	0.018	0.973	0.974	0.008	0.013
424	ResearchSetting	XrayDiffraction	0.01	0.035	0.973	0.976	0.111	0.129
425	Residual	XrayDiffraction	0.012	0.013	0.973	0.973	0.009	0.01
426	Result	XrayDiffraction	0.027	0.079	0.976	0.977	0.02	0.06
427	Representation	XrayDiffraction	0.012	0.013	0.971	0.971	0.009	0.01
428	Variable	XrayDiffraction	0.026	0.099	0.971	0.973	0.019	0.076

429	Assesment	XrayDiffraction	0	0	0.973	0.973	0	0
430	Evidence	XrayDiffraction	0.009	0.027	0.976	0.976	0.007	0.02
431	Experiment	XrayDiffraction	0.018	0.093	0.97	0.972	0.013	0.074
432	ExperimentActivity	XrayDiffraction	0.01	0.02	0.974	0.974	0.007	0.015
433	Campaign	XrayDiffraction	0.013	0.03	0.973	0.974	0.009	0.022
434	Correction	XrayDiffraction	0.008	0.021	0.975	0.975	0.006	0.015
435	Difference	XrayDiffraction	0.015	0.075	0.973	0.974	0.011	0.058
436	Hypothesis	XrayDiffraction	0.033	0.158	0.975	0.978	0.024	0.122
437	Publication	XrayDiffraction	0.004	0.005	0.973	0.973	0.003	0.004
438	Realization	XrayDiffraction	0.014	0.065	0.975	0.976	0.01	0.05
439	Sample	XrayDiffraction	0.027	0.085	0.971	0.972	0.02	0.063
440	Validation	XrayDiffraction	0.021	0.098	0.977	0.979	0.016	0.075
441	Proof	XrayDiffraction	0.009	0.012	0.98	0.98	0.007	0.009
442	Observation	XrayDiffraction	0.036	0.136	0.973	0.975	0.026	0.104

Appendix II: Semantic Network (WordNetSimilarity) Measures Scores

S/N	Concept1	Concept2	LIPA	JCN	WUP	PAth	LIn
1	HumanActivity	ExperimentActivity	0.072	0.074	0.4	0.077	0.071
2	Research	ExperimentActivity	1.096	2.135	0.917	0.334	0.971
3	Analysis	ExperimentActivity	0.86	0.3	0.834	0.2	0.798
4	Investigation	ExperimentActivity	0.957	0.645	0.87	0.25	0.895
5	Monitoring	ExperimentActivity	0.316	0.075	0.616	0.091	0.3
6	Project	ExperimentActivity	0.143	0.059	0.435	0.072	0.141
7	ProofOfConcept	ExperimentActivity	0.16	0.067	0.455	0.077	0.158
8	ResearchSetting	ExperimentActivity	0.056	0.057	0.364	0.067	0.055
9	Residual	ExperimentActivity	0.064	0.065	0.4	0.077	0.063
10	Result	ExperimentActivity	0.001	0.075	0.211	0.063	0
11	Representation	ExperimentActivity	0.32	0.076	0.728	0.143	0.304
12	Variable	ExperimentActivity	0.001	0.052	0.223	0.067	0
13	Assesment	ExperimentActivity	0.248	0.077	0.667	0.125	0.24
14	Evidence	ExperimentActivity	0.188	0.082	0.477	0.084	0.186
15	Experiment	ExperimentActivity	0.179	0.077	0.4	0.063	0.177
16	ExperimentActivity	ExperimentActivity	1	272401	1	1	1
17	Campaign	ExperimentActivity	0.556	0.129	0.72	0.125	0.525
18	Correction	ExperimentActivity	0.236	0.072	0.56	0.084	0.228
19	Difference	ExperimentActivity	0.201	0.072	0.522	0.084	0.197
20	Hypothesis	ExperimentActivity	0.192	0.084	0.435	0.072	0.19
21	Publication	ExperimentActivity	0.001	0.075	0.174	0.05	0
22	Realization	ExperimentActivity	0.148	0.061	0.417	0.067	0.146
23	Sample	ExperimentActivity	0.158	0.066	0.435	0.072	0.156
24	Validation	ExperimentActivity	0.223	0.067	0.539	0.077	0.215
25	Proof	ExperimentActivity	0.223	0.067	0.539	0.077	0.215
26	Observation	ExperimentActivity	0.35	0.087	0.696	0.125	0.334
27	HumanActivity	Methodology	0.066	0.06	0.471	0.1	0.058
28	Research	Methodology	0.307	0.075	0.572	0.1	0.291
29	Analysis	Methodology	0.15	0.059	0.477	0.084	0.142
30	Investigation	Methodology	0.172	0.066	0.5	0.091	0.157
31	Monitoring	Methodology	0.127	0.051	0.435	0.072	0.125
32	Project	Methodology	0.268	0.057	0.6	0.112	0.237
33	ProofOfConcept	Methodology	0.291	0.064	0.632	0.125	0.26
34	ResearchSetting	Methodology	0.055	0.048	0.422	0.084	0.047
35	Residual	Methodology	0.06	0.053	0.471	0.1	0.052
36	Result	Methodology	0.002	0.06	0.25	0.077	0
37	Representation	Methodology	0.142	0.052	0.527	0.1	0.126
38	Variable	Methodology	0.002	0.045	0.267	0.084	0
39	Assesment	Methodology	0.153	0.057	0.556	0.112	0.137
40	Evidence	Methodology	0.331	0.078	0.667	0.143	0.299
41	Experiment	Methodology	0.295	0.074	0.546	0.091	0.287

42	ExperimentActivity	Methodology	0.146	0.06	0.435	0.072	0.144
43	Campaign	Methodology	0.147	0.06	0.455	0.077	0.143
44	Correction	Methodology	0.135	0.054	0.455	0.077	0.131
45	Difference	Methodology	0.152	0.056	0.5	0.091	0.136
46	Hypothesis	Methodology	0.335	0.08	0.6	0.112	0.304
47	Publication	Methodology	0.002	0.06	0.2	0.059	0
48	Realization	Methodology	0.26	0.059	0.572	0.1	0.244
49	Sample	Methodology	0.289	0.063	0.6	0.112	0.258
50	Validation	Methodology	0.127	0.051	0.435	0.072	0.125
51	Proof	Methodology	0.127	0.051	0.435	0.072	0.125
52	Observation	Methodology	0.152	0.057	0.5	0.091	0.137
53	HumanActivity	ResearchSci	0.078	0.077	0.445	0.091	0.074
54	ResearchHuman	ResearchSci	1	272401	1	1	1
55	Analysis	ResearchSci	1.071	0.349	0.91	0.334	0.821
56	Investigation	ResearchSci	1.174	0.924	0.953	0.5	0.924
57	Monitoring	ResearchSci	0.323	0.077	0.667	0.112	0.308
58	Project	ResearchSci	0.152	0.06	0.477	0.084	0.145
59	ProofOfConcept	ResearchSci	0.17	0.069	0.5	0.091	0.162
60	ResearchSetting	ResearchSci	0.061	0.058	0.4	0.077	0.057
61	Residual	ResearchSci	0.068	0.067	0.445	0.091	0.064
62	Result	ResearchSci	0.001	0.077	0.236	0.072	0
63	Representation	ResearchSci	0.374	0.079	0.8	0.2	0.312
64	Variable	ResearchSci	0.001	0.054	0.25	0.077	0
65	Assesment	ResearchSci	0.278	0.08	0.737	0.167	0.247
66	Evidence	ResearchSci	0.2	0.085	0.527	0.1	0.192
67	Experiment	ResearchSci	0.187	0.08	0.435	0.072	0.183
68	ExperimentActivity	ResearchSci	1.096	2.135	0.917	0.334	0.971
69	Campaign	ResearchSci	0.603	0.137	0.783	0.167	0.54
70	Correction	ResearchSci	0.249	0.074	0.609	0.1	0.234
71	Difference	ResearchSci	0.218	0.074	0.572	0.1	0.202
72	Hypothesis	ResearchSci	0.204	0.087	0.477	0.084	0.196
73	Publication	ResearchSci	0.001	0.077	0.191	0.056	0
74	Realization	ResearchSci	0.158	0.063	0.455	0.077	0.15
75	Sample	ResearchSci	0.168	0.068	0.477	0.084	0.16
76	Validation	ResearchSci	0.228	0.069	0.584	0.091	0.22
77	Proof	ResearchSci	0.228	0.069	0.584	0.091	0.22
78	Observation	ResearchSci	0.406	0.091	0.762	0.167	0.343
79	HumanActivity	RetrievalAproach	0.07	0.07	0.422	0.084	0.068
80	Research	RetrievalAproach	0.38	0.093	0.696	0.125	0.349
81	Analysis	RetrievalAproach	0.366	0.088	0.696	0.125	0.335
82	Investigation	RetrievalAproach	0.405	0.104	0.728	0.143	0.374
83	Monitoring	RetrievalAproach	0.304	0.071	0.64	0.1	0.288
84	Project	RetrievalAproach	0.14	0.056	0.455	0.077	0.136
85	ProofOfConcept	RetrievalAproach	0.155	0.064	0.477	0.084	0.151

86	ResearchSetting	RetrievalAproach	0.055	0.054	0.381	0.072	0.053
87	Residual	RetrievalAproach	0.062	0.062	0.422	0.084	0.06
88	Result	RetrievalAproach	0.001	0.07	0.223	0.067	0
89	Representation	RetrievalAproach	0.323	0.072	0.762	0.167	0.292
90	Variable	RetrievalAproach	0.001	0.05	0.236	0.072	0
91	Assesment	RetrievalAproach	0.245	0.073	0.7	0.143	0.23
92	Evidence	RetrievalAproach	0.181	0.077	0.5	0.091	0.177
93	Experiment	RetrievalAproach	0.173	0.073	0.417	0.067	0.169
94	ExperimentActivity	RetrievalAproach	0.355	0.089	0.64	0.1	0.34
95	Campaign	RetrievalAproach	0.367	0.088	0.667	0.112	0.336
96	Correction	RetrievalAproach	0.234	0.068	0.584	0.091	0.218
97	Difference	RetrievalAproach	0.196	0.068	0.546	0.091	0.188
98	Hypothesis	RetrievalAproach	0.184	0.079	0.455	0.077	0.18
99	Publication	RetrievalAproach	0.001	0.07	0.182	0.053	0
100	Realization	RetrievalAproach	0.144	0.058	0.435	0.072	0.14
101	Sample	RetrievalAproach	0.153	0.063	0.455	0.077	0.149
102	Validation	RetrievalAproach	0.214	0.063	0.56	0.084	0.206
103	Proof	RetrievalAproach	0.214	0.063	0.56	0.084	0.206
104	Observation	RetrievalAproach	0.351	0.082	0.728	0.143	0.319
105	HumanActivity	CarbonDating	0	0	0.4	0.077	0
106	ResearcHuman	CarbonDating	0	0	0.834	0.2	0
107	Analysis	CarbonDating	0	0	0.917	0.334	0
108	Investigation	CarbonDating	0	0	0.87	0.25	0
109	Monitoring	CarbonDating	0	0	0.616	0.091	0
110	Project	CarbonDating	0	0	0.435	0.072	0
111	ProofOfConcept	CarbonDating	0	0	0.455	0.077	0
112	ResearchSetting	CarbonDating	0	0	0.364	0.067	0
113	Residual	CarbonDating	0	0	0.4	0.077	0
114	Result	CarbonDating	0	0	0.211	0.063	0
115	Representation	CarbonDating	0	0	0.728	0.143	0
116	Variable	CarbonDating	0	0	0.223	0.067	0
117	Assesment	CarbonDating	0	0	0.667	0.125	0
118	Evidence	CarbonDating	0	0	0.477	0.084	0
119	Experiment	CarbonDating	0	0	0.4	0.063	0
120	ExperimentActivity	CarbonDating	0	0	0.77	0.143	0
121	Campaign	CarbonDating	0	0	0.72	0.125	0
122	Correction	CarbonDating	0	0	0.56	0.084	0
123	Difference	CarbonDating	0	0	0.522	0.084	0
124	Hypothesis	CarbonDating	0	0	0.435	0.072	0
125	Publication	CarbonDating	0	0	0.174	0.05	0
126	Realization	CarbonDating	0	0	0.667	0.112	0
127	Sample	CarbonDating	0	0	0.435	0.072	0
128	Validation	CarbonDating	0	0	0.539	0.077	0
129	Proof	CarbonDating	0	0	0.539	0.077	0

130	Observation	CarbonDating	0	0	0.696	0.125	0
131	HumanActivity	IsotopeAnalysis	0.065	0.063	0.445	0.091	0.061
132	Research	IsotopeAnalysis	0.162	0.065	0.455	0.077	0.155
133	Analysis	IsotopeAnalysis	0.157	0.063	0.455	0.077	0.149
134	Investigation	IsotopeAnalysis	0.172	0.07	0.477	0.084	0.164
135	Monitoring	IsotopeAnalysis	0.132	0.053	0.417	0.067	0.13
136	Project	IsotopeAnalysis	0.262	0.06	0.572	0.1	0.246
137	ProofOfConcept	IsotopeAnalysis	0.287	0.068	0.6	0.112	0.271
138	ResearchSetting	IsotopeAnalysis	0.053	0.05	0.4	0.077	0.049
139	Residual	IsotopeAnalysis	0.059	0.056	0.445	0.091	0.055
140	Result	IsotopeAnalysis	0.001	0.063	0.236	0.072	0
141	Representation	IsotopeAnalysis	0.139	0.054	0.5	0.091	0.131
142	Variable	IsotopeAnalysis	0.001	0.047	0.25	0.077	0
143	Assesment	IsotopeAnalysis	0.151	0.06	0.527	0.1	0.143
144	Evidence	IsotopeAnalysis	0.329	0.083	0.632	0.125	0.314
145	Experiment	IsotopeAnalysis	0.696	0.149	0.783	0.167	0.634
146	ExperimentActivity	IsotopeAnalysis	0.153	0.063	0.417	0.067	0.151
147	Campaign	IsotopeAnalysis	0.153	0.063	0.435	0.072	0.149
148	Correction	IsotopeAnalysis	0.141	0.057	0.435	0.072	0.137
149	Difference	IsotopeAnalysis	0.149	0.059	0.477	0.084	0.142
150	Hypothesis	IsotopeAnalysis	0.335	0.086	0.572	0.1	0.319
151	Publication	IsotopeAnalysis	0.001	0.063	0.191	0.056	0
152	Realization	IsotopeAnalysis	0.571	0.094	0.728	0.143	0.509
153	Sample	IsotopeAnalysis	0.284	0.067	0.572	0.1	0.268
154	Validation	IsotopeAnalysis	0.132	0.053	0.417	0.067	0.13
155	Proof	IsotopeAnalysis	0.132	0.053	0.417	0.067	0.13
156	Observation	IsotopeAnalysis	0.15	0.059	0.477	0.084	0.142
157	HumanActivity	Photometry	0	0	0.471	0.1	0
158	Research	Photometry	0	0	0.762	0.167	0
159	Analysis	Photometry	0	0	0.762	0.167	0
160	Investigation	Photometry	0	0	0.8	0.2	0
161	Monitoring	Photometry	0	0	0.696	0.125	0
162	Project	Photometry	0	0	0.5	0.091	0
163	ProofOfConcept	Photometry	0	0	0.527	0.1	0
164	ResearchSetting	Photometry	0	0	0.422	0.084	0
165	Residual	Photometry	0	0	0.471	0.1	0
166	Result	Photometry	0	0	0.25	0.077	0
167	Representation	Photometry	0	0	0.843	0.25	0
168	Variable	Photometry	0	0	0.267	0.084	0
169	Assesment	Photometry	0	0	0.778	0.2	0
170	Evidence	Photometry	0	0	0.556	0.112	0
171	Experiment	Photometry	0	0	0.455	0.077	0
172	ExperimentActivity	Photometry	0	0	0.696	0.125	0
173	Campaign	Photometry	0	0	0.728	0.143	0

174	Correction	Photometry	0	0	0.637	0.112	0
175	Difference	Photometry	0	0	0.6	0.112	0
176	Hypothesis	Photometry	0	0	0.5	0.091	0
177	Publication	Photometry	0	0	0.2	0.059	0
178	Realization	Photometry	0	0	0.477	0.084	0
179	Sample	Photometry	0	0	0.5	0.091	0
180	Validation	Photometry	0	0	0.609	0.1	0
181	Proof	Photometry	0	0	0.609	0.1	0
182	Observation	Photometry	0	0	0.9	0.334	0
183	HumanActivity	RadioactiveDating	0.008	0.055	0.267	0.084	0
184	Research	RadioactiveDating	0.001	0.052	0.211	0.063	0
185	Analysis	RadioactiveDating	0.001	0.05	0.211	0.063	0
186	Investigation	RadioactiveDating	0.002	0.055	0.223	0.067	0
187	Monitoring	RadioactiveDating	0.001	0.044	0.191	0.056	0
188	Project	RadioactiveDating	0.002	0.043	0.223	0.067	0
189	ProofOfConcept	RadioactiveDating	0.004	0.047	0.236	0.072	0
190	ResearchSetting	RadioactiveDating	0.004	0.045	0.236	0.072	0
191	Residual	RadioactiveDating	0.008	0.05	0.267	0.084	0
192	Result	RadioactiveDating	0.485	0.107	0.572	0.143	0.454
193	Representation	RadioactiveDating	0.004	0.044	0.236	0.072	0
194	Variable	RadioactiveDating	0.095	0.048	0.462	0.125	0.08
195	Assesment	RadioactiveDating	0.008	0.048	0.25	0.077	0
196	Evidence	RadioactiveDating	0.008	0.054	0.25	0.077	0
197	Experiment	RadioactiveDating	0.001	0.052	0.2	0.059	0
198	ExperimentActivity	RadioactiveDating	0.001	0.051	0.191	0.056	0
199	Campaign	RadioactiveDating	0.001	0.05	0.2	0.059	0
200	Correction	RadioactiveDating	0.001	0.046	0.2	0.059	0
201	Difference	RadioactiveDating	0.002	0.048	0.223	0.067	0
202	Hypothesis	RadioactiveDating	0.002	0.055	0.223	0.067	0
203	Publication	RadioactiveDating	0.11	0.065	0.334	0.077	0.106
204	Realization	RadioactiveDating	0.001	0.044	0.211	0.063	0
205	Sample	RadioactiveDating	0.002	0.046	0.223	0.067	0
206	Validation	RadioactiveDating	0.001	0.044	0.191	0.056	0
207	Proof	RadioactiveDating	0.001	0.044	0.191	0.056	0
208	Observation	RadioactiveDating	0.002	0.048	0.223	0.067	0
209	HumanActivity	Spectroscopy	0.059	0.06	0.4	0.077	0.058
210	Research	Spectroscopy	0.751	0.169	0.834	0.2	0.689
211	Analysis	Spectroscopy	0.97	0.326	0.917	0.334	0.845
212	Investigation	Spectroscopy	0.793	0.206	0.87	0.25	0.73
213	Monitoring	Spectroscopy	0.272	0.06	0.616	0.091	0.256
214	Project	Spectroscopy	0.123	0.049	0.435	0.072	0.121
215	ProofOfConcept	Spectroscopy	0.135	0.055	0.455	0.077	0.133
216	ResearchSetting	Spectroscopy	0.048	0.048	0.364	0.067	0.047
217	Residual	Spectroscopy	0.053	0.053	0.4	0.077	0.052

218	Result	Spectroscopy	0.001	0.06	0.211	0.063	0
219	Representation	Spectroscopy	0.275	0.061	0.728	0.143	0.259
220	Variable	Spectroscopy	0.001	0.045	0.223	0.067	0
221	Assesment	Spectroscopy	0.209	0.062	0.667	0.125	0.202
222	Evidence	Spectroscopy	0.155	0.065	0.477	0.084	0.153
223	Experiment	Spectroscopy	0.149	0.062	0.4	0.063	0.147
224	ExperimentActivity	Spectroscopy	0.735	0.156	0.77	0.143	0.672
225	Campaign	Spectroscopy	0.468	0.09	0.72	0.125	0.437
226	Correction	Spectroscopy	0.201	0.058	0.56	0.084	0.193
227	Difference	Spectroscopy	0.169	0.058	0.522	0.084	0.165
228	Hypothesis	Spectroscopy	0.157	0.066	0.435	0.072	0.155
229	Publication	Spectroscopy	0.001	0.06	0.174	0.05	0
230	Realization	Spectroscopy	0.127	0.051	0.417	0.067	0.125
231	Sample	Spectroscopy	0.134	0.054	0.435	0.072	0.132
232	Validation	Spectroscopy	0.191	0.055	0.539	0.077	0.183
233	Proof	Spectroscopy	0.191	0.055	0.539	0.077	0.183
234	Observation	Spectroscopy	0.296	0.068	0.696	0.125	0.281
235	HumanActivity	ResearchExploration	0.067	0.061	0.471	0.1	0.059
236	Research	ResearchExploration	0.37	0.077	0.762	0.167	0.308
237	Analysis	ResearchExploration	0.359	0.073	0.762	0.167	0.297
238	Investigation	ResearchExploration	0.452	0.084	0.8	0.2	0.327
239	Monitoring	ResearchExploration	0.275	0.061	0.696	0.125	0.26
240	Project	ResearchExploration	0.138	0.05	0.5	0.091	0.123
241	ProofOfConcept	ResearchExploration	0.15	0.056	0.527	0.1	0.135
242	ResearchSetting	ResearchExploration	0.056	0.048	0.422	0.084	0.048
243	Residual	ResearchExploration	0.061	0.054	0.471	0.1	0.053
244	Result	ResearchExploration	0.002	0.061	0.25	0.077	0
245	Representation	ResearchExploration	0.387	0.062	0.843	0.25	0.262
246	Variable	ResearchExploration	0.002	0.045	0.267	0.084	0
247	Assesment	ResearchExploration	0.267	0.063	0.778	0.2	0.204
248	Evidence	ResearchExploration	0.171	0.066	0.556	0.112	0.155
249	Experiment	ResearchExploration	0.153	0.063	0.455	0.077	0.149
250	ExperimentActivity	ResearchExploration	0.316	0.075	0.696	0.125	0.3
251	Campaign	ResearchExploration	0.329	0.074	0.728	0.143	0.297
252	Correction	ResearchExploration	0.211	0.059	0.637	0.112	0.195
253	Difference	ResearchExploration	0.199	0.059	0.6	0.112	0.168
254	Hypothesis	ResearchExploration	0.174	0.067	0.5	0.091	0.158
255	Publication	ResearchExploration	0.195	0.056	0.609	0.1	0.188
256	Realization	ResearchExploration	0.134	0.052	0.477	0.084	0.126
257	Sample	ResearchExploration	0.149	0.055	0.5	0.091	0.134
258	Validation	ResearchExploration	0.193	0.056	0.609	0.1	0.186
259	Proof	ResearchExploration	0.193	0.056	0.609	0.1	0.186
260	Observation	ResearchExploration	0.409	0.069	0.8	0.2	0.284
261	HumanActivity	Engineering	0.067	0.061	0.471	0.1	0.059

262	Research	Engineering	0.158	0.063	0.477	0.084	0.15
263	Analysis	Engineering	0.152	0.06	0.477	0.084	0.145
264	Investigation	Engineering	0.175	0.068	0.5	0.091	0.159
265	Monitoring	Engineering	0.128	0.052	0.435	0.072	0.126
266	Project	Engineering	0.343	0.061	0.7	0.143	0.28
267	ProofOfConcept	Engineering	0.295	0.065	0.632	0.125	0.264
268	ResearchSetting	Engineering	0.056	0.048	0.422	0.084	0.048
269	Residual	Engineering	0.061	0.054	0.471	0.1	0.053
270	Result	Engineering	0.002	0.061	0.25	0.077	0
271	Representation	Engineering	0.143	0.052	0.527	0.1	0.128
272	Variable	Engineering	0.002	0.045	0.267	0.084	0
273	Assesment	Engineering	0.155	0.058	0.556	0.112	0.139
274	Evidence	Engineering	0.335	0.08	0.667	0.143	0.304
275	Experiment	Engineering	0.3	0.075	0.546	0.091	0.292
276	ExperimentActivity	Engineering	0.148	0.061	0.435	0.072	0.146
277	Campaign	Engineering	0.149	0.061	0.455	0.077	0.145
278	Correction	Engineering	0.137	0.055	0.455	0.077	0.133
279	Difference	Engineering	0.153	0.057	0.5	0.091	0.138
280	Hypothesis	Engineering	0.424	0.088	0.7	0.143	0.361
281	Publication	Engineering	0.211	0.062	0.584	0.091	0.203
282	Realization	Engineering	0.144	0.057	0.455	0.077	0.137
283	Sample	Engineering	0.153	0.061	0.477	0.084	0.145
284	Validation	Engineering	0.128	0.052	0.435	0.072	0.126
285	Proof	Engineering	0.128	0.052	0.435	0.072	0.126
286	Observation	Engineering	0.154	0.058	0.5	0.091	0.139
287	HumanActivity	Imaging	0	0	0.445	0.091	0
288	Research	Imaging	0	0	0.728	0.143	0
289	Analysis	Imaging	0	0	0.728	0.143	0
290	Investigation	Imaging	0	0	0.762	0.167	0
291	Monitoring	Imaging	0	0	0.667	0.112	0
292	Project	Imaging	0	0	0.477	0.084	0
293	ProofOfConcept	Imaging	0	0	0.5	0.091	0
294	ResearchSetting	Imaging	0	0	0.4	0.077	0
295	Residual	Imaging	0	0	0.445	0.091	0
296	Result	Imaging	0	0	0.236	0.072	0
297	Representation	Imaging	0	0	0.9	0.334	0
298	Variable	Imaging	0	0	0.25	0.077	0
299	Assesment	Imaging	0	0	0.737	0.167	0
300	Evidence	Imaging	0	0	0.527	0.1	0
301	Experiment	Imaging	0	0	0.435	0.072	0
302	ExperimentActivity	Imaging	0	0	0.667	0.112	0
303	Campaign	Imaging	0	0	0.696	0.125	0
304	Correction	Imaging	0	0	0.609	0.1	0
305	Difference	Imaging	0	0	0.572	0.1	0

306	Hypothesis	Imaging	0	0	0.477	0.084	0
307	Publication	Imaging	0	0	0.191	0.056	0
308	Realization	Imaging	0	0	0.455	0.077	0
309	Sample	Imaging	0	0	0.477	0.084	0
310	Validation	Imaging	0	0	0.584	0.091	0
311	Proof	Imaging	0	0	0.584	0.091	0
312	Observation	Imaging	0	0	0.762	0.167	0
313	HumanActivity	Optics	0	0	0.4	0.077	0
314	Research	Optics	0	0	0.417	0.067	0
315	Analysis	Optics	0	0	0.417	0.067	0
316	Investigation	Optics	0	0	0.435	0.072	0
317	Monitoring	Optics	0	0	0.385	0.059	0
318	Project	Optics	0	0	0.609	0.1	0
319	ProofOfConcept	Optics	0	0	0.546	0.091	0
320	ResearchSetting	Optics	0	0	0.364	0.067	0
321	Residual	Optics	0	0	0.4	0.077	0
322	Result	Optics	0	0	0.211	0.063	0
323	Representation	Optics	0	0	0.455	0.077	0
324	Variable	Optics	0	0	0.223	0.067	0
325	Assesment	Optics	0	0	0.477	0.084	0
326	Evidence	Optics	0	0	0.572	0.1	0
327	Experiment	Optics	0	0	0.385	0.059	0
328	ExperimentActivity	Optics	0	0	0.48	0.072	0
329	Campaign	Optics	0	0	0.4	0.063	0
330	Correction	Optics	0	0	0.4	0.063	0
331	Difference	Optics	0	0	0.435	0.072	0
332	Hypothesis	Optics	0	0	0.609	0.1	0
333	Publication	Optics	0	0	0.174	0.05	0
334	Realization	Optics	0	0	0.5	0.077	0
335	Sample	Optics	0	0	0.522	0.084	0
336	Validation	Optics	0	0	0.385	0.059	0
337	Proof	Optics	0	0	0.385	0.059	0
338	Observation	Optics	0	0	0.435	0.072	0
339	HumanActivity	Photography	0.016	0.053	0.334	0.112	0
340	Research	Photography	0.001	0.05	0.25	0.077	0
341	Analysis	Photography	0.001	0.048	0.25	0.077	0
342	Investigation	Photography	0.002	0.053	0.267	0.084	0
343	Monitoring	Photography	0.001	0.043	0.223	0.067	0
344	Project	Photography	0.002	0.042	0.267	0.084	0
345	ProofOfConcept	Photography	0.004	0.045	0.286	0.091	0
346	ResearchSetting	Photography	0.004	0.043	0.286	0.091	0
347	Residual	Photography	0.016	0.048	0.334	0.112	0
348	Result	Photography	0.562	0.1	0.728	0.25	0.437
349	Representation	Photography	0.004	0.043	0.286	0.091	0

350	Variable	Photography	0.202	0.046	0.6	0.2	0.077
351	Assesment	Photography	0.008	0.047	0.308	0.1	0
352	Evidence	Photography	0.008	0.052	0.308	0.1	0
353	Experiment	Photography	0.001	0.05	0.236	0.072	0
354	ExperimentActivity	Photography	0.001	0.049	0.223	0.067	0
355	Campaign	Photography	0.001	0.049	0.236	0.072	0
356	Correction	Photography	0.001	0.045	0.236	0.072	0
357	Difference	Photography	0.002	0.046	0.267	0.084	0
358	Hypothesis	Photography	0.002	0.053	0.267	0.084	0
359	Publication	Photography	0.106	0.063	0.4	0.1	0.102
360	Realization	Photography	0.001	0.043	0.25	0.077	0
361	Sample	Photography	0.002	0.045	0.267	0.084	0
362	Validation	Photography	0.001	0.043	0.223	0.067	0
363	Proof	Photography	0.001	0.043	0.223	0.067	0
364	Observation	Photography	0	0	0.435	0.072	0
365	HumanActivity	RemoteSensing	0.069	0.063	0.471	0.1	0.061
366	Research	RemoteSensing	0.162	0.065	0.477	0.084	0.155
367	Analysis	RemoteSensing	0.157	0.063	0.477	0.084	0.149
368	Investigation	RemoteSensing	0.18	0.07	0.5	0.091	0.164
369	Monitoring	RemoteSensing	0.132	0.053	0.435	0.072	0.13
370	Project	RemoteSensing	0.277	0.06	0.6	0.112	0.246
371	ProofOfConcept	RemoteSensing	0.302	0.068	0.632	0.125	0.271
372	ResearchSetting	RemoteSensing	0.057	0.05	0.422	0.084	0.049
373	Residual	RemoteSensing	0.062	0.056	0.471	0.1	0.055
374	Result	RemoteSensing	0.002	0.063	0.25	0.077	0
375	Representation	RemoteSensing	0.147	0.054	0.527	0.1	0.131
376	Variable	RemoteSensing	0.002	0.047	0.267	0.084	0
377	Assesment	RemoteSensing	0.159	0.06	0.556	0.112	0.143
378	Evidence	RemoteSensing	0.345	0.083	0.667	0.143	0.314
379	Experiment	RemoteSensing	0.558	0.12	0.637	0.112	0.542
380	ExperimentActivity	RemoteSensing	0.153	0.063	0.435	0.072	0.151
381	Campaign	RemoteSensing	0.153	0.063	0.455	0.077	0.149
382	Correction	RemoteSensing	0.141	0.057	0.455	0.077	0.137
383	Difference	RemoteSensing	0.157	0.059	0.5	0.091	0.142
384	Hypothesis	RemoteSensing	0.002	0.063	0.2	0.059	0
385	Publication	RemoteSensing	0.35	0.086	0.6	0.112	0.319
386	Realization	RemoteSensing	0.489	0.085	0.667	0.125	0.458
387	Sample	RemoteSensing	0.3	0.067	0.6	0.112	0.268
388	Validation	RemoteSensing	0.132	0.053	0.435	0.072	0.13
389	proof	RemoteSensing	0.132	0.053	0.435	0.072	0.13
390	Observation	RemoteSensing	0.158	0.059	0.5	0.091	0.142
391	HumanActivity	Tomography	0	0	0.445	0.091	0
392	Research	Tomography	0	0	0.728	0.143	0
393	Analysis	Tomography	0	0	0.728	0.143	0

394	Investigation	Tomography	0	0	0.762	0.167	0
395	Monitoring	Tomography	0	0	0.667	0.112	0
396	Project	Tomography	0	0	0.477	0.084	0
397	ProofOfConcept	Tomography	0	0	0.584	0.091	0
398	ResearchSetting	Tomography	0	0	0.4	0.077	0
399	Residual	Tomography	0	0	0.445	0.091	0
400	Result	Tomography	0	0	0.236	0.072	0
401	Representation	Tomography	0	0	0.9	0.334	0
402	Variable	Tomography	0	0	0.25	0.077	0
403	Assesment	Tomography	0	0	0.737	0.167	0
404	Evidence	Tomography	0	0	0.527	0.1	0
405	Experiment	Tomography	0	0	0.435	0.072	0
406	ExperimentActivity	Tomography	0	0	0.667	0.112	0
407	Campaign	Tomography	0	0	0.696	0.125	0
408	Correction	Tomography	0	0	0.609	0.1	0
409	Difference	Tomography	0	0	0.572	0.1	0
410	Hypothesis	Tomography	0	0	0.477	0.084	0
411	Publication	Tomography	0	0	0.191	0.056	0
412	Realization	Tomography	0	0	0.455	0.077	0
413	Sample	Tomography	0	0	0.477	0.084	0
414	Validation	Tomography	0	0	0.584	0.091	0
415	Proof	Tomography	0	0	0.584	0.091	0
416	Observation	Tomography	0	0	0.762	0.167	0
417	HumanActivity	XrayDiffraction	0.004	0.057	0.25	0.077	0
418	Research	XrayDiffraction	0.001	0.054	0.2	0.059	0
419	Analysis	XrayDiffraction	0.001	0.052	0.2	0.059	0
420	Investigation	XrayDiffraction	0.002	0.057	0.211	0.063	0
421	Monitoring	XrayDiffraction	0.001	0.045	0.182	0.053	0
422	Project	XrayDiffraction	0.002	0.044	0.211	0.063	0
423	ProofOfConcept	XrayDiffraction	0.004	0.048	0.223	0.067	0
424	ResearchSetting	XrayDiffraction	0.004	0.046	0.223	0.067	0
425	Residual	XrayDiffraction	0.004	0.051	0.25	0.077	0
426	Result	XrayDiffraction	0.555	0.128	0.667	0.167	0.524
427	Representation	XrayDiffraction	0.004	0.046	0.223	0.067	0
428	Variable	XrayDiffraction	0.09	0.049	0.429	0.112	0.082
429	Assesment	XrayDiffraction	0.004	0.05	0.236	0.072	0
430	Evidence	XrayDiffraction	0.004	0.056	0.236	0.072	0
431	Experiment	XrayDiffraction	0.001	0.053	0.191	0.056	0
432	ExperimentActivity	XrayDiffraction	0.001	0.052	0.182	0.053	0
433	Campaign	XrayDiffraction	0.001	0.052	0.191	0.056	0
434	Correction	XrayDiffraction	0.001	0.048	0.191	0.056	0
435	Difference	XrayDiffraction	0.002	0.049	0.211	0.063	0
436	Hypothesis	XrayDiffraction	0.002	0.057	0.211	0.063	0
437	Publication	XrayDiffraction	0.114	0.068	0.316	0.072	0.11

438	Realization	XrayDiffraction	0.001	0.045	0.2	0.059	0
439	Sample	XrayDiffraction	0.002	0.048	0.211	0.063	0
440	Validation	XrayDiffraction	0.001	0.045	0.182	0.053	0
441	Proof	XrayDiffraction	0.001	0.045	0.182	0.053	0
442	Observation	XrayDiffraction	0.002	0.05	0.211	0.063	0

Appendix III: OWL Ontology on HumanResearch

```
>?xml
version="1.0"
encoding="
UTF-8"?>
<!DOCTYPErdf:RDF [
<!ENTITYrepr"http://sweet.jpl.nasa.gov/2.3/repr.owl">
<!ENTITYhuma"http://sweet.jpl.nasa.gov/2.3/human.owl">
<!ENTITYres"http://sweet.jpl.nasa.gov/2.3/humanResearch.o
wl">
<!ENTITYowl"http://www.w3.org/2002/07/owl#">
<!ENTITYrdf"http://www.w3.org/1999/02/22-rdf-syntax-
ns#">
<!ENTITYrdfs"http://www.w3.org/2000/01/rdf-schema#">
<!ENTITYxsd"http://www.w3.org/2001/XMLSchema#">
]>
<rdf:RDFxml:base="&res;"
xmlns:res="&res;"
xmlns:repr="&repr;"
xmlns:huma="&huma;"
xmlns:owl="&owl;"
xmlns:rdf="&rdf;"
xmlns:rdfs="&rdfs;"
          xmlns:xsd = "&xsd;">
<!-- Ontology Information -->
<owl:Ontologyrdf:about=""
owl:versionInfo="2.3">
<rdfs:label>SWEET Ontology</rdfs:label>
<owl:importsrdf:resource="&huma;"/>
<owl:importsrdf:resource="&repr;"/>
</owl:Ontology>
<!-- Experiment -->
<owl:Classrdf:about="#Analysis">
<rdfs:subClassOfrdf:resource="#Research"/>
</owl:Class>
<owl:Classrdf:about="#AppliedResearch">
<rdfs:subClassOfrdf:resource="#Research"/>
</owl:Class>
<owl:Classrdf:about="#Assessment">
<rdfs:subClassOfrdf:resource="#Investigation"/>
</owl:Class>
<owl:Classrdf:about="#Campaign">
<rdfs:subClassOfrdf:resource="#ExperimentActivity"/>
</owl:Class>
<owl:Classrdf:about="#Correction">
<rdfs:subClassOfrdf:resource="#ExperimentActivity"/>
</owl:Class>
<owl:Classrdf:about="#Difference">
<rdfs:subClassOfrdf:resource="#ExperimentActivity"/>
```

```
</owl:Class>
<owl:Classrdf:about="#Evidence">
<rdfs:subClassOfrdf:resource="#Assessment"/>
</owl:Class>
<owl:Classrdf:about="#Experiment">
<rdfs:subClassOfrdf:resource="#Investigation"/>
</owl:Class>
<owl:Classrdf:about="#ExperimentActivity">
<rdfs:subClassOfrdf:resource="#Experiment"/>
</owl:Class>
<owl:Classrdf:about="#HypothesisTest">
<rdfs:subClassOfrdf:resource="#ExperimentActivity"/>
</owl:Class>
<owl:Classrdf:about="#Investigation">
<rdfs:subClassOfrdf:resource="#Research"/>
<owl:equivalentClassrdf:resource="#Investigate"/>
</owl:Class>
<owl:Classrdf:about="#Investigate"/>
<owl:Classrdf:about="#Monitor">
<rdfs:subClassOfrdf:resource="#Research"/>
<owl:equivalentClassrdf:resource="#Monitoring"/>
</owl:Class>
<owl:Classrdf:about="#Monitoring"/>
<owl:Classrdf:about="#Observation">
<rdfs:subClassOfrdf:resource="#Investigation"/>
<owl:equivalentClassrdf:resource="#Observe"/>
</owl:Class>
<owl:Classrdf:about="#Observe"/>
<owl:Classrdf:about="#Project">
<rdfs:subClassOfrdf:resource="#Research"/>
</owl:Class>
<owl:Classrdf:about="#Proof">
<rdfs:subClassOfrdf:resource="#Validation"/>
</owl:Class>
<owl:Classrdf:about="#Publication">
<rdfs:subClassOfrdf:resource="#Research"/>
</owl:Class>
<owl:Classrdf:about="#Realization">
<rdfs:subClassOfrdf:resource="#ExperimentActivity"/>
</owl:Class>
<owl:Classrdf:about="#Research">
<rdfs:subClassOfrdf:resource="&huma;#HumanActivity"/>
</owl:Class>
<owl:Classrdf:about="#Residual">
<rdfs:subClassOfrdf:resource="#Research"/>
</owl:Class>
<owl:Classrdf:about="#Result">
<rdfs:subClassOfrdf:resource="#Research"/>
</owl:Class>
<owl:Classrdf:about="#Sample">
```

```
<rdfs:subClassOfrdf:resource="#ExperimentActivity"/>
</owl:Class>
<owl:Classrdf:about="#Validation">
<rdfs:subClassOfrdf:resource="#ExperimentActivity"/>
</owl:Class>
<owl:Classrdf:about="#Variable">
<rdfs:subClassOfrdf:resource="&repr;#Representation"/>
</owl:Class>
<owl:Classrdf:about="#WeightOfEvidence">
<rdfs:subClassOfrdf:resource="#Assessment"/>
</owl:Class>
<!-- Research environments -->
<owl:Classrdf:about="#ResearchSetting">
<rdfs:subClassOfrdf:resource="#Research"/>
</owl:Class>
<owl:Classrdf:about="#ProofOfConcept">
<rdfs:subClassOfrdf:resource="#Proof"/>
<rdfs:subClassOfrdf:resource="#Research"/>
</owl:Class>
<owl:Classrdf:about="#EndToEndEnvironment">
<rdfs:subClassOfrdf:resource="#ResearchSetting"/>
<owl:disjointWithrdf:resource="#LaboratoryEnvironment" />
</owl:Class>
<owl:Classrdf:about="#LaboratoryEnvironment">
<rdfs:subClassOfrdf:resource="#ResearchSetting"/>
</owl:Class>
<owl:Classrdf:about="#MissionTestedEnvironment">
<rdfs:subClassOfrdf:resource="#ResearchSetting"/>
<owl:disjointWithrdf:resource="#LaboratoryEnvironment" />
</owl:Class>
<owl:Classrdf:about="#OperationalEnvironment">
<rdfs:subClassOfrdf:resource="#ResearchSetting"/>
<owl:disjointWithrdf:resource="#LaboratoryEnvironment" />
</owl:Class>
<owl:Classrdf:about="#RepresentativeEnvironment">
<rdfs:subClassOfrdf:resource="#ResearchSetting"/>
<owl:disjointWithrdf:resource="#LaboratoryEnvironment" />
</owl:Class>
<owl:Classrdf:about="#VerifiedValidatedEnvironment">
<rdfs:subClassOfrdf:resource="#ResearchSetting"/>
<owl:disjointWithrdf:resource="#LaboratoryEnvironment" />
</owl:Class>
</rdf:RDF>
```

Appendix IV: OWL Ontology on SciMethodology

```
 >?xml
 version="1.0"
 encoding="
TF-8"?>
 <!DOCTYPErdf:RDF [
 <!ENTITYres"http://sweet.jpl.nasa.gov/2.3/humanResearch.owl">
 <!ENTITYmeth"http://sweet.jpl.nasa.gov/2.3/reprSciMethodology.owl#">
 <!ENTITYowl"http://www.w3.org/2002/07/owl#">
 <!ENTITYrdf"http://www.w3.org/1999/02/22-rdf-syntax-ns#">
 <!ENTITYrdfs"http://www.w3.org/2000/01/rdf-schema#">
 <!ENTITYxsd"http://www.w3.org/2001/XMLSchema#">
 ]>
 <rdf:RDFxml:base="&meth;"
 xmlns:res="&res;"
 xmlns:meth="&meth;"
 xmlns:owl="&owl;"
 xmlns:rdf="&rdf;"
 xmlns:rdfs="&rdfs;"
           xmlns:xsd = "&xsd;">
 <!-- Ontology Information -->
 <owl:Ontologyrdf:about=""
 owl:versionInfo="2.3">
 <rdfs:label>SWEET Ontology</rdfs:label>
 <owl:importsrdf:resource="&res;"/>
 </owl:Ontology>
 <owl:Classrdf:about="#ResearchExploration">
 <rdfs:subClassOfrdf:resource="&res;#Research"/>
 </owl:Class>
 <owl:Classrdf:about="#Altimetry">
 <rdfs:subClassOfrdf:resource="#Methodology" />
 </owl:Class>
 <owl:Classrdf:about="#CarbonDating">
 <rdfs:subClassOfrdf:resource="#Methodology" />
 </owl:Class>
 <owl:Classrdf:about="#Interferometry">
 <rdfs:subClassOfrdf:resource="#Methodology" />
 </owl:Class>
 <owl:Classrdf:about="#IsotopeAnalysis">
 <rdfs:subClassOfrdf:resource="#Methodology" />
 </owl:Class>
 <owl:Classrdf:about="#Methodology">
 <rdfs:subClassOfrdf:resource="&res;#ExperimentActivity" />
 </owl:Class>
 <owl:Classrdf:about="#Photometry">
 <rdfs:subClassOfrdf:resource="#Methodology" />
 </owl:Class>
 <owl:Classrdf:about="#Polarimetry">
 <rdfs:subClassOfrdf:resource="#Methodology" />
 </owl:Class>
```

```
<owl:Classrdf:about="#RadioactiveDating">
<rdfs:subClassOfrdf:resource="#Methodology" />
</owl:Class>
<owl:Classrdf:about="#Spectroscopy">
<rdfs:subClassOfrdf:resource="#Methodology" />
</owl:Class>
<owl:Classrdf:about="#RetrievalApproach">
<rdfs:subClassOfrdf:resource="#ResearchExploration"/>
</owl:Class>
<owl:Classrdf:about="#Imaging">
<rdfs:subClassOfrdf:resource="#ResearchExploration"/>
</owl:Class>
<owl:Classrdf:about="#Photography">
<rdfs:subClassOfrdf:resource="#ResearchExploration"/>
</owl:Class>
<owl:Classrdf:about="#AerialPhotography">
<rdfs:subClassOfrdf:resource="#Photography"/>
</owl:Class>
<owl:Classrdf:about="#RemoteSensing">
<rdfs:subClassOfrdf:resource="#ResearchExploration"/>
</owl:Class>
<owl:Classrdf:about="#Tomography">
<rdfs:subClassOfrdf:resource="#ResearchExploration"/>
</owl:Class>
<owl:Classrdf:about="#XRayDiffraction">
<rdfs:subClassOfrdf:resource="#ResearchExploration"/>
</owl:Class>
<owl:Classrdf:about="#Optics">
<rdfs:subClassOfrdf:resource="#ResearchExploration"/>
<rdfs:comment>Optics is a branch of physics that describes the behavior and properties
of light and the interaction of light with matter. Optics explains optical
phenomena</rdfs:comment>
</owl:Class>
<owl:Classrdf:about="#Engineering">
<rdfs:subClassOfrdf:resource="#ResearchExploration"/>
</owl:Class>
<meth:RetrievalApproachrdf:about="#Passive">
</meth:RetrievalApproach>
<meth:RetrievalApproachrdf:about="#Active">
</meth:RetrievalApproach>
</rdf:RDF>
```

Appendix V: Java Implementation of STC with CARROT2

```
package com.adebisi.carrot2;
import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;
import org.carrot2.clustering.stc.STCClusteringAlgorithm;
import org.carrot2.core.Controller;
import org.carrot2.core.ControllerFactory;
import org.carrot2.core.ProcessingResult;
import org.carrot2.source.google.GoogleDocumentSource;
import org.carrot2.source.google.GoogleDocumentSourceDescriptor;
import com.adebisi.carrot2.config.GoogleConfig;
import com.adebisi.carrot2.formatting.ConsoleFormatter;
public class Carrot2 {
        static Scanner scanner = null;

        static
        {
                scanner = new Scanner(System.in);
        }

        public static void main(String[] args)
        {
                System.out.print("Enter the search query: ");
                String search_term = scanner.nextLine();
                final Controller controller = ControllerFactory.createSimple();
                final Map<String, Object> attributes = new HashMap<String,
Object>();

                attributes.put(GoogleDocumentSourceDescriptor.Keys.QUERY,
search_term);
                attributes.put(GoogleDocumentSourceDescriptor.Keys.RESULTS,
GoogleConfig.RESULTS_MAX_NUM);
                attributes.put(GoogleDocumentSourceDescriptor.Keys.START, 0);
                GoogleDocumentSourceDescriptor.attributeBuilder(attributes);

                //final ProcessingResult result_byfield = controller.process(attributes,
GoogleDocumentSource.class, ByFieldClusteringAlgorithm.class);
                //final ProcessingResult result_lingo = controller.process(attributes,
GoogleDocumentSource.class, LingoClusteringAlgorithm.class);
                final ProcessingResult result_stc = controller.process(attributes,
GoogleDocumentSource.class, STCClusteringAlgorithm.class);
```

```
//                final ProcessingResult results_kmeans = controller.process(attributes,
GoogleDocumentSource.class, BisectingKMeansClusteringAlgorithm.class);

                //ConsoleFormatter.displayResults(result_byfield);
      //ConsoleFormatter.displayResults(result_lingo);
       ConsoleFormatter.displayResults(result_stc);
//       ConsoleFormatter.displayResults(results_kmeans);

                cleanUp();
        }

        public static void cleanUp()
        {
                scanner.close();
        }
}

/*


package com.adebisi.carrot2.formatting;

import java.text.NumberFormat;
import java.util.Collection;
import java.util.Map;

import org.apache.commons.lang.StringUtils;
import org.carrot2.core.Cluster;
import org.carrot2.core.Document;
import org.carrot2.core.ProcessingResult;
import org.carrot2.core.attribute.CommonAttributesDescriptor;

/**
 * Simple console formatter for dumping {@link ProcessingResult}.
 */
public class ConsoleFormatter
{
   public static void displayResults(ProcessingResult processingResult)
   {
      final Collection<Document> documents = processingResult.getDocuments();
      final Collection<Cluster> clusters = processingResult.getClusters();
      final Map<String, Object> attributes = processingResult.getAttributes();

      // Show documents
      if (documents != null)
      {
         //displayDocuments(documents);
      }

      // Show clusters
```

```
        if (clusters != null)
        {
            displayClusters(clusters);
        }

        // Show attributes other attributes
        displayAttributes(attributes);
    }

    public static void displayDocuments(final Collection<Document> documents)
    {
        System.out.println("Collected " + documents.size() + " documents\n");
        for (final Document document : documents)
        {
            displayDocument(0, document);
        }
    }

    public static void displayAttributes(final Map<String, Object> attributes)
    {
        System.out.println("Attributes:");

        String                    DOCUMENTS_ATTRIBUTE                    =
CommonAttributesDescriptor.Keys.DOCUMENTS;
        String                    CLUSTERS_ATTRIBUTE                     =
CommonAttributesDescriptor.Keys.CLUSTERS;
        for (final Map.Entry<String, Object> attribute : attributes.entrySet())
        {
            if (!DOCUMENTS_ATTRIBUTE.equals(attribute.getKey())
                && !CLUSTERS_ATTRIBUTE.equals(attribute.getKey()))
            {
                System.out.println(attribute.getKey() + ":   " + attribute.getValue());
            }
        }
    }

    public static void displayClusters(final Collection<Cluster> clusters)
    {
        displayClusters(clusters, Integer.MAX_VALUE);
    }

    public static void displayClusters(final Collection<Cluster> clusters,
        int maxNumberOfDocumentsToShow)
    {
        displayClusters(clusters, maxNumberOfDocumentsToShow,
            ClusterDetailsFormatter.INSTANCE);
    }

    public static void displayClusters(final Collection<Cluster> clusters,
    int maxNumberOfDocumentsToShow, ClusterDetailsFormatter
clusterDetailsFormatter)
  {
    System.out.println("\n\nCreated " + clusters.size() + " clusters\n");
    int clusterNumber = 1;
    for (final Cluster cluster : clusters)
    {
      displayCluster(0, "" + clusterNumber++, cluster,
maxNumberOfDocumentsToShow,
        clusterDetailsFormatter);
    }
  }

  private static void displayDocument(final int level, Document document)
  {
    final String indent = getIndent(level);

    System.out.printf(indent + "[%2s] ", document.getStringId());
    System.out.println(document.getField(Document.TITLE));
    final String url = document.getField(Document.CONTENT_URL);
    final String summary = document.getField(Document.SUMMARY);
    if (StringUtils.isNotBlank(url))
    {
      System.out.println(indent + "     " + url);
    }
    if (StringUtils.isNotBlank(summary))
    {
      System.out.println(indent + "     " + summary);
    }
    System.out.println();
  }

  private static void displayCluster(final int level, String tag, Cluster cluster,
    int maxNumberOfDocumentsToShow, ClusterDetailsFormatter
clusterDetailsFormatter)
  {
    final String label = cluster.getLabel();

    // indent up to level and display this cluster's description phrase
    for (int i = 0; i < level; i++)
    {
      System.out.print("  ");
    }
    System.out.println(label + "  "
      + clusterDetailsFormatter.formatClusterDetails(cluster));

    // if this cluster has documents, display three topmost documents.
    int documentsShown = 0;
    for (final Document document : cluster.getDocuments())
    {
      if (documentsShown >= maxNumberOfDocumentsToShow)
      {
        break;
      }
      displayDocument(level + 1, document);
      documentsShown++;
    }
    if (maxNumberOfDocumentsToShow > 0
      && (cluster.getDocuments().size() > documentsShown))
    {
      System.out.println(getIndent(level + 1) + "... and "
        + (cluster.getDocuments().size() - documentsShown) + " more\n");
    }

    // finally, if this cluster has subclusters, descend into recursion.
    final int num = 1;
    for (final Cluster subcluster : cluster.getSubclusters())
    {
      displayCluster(level + 1, tag + "." + num, subcluster,
        maxNumberOfDocumentsToShow, clusterDetailsFormatter);
    }
  }

  private static String getIndent(final int level)
  {
    final StringBuilder indent = new StringBuilder();
    for (int i = 0; i < level; i++)
    {
      indent.append("  ");
    }

    return indent.toString();
  }

  public static class ClusterDetailsFormatter
  {
    public    final    static    ClusterDetailsFormatter    INSTANCE    =    new
ClusterDetailsFormatter();

    protected NumberFormat numberFormat;

    public ClusterDetailsFormatter()
    {
      numberFormat = NumberFormat.getInstance();
      numberFormat.setMaximumFractionDigits(2);
    }

    public String formatClusterDetails(Cluster cluster)
    {
      final Double score = cluster.getScore();
```

```
            return "(" + cluster.getAllDocuments().size() + " docs"
               + (score != null ? ", score: " + numberFormat.format(score) : "") + ")";
        }
    }
}

package com.adebisi.carrot2.config;

public class GoogleConfig {

            public static String API_KEY = "";

            public static int RESULTS_MAX_NUM = 100;
}
```

Appendix VI: Python Implementation of Feature Extraction in MDSs

```
# -*- coding: cp1252 -*-
#import csv
#!/usr/bin/env python
# -*- coding: cp1252 -*-
from __future__ import print_function
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.metrics.pairwise import linear_kernel
from sklearn.metrics.pairwise import rbf_kernel
from sklearn.metrics.pairwise import euclidean_distances
from sklearn.feature_extraction.stop_words import ENGLISH_STOP_WORDS
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.corpus import wordnet
from pprint import pprint
import csv
documents = (
"""
DA1
""",
"""
DA2
""",
 """
 DA3
""",
"""
DA4
""",
"""
DB1
 ...
""",
"""
DB2
""",
"""
DB3
""",
"""
DB4
""",
)
wnl = WordNetLemmatizer()
filtered_documents = []
stop_words_count = []
# the number of stop words in the documents
for document in documents:
    document_tokens = nltk.word_tokenize(document)
    #extracts the tokens from the document
    print(' '.join(document_tokens))
                stop_words = [token.lower() for token in document_tokens if token.lower()
            in ENGLISH_STOP_WORDS]
    # we get the actual stop words found -- temporarily -- we only need it for counting
    stop_words_count.append(len(stop_words))
    # we add the count [or number] of stop words for the document
    filtered_tokens = [token.lower() for token in document_tokens if token.lower() not
in ENGLISH_STOP_WORDS]
    # we strip off the stop words as usual
    print(' '.join(filtered_tokens))
    print('')
    print('STOP WORD COUNT: #', len(stop_words))
    print('')
    lemmalized = []
    for token in filtered_tokens:
        if token.isalpha():
            token_as_verb = wnl.lemmatize(token.lower(), pos=wordnet.VERB)
            lemmalized.append(wnl.lemmatize(token_as_verb.lower()))
    print(' '.join(lemmalized))
    print('')
    filtered_documents.append(' '.join(lemmalized))
print('*'*8,'FILTERED DOCUMENTS','*'*8)
pprint(filtered_documents)
print('')

tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix,                    tfidfext_matrix                    =
tfidf_vectorizer.fit_transform(tuple(filtered_documents))
feature_names = tfidf_vectorizer.get_feature_names()
#get the feature names from the vectorizer

print('*'*8,'FEATURE NAMES','*'*8)
print(feature_names)
print('')

print('*'*8,'MATRIX ALL VALUES','*'*8)
print(tfidf_matrix)
print('')

print('*'*8,'FEATURE INDEXES','*'*8)
for index in range(0, len(feature_names)):
    print('{0}: {1}'.format(index, feature_names[index]))
print('')

print('*'*8,'TFIDF SCORES: DEFAULT FORMULA','*'*8)
print('')
t_rows, t_cols = tfidf_matrix.shape
print('(x,y)', end='\t')
for index in range(0, len(feature_names)):
    print('{0:^5d}'.format(index), end='\t')
print('')
for row in range(0, t_rows):
    print('{:^5d}'.format(row), end='\t')
    for column in range(0, t_cols):
        t_val = tfidf_matrix[row, column]
        print('{:^7.4f}'.format(t_val), end='\t')
    print('')
print('')

print('*'*8,'TFIDF SCORES: EXTENDED FORMULA','*'*8)
print('')
t_rows, t_cols = tfidfext_matrix.shape
print('')
print('(x,y)', end='\t')
for index in range(0, len(feature_names)):
    print('{0:^5d}'.format(index), end='\t')
print('')
for row in range(0, t_rows):
    print('{:^5d}'.format(row), end='\t')
    for column in range(0, t_cols):
        t_val = tfidfext_matrix[row, column]
        print('{:^7.4f}'.format(t_val), end='\t')
    print('')
print('')

print('*'*8,'COSINE SIMILARITY -- USING DEFAULT TFIDF MATRIX','*'*8)
row1 = list()
for row in tfidf_matrix:
    c_sim = cosine_similarity(row, tfidf_matrix)
    print(c_sim)

    row1.append(c_sim)

print('')

print('*'*8,'COSINE SIMILARITY -- USING EXTENDED TFIDF MATRIX','*'*8)
row2 = list()
for row in tfidfext_matrix:
    c_sim = cosine_similarity(row, tfidfext_matrix)
    print(c_sim)

    row2.append(c_sim)

print('')

print('*'*8,'LINEAR KERNEL DISTANCE SIMILARITY -- USING DEFAULT
TFIDF MATRIX','*'*8)
row3 = list()
for row in tfidf_matrix:
    c_sim = linear_kernel(row, tfidf_matrix)
    print(c_sim)
    row3.append(c_sim)

print('')

print('*'*8,'LINEAR KERNEL DISTANCE SIMILARITY -- USING EXTENDED TFIDF MATRIX','*'*8)
row4 = list()
for row in tfidfext_matrix:
    c_sim = linear_kernel(row, tfidfext_matrix)
    print(c_sim)
    row4.append(c_sim)

print('')

print('*'*8,'RBF (GAUSSIAN) KERNEL DISTANCE SIMILARITY -- USING DEFAULT TFIDF MATRIX','*'*8)
row5 = list()
for row in tfidf_matrix:
    c_sim = rbf_kernel(row, tfidf_matrix)
    print(c_sim)
    row5.append(c_sim)
print('')

print('*'*8,'RBF (GAUSSIAN) KERNEL DISTANCE SIMILARITY -- USING EXTENDED TFIDF MATRIX','*'*8)
row6 = list()
for row in tfidfext_matrix:
    c_sim = rbf_kernel(row, tfidfext_matrix)
    print(c_sim)
    row6.append(c_sim)
print('')

print('*'*8,'EUCLIDEAN DISTANCE SIMILARITY -- USING DEFAULT TFIDF MATRIX','*'*8)
row7 = list()
for row in tfidf_matrix:
    c_sim = euclidean_distances(row, tfidf_matrix)
    print(c_sim)
    row7.append(c_sim)
print('')

print('*'*8,'EUCLIDEAN DISTANCE SIMILARITY -- USING EXTENDED TFIDF MATRIX','*'*8)
row8 = list()
for row in tfidfext_matrix:
    c_sim = euclidean_distances(row, tfidfext_matrix)
    print(c_sim)
    row8.append(c_sim)
```

```
print('')
fd = open('result.csv', 'wb')
try:
    writer = csv.writer(fd,dialect='excel',quotechar='"',quoting=csv.QUOTE_ALL)
    rowHeader                                                                                                  =
[['d1_d1','d1_d2','d1_d3','d1_d4','d1_d5','d1_d6','d1_d7','d1_d8','d2_d1','d2_d2','d2_d3',
'd2_d4','d2_d5','d2_d6','d2_d7','d2_d8','d3_d1','d3_d2','d3_d3','d3_d4','d3_d5','d3_d6','d
3_d7','d3_d8','d4_d1','d4_d2','d4_d3','d4_d4','d4_d5','d4_d6','d4_d7','d4_d8','d5_d1','d5
_d2','d5_d3','d5_d4','d5_d5','d5_d6','d5_d7','d5_d8','d6_d1','d6_d2','d6_d3','d6_d4','d6_
d5','d6_d6','d6_d7','d7_d8','d7_d1','d7_d2','d7_d3','d7_d4','d7_d5','d7_d6','d7_d7','d7_d
8','d8_d1','d8_d2','d8_d3','d8_d4','d8_d5','d8_d6','d8_d7','d8_d8']]
    writer.writerows(rowHeader)
         writer.writerows(["Cosine"])
    #writer.writerows(row1)
    #First Cosine Generator
    ls_row1 = []
    for r1 in row1:
        data1 = []
        for mydata1 in r1:
            #data1.append(mydata1)
            for simpledata in mydata1:
                ls_row1.append(simpledata)
                #print(simpledata)
    #print(len(ls_row1))
    writer.writerows([ls_row1])
    #writer.writerows(ls_row1)
    #End Cosine Generator
    writer.writerows(["CW-Cosine "])
    #Second Cosine Generator
    ls_row2 = []
    for r2 in row2:
        data2 = []
        for mydata2 in r2:
            #data1.append(mydata1)
            for simpledata2 in mydata2:
                ls_row2.append(simpledata2)
    writer.writerows([ls_row2])
    #End Cosine Generator
    writer.writerows(["LINEAR KERNEL"])
    #LINEAR Line Generator
    ls_row3 = []
    for r3 in row3:
        data3 = []
        for mydata3 in r3:
            #data1.append(mydata1)
            for simpledata3 in mydata3:
                ls_row3.append(simpledata3)
    writer.writerows([ls_row3])
    #End LINEAR Generator
    writer.writerows(["CW-LINEAR KERNEL"])
```

```
      #LINEAR Line Generator
ls_row4 = []
for r4 in row4:
    data4 = []
    for mydata4 in r4:
        #data1.append(mydata1)
        for simpledata4 in mydata4:
            ls_row4.append(simpledata4)
writer.writerows([ls_row4])
#End LINEAR Generator
writer.writerows(["RBF"])
    #RBF Line Generator
ls_row5 = []
for r5 in row5:
    data5 = []
    for mydata5 in r5:
        #data1.append(mydata1)
        for simpledata5 in mydata5:
            ls_row5.append(simpledata5)
writer.writerows([ls_row5])
#End RBF Generator
writer.writerows(["CW-RBF"])
     #CoRBF Line Generator
ls_row6 = []
for r6 in row6:
    data6 = []
    for mydata6 in r6:
        #data1.append(mydata1)
        for simpledata6 in mydata6:
            ls_row6.append(simpledata6)
writer.writerows([ls_row6])
#End RBF Generator
writer.writerows(["EUCLIDEAN"])
     #EUCLIDEAN  Line Generator
ls_row7 = []
for r7 in row7:
    data7 = []
    for mydata7 in r7:
        #data1.append(mydata1)
        for simpledata7 in mydata7:
            ls_row7.append(simpledata7)
writer.writerows([ls_row7])
#End EUCLIDEAN Generator
writer.writerows(["CW-EUCLIDEAN"])
    #CW-EUCLIDEAN Line Generator
ls_row8 = []
for r8 in row8:
    data8 = []
    for mydata8 in r8:
        #data1.append(mydata1)
```

```
            for simpledata8 in mydata8:
                ls_row8.append(simpledata8)
    writer.writerows([ls_row8])
    #End EUCLIDEAN Generator
   finally:
    fd.close()
print("Excel / CSV File Generated for this Operation")
raw_input('Press Enter to exit...')
```

Appendix VII: Java Implementation of WordNetSimilarity

```
package com.adebisi.wordnetsimilarity;
import java.util.Scanner;
import java.util.Set;
import com.adebisi.wordnetsimilarity.impl.Lipa;
import edu.cmu.lti.jawjaw.pobj.POS;
import edu.cmu.lti.lexical_db.ILexicalDatabase;
import edu.cmu.lti.lexical_db.NictWordNet;
import edu.cmu.lti.lexical_db.data.Concept;
import edu.cmu.lti.ws4j.RelatednessCalculator;
import edu.cmu.lti.ws4j.WS4J;
import edu.cmu.lti.ws4j.impl.HirstStOnge;
import edu.cmu.lti.ws4j.impl.JiangConrath;
import edu.cmu.lti.ws4j.impl.LeacockChodorow;
import edu.cmu.lti.ws4j.impl.Lin;
import edu.cmu.lti.ws4j.impl.Path;
import edu.cmu.lti.ws4j.impl.Resnik;
import edu.cmu.lti.ws4j.impl.WuPalmer;
import edu.cmu.lti.ws4j.util.WS4JConfiguration;

public class Application {

        private Concept concept1;
        private Concept concept2;

        private ILexicalDatabase db = new NictWordNet();
        private RelatednessCalculator[] measures = {
                        new HirstStOnge(db),
                        new LeacockChodorow(db),
                        new Resnik(db),
                        new JiangConrath(db),
                        new Lin(db),
                        new Path(db),
                        new WuPalmer(db),
                        new Lipa(db)
        };

        public static void main(String[] args) {
                Application app = new Application();
                app.run();
        }

        public Application() {

        }

        public void run() {
                Scanner input = new Scanner(System.in);
                this.printWelcome();
                this.concept1 = this.readTerm(input, "Please enter the first word:");
                Console.inform("Your word 1 is: "+this.concept1.getSynset());
                this.concept2 = this.readTerm(input, "Please enter the second word:");
                Console.inform("Your word 2 is: "+this.concept2.getSynset());
                // this.getConceptSimilarities(this.concept1, this.concept2);
                this.getWordSimilarities(this.concept1.getSynset(),
this.concept2.getSynset());
        }

        /**
         *
         * @param word1
         * @param word2
         */
        public void getWordSimilarities(String word1, String word2) {
                WS4JConfiguration.getInstance().setMFS(true);
                for (RelatednessCalculator rc : this.measures) {
                        try {
                            double score = rc.calcRelatednessOfWords(word1,
word2);
                            Console.inform(rc.getClass().getSimpleName()+" =>
"+score);
                        } catch (NullPointerException e) {
                            Console.inform("Error: "+e.getMessage());
                        }
                }
        }

        /**
         * Displays the similarity scores between the provided concepts
         *
         * @param concept1
         * @param concept2
         */
        public void getConceptSimilarities(Concept concept1, Concept concept2) {
                WS4JConfiguration.getInstance().setMFS(true);
                for (RelatednessCalculator rc : this.measures) {
                        try {
                            double score = rc.calcRelatednessOfSynset(concept1,
concept2).getScore();
                            Console.inform(rc.getClass().getSimpleName()+" =>
"+score);
                        } catch (NullPointerException e) {
                            Console.inform("Error: "+e.getMessage());
                        }
                }
        }

        /**
         * Prints the welcome message to the console
         */
        private void printWelcome() {
                System.out.println("Welcome to WordnetSimilarity app built in Java");
                System.out.println("you will be guided through the process of entering
your terms.");
                System.out.println(" ");
        }

        /**
         * Performs the process of taking a word
         *
         * @param in
         * @param message
         * @return
         */
        private Concept readTerm(Scanner in, String message) {
                String word = this.readTerm(in, message, false);
                return new Concept(word, POS.n);
        }

        /**
         * Performs the process of taking a word and drilling down to a specific
definition, if required
         *
         * @param in
         * @param message
         * @param showDefinitions
         * @return
         */
        private String readTerm(Scanner in, String message, boolean showDefinitions)
{
                String word = Console.ask(in, message, true);
                String definition = "";
                if (showDefinitions) {
                        Set<String> definitions = WS4J.findDefinitions(word, POS.n);
                        Console.inform("Which of these definitions describes the word
you're interested in (enter the number)?");
                        Console.list(definitions);
                        definition = Console.ask(in, ">", true);
                }
                return !showDefinitions ? word : word+"#n#"+definition;
        }
}
package com.adebisi.wordnetsimilarity;

import java.util.Scanner;
import java.util.Set;

public class Console {
        /**
```

```
 * Requests input from the user on the console
 *
 * @param in
 * @return
 */
public static String ask(Scanner in) {
	return ask(in, "> ");
}

/**
 * Requests input from the user on the console with the specified message
 *
 * @param in
 * @param message
 * @return
 */
public static String ask(Scanner in, String message) {
	return ask(in, message, false);
}

/**
 *
 * @param in
 * @param message
 * @param onOneLine
 * @return
 */
public static String ask(Scanner in, String message, boolean onOneLine) {
	inform(message, !onOneLine);
	return in.next();
}

/**
 * Prints a message to the console
 *
 * @param message
 */
public static void inform(String message) {
	System.out.println(message);
}

/**
 *
 * @param message
 * @param useNewLine
 */
public static void inform(String message, boolean useNewLine) {
	if (useNewLine) {
		System.out.println(message);
	} else {
```

```
            System.out.print(message+" ");
        }
    }

    /**
     * Displays a list of items on the console
     *
     * @param strings
     */
    public static void list(Set<String> strings) {
        int s = strings.size();
        String[] stringList = new String[s];
        strings.toArray(stringList);
        int optionNumber = 1;
        for (int i = 0; i < s; i++) {
            inform("["+optionNumber+"] " + stringList[i]);
            ++optionNumber;
        }
    }
}

package com.adebisi.wordnetsimilarity.impl;

import java.util.ArrayList;
import java.util.List;

import edu.cmu.lti.jawjaw.pobj.POS;
import edu.cmu.lti.lexical_db.ILexicalDatabase;
import edu.cmu.lti.lexical_db.data.Concept;
import edu.cmu.lti.ws4j.Relatedness;
import edu.cmu.lti.ws4j.RelatednessCalculator;
import edu.cmu.lti.ws4j.util.ICFinder;
import edu.cmu.lti.ws4j.util.PathFinder.Subsumer;

public class Lipa extends RelatednessCalculator {

    protected static double min = 0; // or -Double.MAX_VALUE ?
    protected static double max = 1;

    protected static double k = 0.5; // our special constant

    @SuppressWarnings("serial")
    private static List<POS[]> posPairs = new ArrayList<POS[]>(){{
        add(new POS[]{POS.n,POS.n});
        add(new POS[]{POS.v,POS.v});
    }};

    public Lipa(ILexicalDatabase db) {
        super(db);
    }
    @Override
    protected Relatedness calcRelatedness(Concept synset1, Concept synset2) {
        StringBuilder tracer = new StringBuilder();
        if (synset1 == null || synset2 == null) return new Relatedness(min, null,
illegalSynset);
        if (synset1.getSynset().equals(synset2.getSynset())) return new
Relatedness(max, identicalSynset, null);

        StringBuilder subTracer = enableTrace ? new StringBuilder() : null;
        List<Subsumer> lcsList =
ICFinder.getInstance().getLCSbyIC(pathFinder, synset1, synset2, subTracer);
        if ( lcsList.size() == 0 ) return new Relatedness(min, tracer.toString(),
null);

        Concept lcsConcept = lcsList.get(0).subsumer;
        StringBuilder tracerPaths = new StringBuilder();

        List<Subsumer> shortestPath1 = pathFinder.getShortestPaths(synset1,
lcsConcept, tracerPaths);
        // we get the shortest path between synset1 and the LCS
        List<Subsumer> shortestPath2 = pathFinder.getShortestPaths(synset2,
lcsConcept, tracerPaths);
        // we get the shortest path between synset2 and the LCS
        int maxDistance = shortestPath1.size() > shortestPath2.size() ?
shortestPath1.size() : shortestPath2.size();
        // we use the longest path length

        double ic1 = ICFinder.getInstance().ic(pathFinder, synset1);
        double ic2 = ICFinder.getInstance().ic(pathFinder, synset2);
        double score = ( ic1>0 && ic2>0 )
                ? ((2D * lcsList.get(0).ic / ( ic1 + ic2 )) + Math.pow((1D
- k), maxDistance))
                : 0D;

        if ( enableTrace ) {
            tracer.append(subTracer.toString());
            for ( Subsumer lcs : lcsList ) {
                tracer.append("Lowest Common Subsumer(s): ");
                tracer.append(db.conceptToString(
lcs.subsumer.getSynset() )+" (IC="+lcs.ic+")\n");
            }
            tracer.append("Concept1:
"+db.conceptToString(synset1.getSynset())+" (IC="+ic1+")\n");
            tracer.append("Concept2:
"+db.conceptToString(synset2.getSynset())+" (IC="+ic2+")\n");
        }
        return new Relatedness( score, tracer.toString(), null );
    }
```

```
        @Override
        public List<POS[]> getPOSPairs() {
                return posPairs;
        }

}
```