EXTENDED REFINEMENT METHODOLOGY FOR AUTOMATIC KEYPHRASE ASSIGNMENT

by

Rabia Irfan
MSIT-10

Supervisor

Dr. Sharifullah Khan

Contents

List of Abbreviations

ACM	Association for Computing Machinery
CCS	Computing Classification System
MeSH	Medical Subject Headings
NLP	Natural Language Processing
FAO	Food and Agriculture Organization
BT	Broader Term
NT	Narrower Term
RT	Related Term
HEP	High Energy Physics
SKOS	Simple Knowledge Organization System
W3C	World Wide Web Consortium
RDF	Resource Description Framework
URI	Uniform Resource Identifier
XML	Extended Markup Language
XTM	XML Topic Maps
LIMBER	Language Independent Metadata Browsing of European Resources
CERES	California Environmental Resource Evaluation System
OWL	Ontology Web Language
RA	Refinement Algorithm
ERA	Extended Refinement Algorithm
TP	True Positives
FP	False Positives
FN	False Negatives
TN	True Negatives
CV	Critical Value

List of Figures

List of Tables

List of Algorithms

ABSTRACT

Keyphrases facilitate in finding right information from digital documents. Keyphrase assignment is the alignment of document or text with the keyphrases of any standard classification taxonomy. Kea++ is a famous tool for performing keyphrase assignment automatically; however it assigns irrelevant terms along with the relevant ones. In order to reduce noise in the Kea++ result set, refinement rules were defined in the refinement methodology to exploit the semantics of the hierarchical structure of the taxonomy. This methodology is a top layer on Kea++. It was evaluated on computing domain taxonomy and showed better results than Kea++.

However the refinement methodology is more focused on computing domain taxonomy and does not perform well in case of taxonomies having deep hierarchy of keyphrases. Training-level is the hierarchical level of taxonomy which is adopted in manually generated keyphrases for documents in the training dataset of Kea++. In refinement methodology, the training-level is the key parameter for selection or rejection of any keyphrase in Kea++ result set. But its selection process does not offer priority to the taxonomy level where maximum keyphrases are aligned in the training dataset. Moreover, the methodology lacks in applying standard terminology used in taxonomy languages.

This work is aimed to extend and generalize the refinement methodology for multiple domains and improve its results. In the proposed extended refinement methodology, the training-level selection process has been revised and due consideration has been given to taxonomies having deep hierarchy of keyphrases. Standard terminology used in taxonomy languages has been adopted and amended the refinement methodology accordingly to be practical in multiple domains.

The extended refinement methodology was evaluated on three different domain taxonomies and datasets: computing, agriculture and mathematics. Evaluation metrics used were (i) precision, recall and f-measure (ii) average number of assigned keyphrases to test documents and (iii) statistical t-test. The evaluation demonstrates significant improvement in reducing noise in the Kea++ result set for multiple domains. We conclude that the extended refinement methodology has been generalized and can be applied in domains other than computing. It has also shown better results than its predecessor.

Chapter 1

INTRODUCTION

This chapter introduces the research work that has been performed in this thesis. Before identifying the problem areas, problem background is explored. The problems which are being addressed in this research work are listed, which ultimately form the objectives of this work.

1.1 Problem Background

From the plethora of information available online accessing the relevant content efficiently and accurately is important. Keyphrases are one of the ways to facilitate this finding for right information from any digital source. Keyphrases (keywords) are the terms that can describe the whole content of the document precisely and accurately [15, 3], so that one can quickly grasp the idea of the whole content of the document without looking into the details. Since the last two decades the amount of digital content is growing and people are relying more and more on search engines. Keywords and keyphrases are the kind of metadata [45] and this metadata has great significance in digital document repositories. Similarly they can be used to index the documents [45], assist in browsing the collection [18] and help in clustering of documents [24].

Because of the usage of keywords and keyphrases in many areas, need for tools automatically generating them had been felt years ago and many attempts were made in this regard. Kea [48] and its later version Kea++ [28] developed at University of Waikato, are amongst the famous tools used to generate keyphrases automatically. Two approaches used for keyphrase generation are; *keyphrase extraction* and *keyphrase assignment*. In keyphrase extraction, keyphrases are assigned to document from the document text. On the other hand in keyphrase assignment, keyphrases are assigned to document from the domain specific taxonomy. Kea++ is a tool that can perform both keyphrase assignment and extraction based on the given input and is termed as *keyphrase indexing algorithm* [32]. There are many other tools that can perform keyphrase extraction, such as [45, 22, 38, 44], but very few tools are there that can perform keyphrase assignment. Our main focus in this work is on keyphrase assignment.

However the assignment performed by Kea++ contains irrelevant terms in the result set. The refinement methodology [13] was developed to fine tune the result set of Kea++. The methodology consists of refinement rules. The rules were designed to exploit the hierarchical structure of taxonomy for reducing noise in the Kea++ result set. Based on refinement rules, refinement algorithm was developed. It works as a top layer over Kea++. This algorithm was tested on computing domain dataset. Results returned by the algorithm showed improvement.

1.2 Problem Definition

The refinement algorithm proposed in [13], resulted in the minimization of noise and irrelevant terms from the result set of Kea++. However the formation of the refinement methodology is tilted towards computing domain. The refinement rules defined in refinement methodology were formulated by keeping ACM Computing Classification System in mind. The refinement methodology does not perform well especially for taxonomies having deep hierarchical tree structure. For instance MeSH, which is the taxonomy for medicine domain, has a deep hierarchical tree structure, because MeSH taxonomy tree is deep up to twelve levels. Training-level is the hierarchical level of taxonomy which is adopted in manually assigned keyphrases for documents in the training dataset of Kea++. In refinement methodology, the training-level is the key parameter for selection and rejection of any keyphrases in the result set of Kea+. But its selection process does not offer priority to the taxonomy level where maximum keyphrases are aligned in the training dataset, because of which refinement methodology does not produce desired result. Furthermore, the rules are not compliant with the standard language used for taxonomy representation, making application of the refinement methodology hard on different domain's taxonomy.

1.3 Research Objectives

Our main focus in this work will be on the extension, generalization and improvement of the refinement methodology developed earlier. In order to meet the objectives, different domain specific taxonomies will be analyzed like computing, agriculture, mathematics, health sciences, art and architecture etc. The refinement methodology will also be adjusted for dealing with the taxonomies having deep hierarchical tree structure. The founding rule dealing with the selection of training-level will be revised. A different approach will be adopted this time for the selection of training-level. This new approach will give priority to the level upon which majority of the keyphrases in the training dataset are aligned. Standard language mostly used for the representation of taxonomies will also be identified and an attempt will be made to reformulate the refinement methodology, making it applicable to different domain taxonomies. As a result of this, the extended refinement rules will be proposed. These rules will be extended, generalized and will be aiming to improve the refinement methodology proposed earlier.

In order to test the extended refinement algorithm, datasets will be taken from three different domains: computing, agriculture and mathematics. The dataset belonging to each domain will be divided into many groups. Multiple but same tests will be performed on each dataset to check any reduction in noise and irrelevant terms from the result set of Kea++. The results obtained will also undergo statistical t-test, to strengthen the claim of the improvement achieved by the extended refinement methodology.

1.4 Thesis Outline

The rest of the thesis is organized as follows: Chapter 2 presents the detail background of important and founding concepts. Chapter 3 highlights the related work associated with this thesis. In this chapter different existing keyphrase extraction and assignment algorithms are explained in detail. Chapter 4 presents the work proposed in this thesis, along with the implementation details. Chapter 5 deals with the testing and evaluation of the proposed methodology. Chapter 6 formulates the conclusion obtained as a result of this work and also mentions the future directions.

Chapter 2

BACKGROUND

Internet has become the greatest source of information since last two decades. According to world's Internet usage statistics[1] currently, nearly 2 billion people are using Internet. Day by day number of users looking for the information on Internet is increasing. With time this searching for information has been improved. Lots of mechanism, methodologies and systems have been introduced to make the process of looking for information fast, accurate and reliable. One of the approaches towards facilitation of search to user is to provide them with categories, keywords or subjects headings, so that they can look for the information according to their requirement and get the accurate results. These categories, keywords or subject headings help in organizing the content, so that access towards information becomes quick and precise. This is not the only use of keywords, they can be utilized in a number of ways which we will seen in the proceeding sections.

This thesis is particularly dealing with the automatic keyphrase assignment task. In this chapter we will first look into the basic concept of what keyphrases are, how many types of keyphrases are there according to the domain of their usage, types of keyphrase generation and what are the application areas in which keyphrases can play an important role.

[1] http://www.internetworldstats.com/stats.htm

2.1 Keyphrases

Keyphrases can consist of more than one words. They are the words that can describe the whole content of the document in a very precise manner. In other words by looking at the keyphrases one can grasp the idea of what the whole document is all about. When keyphrases consist of single word they are referred as keyword. According to [15] keyword can be defined as "word which succinctly and accurately describes the subject or an aspect of the subject."

2.1.1 Types of Keyphrases

According to the domain in which keyphrases are applied they are given different names, which are sometime confusing for the new user of information retrieval and information searching field.

When keyphrases are taken from controlled vocabulary or sets of domain specific concepts they are sometimes referred as index terms, subjects headings and descriptors. In other words when the document is being assigned keyphrases not from terms occurring in the document rather from the set of domain specific concepts they are called as ***index terms, subject headings*** and ***descriptors*** [35].

When the keyphrases are taken from the terms occurring in the documents based on certain feature values they are referred as ***keyphrases*** or ***keyword*** based on how many number of words they consist of [35].

When it's the matter of assigning keyphrases to the websites freely, rather than relevant to any domain specific concepts they are referred as ***tags***. Tagging is mostly done for websites that host user-generated content, such as blogging platforms, online bookmarking services and file sharing sites. Often several users tag the same object and their tags are merged into a single set. The entire set of tags assigned by all users of a given website is called a folksonomy [29].

As an example we can say that a person not related to medical domain maintains a blog that can tell about the abuse of painkillers. He might choose painkillers as a tag for his web content. But if a medical researcher would publish his new research related to painkiller, it is probable that he will assign the term analgesic to his research publication. Similarly if in the publication mostly word painkiller is used then as a keyword, word painkiller will be extracted from it. Though these terms are marked as synonyms by Wordnet and Wikipedia but this shows the difference in the same concept according to the scenario in which they are being chosen as words to represent the content.

2.1.2 Types of Keyphrase Generation

Similarly based on the scenario in which keyphrases are being used, the task of generating the keyphrases can also be categorized in the following manner.

Keyphrase Extraction: The process of selecting those keyphrases or keywords that appear to be the most prominent occurring word in the document is called keyphrase extraction [48].

Keyphrase Assignment: The process of assigning keyphrases to the document from the set of domain specific concept expressed in the form of taxonomy or controlled vocabulary is called keyphrase assignment [48]. This can also be referred as subject indexing.

Text Categorization: The process of classifying the documents under generalized concepts or categories is called text categorization. This task is somewhat similar to that of keyphrase assignment in the scenario where there are very few generalized concepts or categories available and the documents are categorized or classified according to them. This is sometime referred as text classification too [41].

Full-text Indexing: When all the terms occurring in the documents excluding stopwords are being extracted and stored to facilitate the organization and searching task. This is also called as free-text indexing [29].

Keyphrase Indexing: Keyphrase indexing is the process of extracting the keyphrases from document content or assigning them from the controlled vocabulary based on the given inputs [28]. This is a more general term that can be used in the scenario where both keyphrase assignment and extraction can be performed depending upon the kind of input given.

Tagging: The process of assigning tags to the web content is called tagging. The user defines as many topics as desired. This can also be referred as social or collaborative tagging [30].

As part of this research we will throughout use the term keyphrase instead of subject headings, index terms or descriptors. We will be dealing with keyphrase indexing task mainly, in which we will make use of controlled vocabulary for assigning keyphrases to documents. So in this thesis the word keyphrase assignment is used to refer to the assignment task performed by keyphrase indexing algorithm. Keyphrase indexing, keyphrase assignment and keyphrase extraction are closely related and in the next section we will present a comparative analysis of these tasks.

2.1.3 Extraction vs. Assignment vs. Indexing

Keyphrase extraction is relatively simpler task when compared to keyphrase assignment and indexing [29]. The properties of phrases occurring in the document were used to determine whether it was a keyphrase or not. This approach though simple, but resulted most of the time in the extraction of irrelevant keyphrases. Because the semantic links or connection between words were not at all considered. This approach was improved by the usage of Natural Language Processing (NLP) techniques. Despite that it was not able to produce accurate results or results close to that produced by human indexers [28].

Keyphrase assignment is meant to select keyphrases from a controlled vocabulary that best explains a document. Earlier it happened that this approach relate a set of training documents with each phrase in the vocabulary. Then it built a classifier for each phrase occurring in the manually assigned keyphrases in the training dataset. Document supplied in the test dataset was processed by each classifier. Then associate the keyphrase with the document if the document feature values positively classified to that keyphrase. This approach considered the document sense. It analyzed the content of document by taking into account the co-occurrence statistics between terms. In that manner, it is better than that of keyphrase extraction which only considered the syntactic properties of phrases occurring in the document, and ignored the content of the document as whole. Thus failed to cover all topics in the document [48, 16]. This is similar to text categorization task as it was mentioned in [10], rather keyphrase assignment earlier was a more problematic version of text categorization. This was because number of documents needed for training purpose grew rapidly as number of categories increased.

Keyphrase indexing technique [28, 31] is claimed to avoid the drawbacks of both keyphrase assignment and extraction and adopts a hybrid approach i.e. it can perform both extraction and assignment task. While performing extraction it worked similar to that of conventional keyphrase extraction. But during assignment it extended the keyphrase extraction by using the semantic (USE/USED FOR) relation between phrases that occur in the document. In other word, phrases that were non-descriptors (terms that can be considered as synonym terms for the main entry terms in the vocabulary) were replaced by the descriptors (main entry terms in the controlled vocabulary). Finally phrases set were matched against the controlled vocabulary to get the result set. This approach proved to be better in performance than that of keyphrase extraction and assignment. It can be improved further if instead of exact matching of phrases with the controlled vocabulary (which is also represented as phrases), it can utilize deeper semantic relations. It means it should identify those phrases in the final result set that can semantically relate to any term in the vocabulary. Instead of finding exact match between phrases identified from document and those from controlled vocabulary [32]. Pros and cons of keyphrase extraction, assignment and indexing are summarized below:

Keyphrase extraction:
Pros:
1. It does not require controlled vocabulary. Instead properties such as, frequency and length of the words occurring in the document are measured to extract the appropriate keyphrases.
Cons:
1. It often results in inappropriate keyphrases.

Keyphrase assignment:
Pros:
1. It assigns well formed and appropriate keyphrases to the documents.
Cons:
1. Controlled vocabularies are not always available as they are expensive to build.
2. It needs larger training document set for each manual keyphrase that has been provided with training dataset, in order to build efficient classifier.
3. Sometimes potentially useful keyphrases are ignored if they are not in the vocabulary.

Keyphrase indexing:
Pros:

1. Semantic relationship between terms is utilized to assign appropriate keyphrases to the document.
2. It does not require large number of training dataset for each manual annotation that is provided with the training dataset.
3. The final result set does not necessarily consist of only those terms that are there in the manual annotation in the training dataset (as it is done in keyphrase assignment). Instead semantically related keyphrases can also be there.
4. User does not need two separate tools for assignment and extraction, rather only change the inputs to perform any of the two tasks from the same algorithm.

Cons:

1. Controlled vocabularies are not always available as they are expensive to build.
2. Limited semantic relationship is applied to assign keyphrases to document from controlled vocabulary.
3. Sometimes potentially useful keyphrases are ignored if they are not in the vocabulary.

2.1.4 Application of Keyphrases

The keyphrases can be used in many applications and in many ways. Following is the brief description of some of their usage as mentioned in Extractor[2]:

Keyphrases for Metadata: One of the uses of keyphrases is that they can be used as metadata. Metadata is the data about data. Metadata is all sort of information that can support or provide information about the main content of the data related to any system.

Keyphrases for Indexing: An index is a systematic arrangement of entries designed to enable users to locate information efficiently from any system that can store some kind of data. Databases spend a lot of their time in finding things. So this finding needs to be performed as fast as possible to speed up the searching mechanism. Indexes provide the basis for both rapid random lookups

[2] http://www.extractor.com/

and efficient ordering of access to data. An alphabetical list of keyphrases, taken from a collection of documents or from single document can serve as an index.

Keyphrases for Interactive Query Refinement: While searching user often adopts the iterative approach. This happens in a manner that the results of query provided initially are being analyzed. Then based on this, user reformulates the query to get the accurate and suitable results according to his requirement. One of the usages of keyphrases is to support search engines with iterative query refinement. It can facilitate the user in a manner that keyphrases from the first round of search should be displayed to the user. He can quickly grasp the idea of what kind of output he has received for his query. Then ultimately refine or regenerate his query to get the better results.

Keyphrases for Web Log Analysis: It is a common practice that web managers analyze kind of users visiting their website. They often do this with the help of web log analyzer programs. Whenever a website is being accessed by the user, it records the client machine Internet address, date and time in which the web page was visited and kind of files that are most commonly accessed. These programs record the traffic pattern and list of popular files that are being accessed. One of the uses of keyphrase can be made here by providing the list of keyphrases to web managers after analyzing the files, instead of providing them the list of files. This can make their task very quick and they can easily get the idea of the topics their website is most commonly accessed for.

2.2 Taxonomy

As mentioned earlier, in this thesis we will be dealing with the task of keyphrase assignment in which keyphrases assigned to the documents are taken from the controlled vocabulary. Often we find words like taxonomy, thesaurus, and domain ontology that are being used interchangeably for controlled vocabulary. In broader sense they can be called as different names for the same thing i.e. set of domain specific concepts but when closely analyzed they have some differences amongst them as highlighted in [47]:

Table 2.1: Differences among various taxonomic structures

	Structure	**Relationship among concepts**
Controlled Vocabulary	Flat	None
Thesaurus	Flat	Related concepts are listed
Taxonomy	Hierarchical	Parent-child and associative relationships
Ontology	Flat/Hierarchical	All kind of possible relationships

Controlled Vocabulary: A list of terms that have been enumerated explicitly is called as controlled vocabulary. This list is controlled by and is available from a controlled vocabulary registration authority. All terms in a controlled vocabulary must have an unambiguous, non-redundant definition [35]. The terms used in controlled vocabulary are the set of domain specific concepts that can be used in constructing the taxonomy, thesauri and indexing schemes.

Thesaurus: Thesaurus is a controlled vocabulary arranged in a known order and structure. So that various relationships among terms are displayed clearly and identified by standardized relationship indicators. Relationship indicators should be employed reciprocally. It lists every important term in a given domain of knowledge along with a set of related terms for each term in the list [35].

Taxonomy: A collection of controlled vocabulary terms organized into a hierarchical structure is called taxonomy [17]. Each term in a taxonomy is in one or more parent/child (broader/narrower) relationships to other terms in the taxonomy [35]. Also sometime associative relationship is incorporated between concepts as well.

Domain Ontology: Domain ontology is a formal specification of concept and all sorts of relationship amongst them [34], so that logic and reasoning can be applied on them [50, 19]. Taxonomy is a kind of lightweight ontology that can represent the hierarchical kind of relationship among concepts.

Table 2.1 highlights the differences between different kind of taxonomies. It is important to mention that, this thesis mainly deals with the hierarchical and associative relationships among concepts that can be assigned as keyphrases to the document. So from now onwards, we will be using the word taxonomy in this thesis. Forth coming section will tell about the taxonomies belonging to various domains and languages that are used to represent these taxonomies in detail.

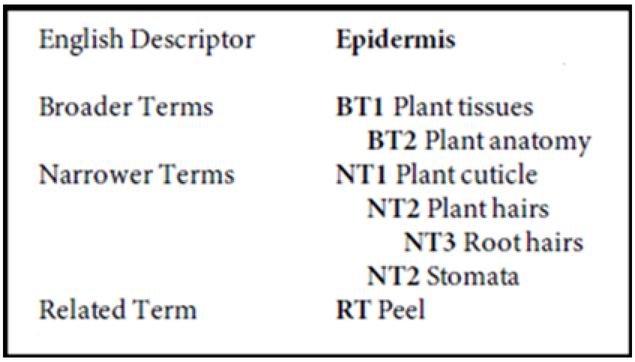
2.2.1 Basic Concepts Related to Taxonomy

Before exploring the various domain specific taxonomies and language available to express the taxonomy, first we will look into some of the basic concepts that can make us better understand the structure of these taxonomies.

Semantic Relations: For assigning keyphrases to the document one must understand the whole context of the document. What the document is all about and what is the exact sense behind main concepts contained in the document. Also the relationships that exist among these concepts should be well understood. Both the text and the knowledge present in the document can be expressed in the form of semantic network. In semantic network the terms are connected to each other by semantic relations, based on the similarity of their meaning and usage. There can be many forms of this semantic relation i.e. it can be equivalence, hierarchical and associative [23].

a. Equivalence Relation: A relationship between terms in a controlled vocabulary that leads to one or more terms, that can be used instead of the term from which the cross-reference is made is called equivalence relation [35]. Terms that are related by this kind of relation are equivalent in their meaning. In other words, this can be expressed as synonym terms. Terms that are similar in their meaning or make similar sense or can be used interchangeably. In taxonomy main concepts are usually termed as descriptors and equivalent concepts can be called as non-descriptors. It can be denoted by word SEE. Also we can denote this kind of relation as USE (or the symmetric USED-FOR) relation. For instance in agrovoc which is the taxonomy for agriculture domain *circulatory system* is a non-descriptor for the descriptor *cardiovascular system*.

b. Hierarchical Relation: A relationship between terms in a controlled vocabulary that depicts broader (generic) to narrower (specific) or whole-part relationship is called hierarchical relation [49]. This can be termed as parent-child



English Descriptor	**Epidermis**
Broader Terms	**BT1** Plant tissues
	BT2 Plant anatomy
Narrower Terms	**NT1** Plant cuticle
	NT2 Plant hairs
	NT3 Root hairs
	NT2 Stomata
Related Term	**RT** Peel

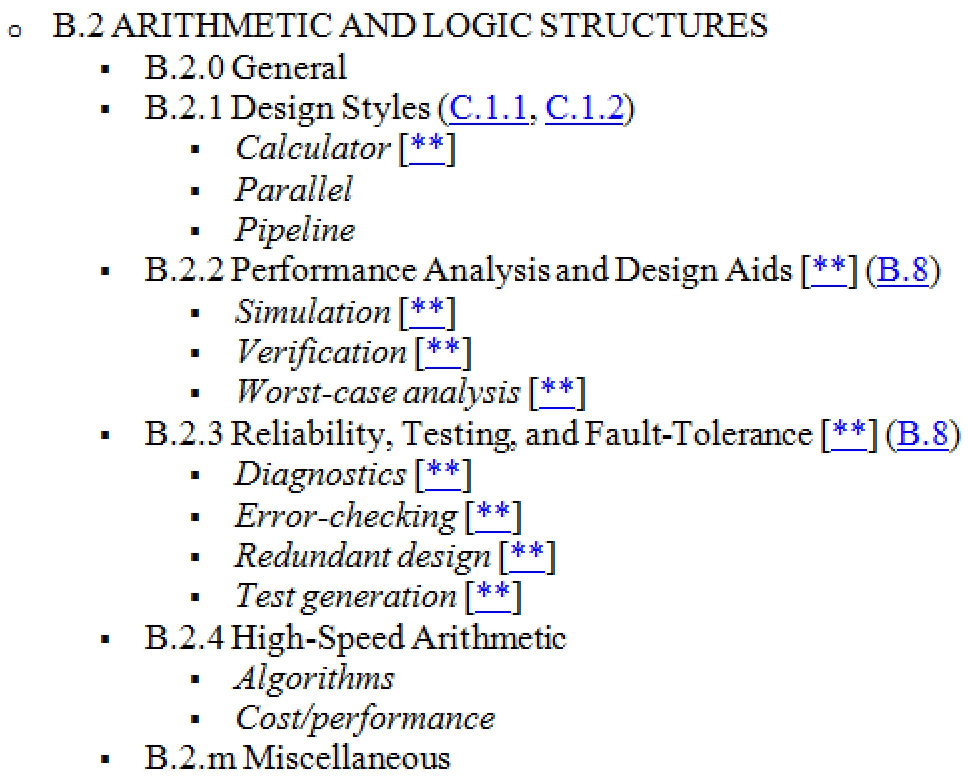
Figure 2.1: Snippet from Agrovoc for entry term *Epidermis* [29]

relationship as well. There exist two kinds of hierarchical relations among concepts within the taxonomy; Broader (parent) term (BT) and Narrower (child) term (NT) relations. BT relation denotes more generalized concept while NT represents the more specific concept when appearing in the hierarchical organization. Hierarchical relationships are based on degrees or levels of superordination and subordination, where the superordinate term represents a class or a whole, and subordinate term refers to its members or parts. Examples of BT and NT relations for descriptor *Epidermis* from agrovoc taxonomy is shown in figure 2.1.

c. Associative Relation: A relationship between terms in a controlled vocabulary that leads from one term to other terms that are related to or associated with it, is called associative relation [49]. This is usually represented with the words SEE ALSO or related term (RT). Related term (RT) relation represents the associative kind of relationship among concepts. This relationship covers association between terms that are neither equivalent nor hierarchical, yet the terms are semantically or conceptually associated to such an extent that the link between them should be made explicit in the taxonomy. Example for RT is also shown in figure 2.1.

2.2.2 Domain Specific Taxonomies

Various real world domain specific taxonomies and their taxonomic structures are explained below:



- B.2 ARITHMETIC AND LOGIC STRUCTURES
  - B.2.0 General
  - B.2.1 Design Styles (C.1.1, C.1.2)
    - *Calculator* [**]
    - *Parallel*
    - *Pipeline*
  - B.2.2 Performance Analysis and Design Aids [**] (B.8)
    - *Simulation* [**]
    - *Verification* [**]
    - *Worst-case analysis* [**]
  - B.2.3 Reliability, Testing, and Fault-Tolerance [**] (B.8)
    - *Diagnostics* [**]
    - *Error-checking* [**]
    - *Redundant design* [**]
    - *Test generation* [**]
  - B.2.4 High-Speed Arithmetic
    - *Algorithms*
    - *Cost/performance*
  - B.2.m Miscellaneous

Figure 2.2: Snippet from ACM Computing Classification [1]

(I) ACM Computing Classification: ACM Computing Classification System (CCS)[3] is used as a standard topic hierarchy for the domain of Computer Science so that users can take the advantage of efficient search and fast content reference. The ACM Computing Classification comprises more than 1250 terms and also specify the relations between them. Documents are aligned on a specific node in the ACM Computing Classification according to the content of the documents. Snippet in figure 2.2 shows ACM Computing Classification's structure.

The tree consists of 11 first-level nodes denoted by letter designation A to K, and one or two sub levels under each of these. The ACM CCS tree has a depth of four in which first three levels are coded and the fourth level is not. Alphanumeric codes are assigned to the second and third levels. Terms at the uncoded level are called *subject descriptors*. They can provide sufficient detail to cater the need of new developments in the field. Initially they were developed with the concept that they can be allowed to change frequently. But with time this has been realized that its very difficult to remove subject descriptors, without obliterating the references to works classified under them. So it was decided to make them a permanent part of the classification tree. As shown in figure 2.2

[3] http://www.acm.org/about/class/1998/

Algorithms and *Cost/performance* that comes under *B.2.4 High- Speed Arithmetic* are the examples of subject descriptors. In ACM CCS retired or non-active terms are marked with either an asterisk (*) or double asterisk (**). By non-active it means that they are not used for classifying the work any more, but can only be used to find out the work classified under them in past. Some of the non-active nodes point in parentheses to the new terms replacing them for instance as shown in figure 2.2, *B.2.3 Reliability, Testing, and Fault Tolerance** (B.8)*. Some active nodes can have semantically similar terms associated with them. These terms are mentioned in parentheses along with them, for instance, *B.2.1 Design Styles (C.1.1, C.1.2)* in figure 2.2. Along with the subject descriptors ACM CCS has implicit subject descriptors. They are nothing but terms that are proper nouns like *C++* is an implicit subject descriptor under *D.3.2 Language Classifications*. Like subject descriptors they are also uncoded. Also there is a set of 16 separate concepts called "General Terms" that apply to all areas for instance *Languages*, *Theory*, and *Human Factors* are the general terms. "Miscellaneous" node in the given area is used to classify those papers that cannot be classified under any other node. In the Figure 2.2 *B.2.0* and *B.2.m* are the examples of General and Miscellaneous nodes respectively. We can also see that each keyphrase (i.e., concept) has some sub-keyphrases (i.e. sub-concepts) which are referred as narrower keyphrases such as *B.2.1 Design Styles* has a sub concept *Pipeline*. Similarly a sub-concept has some broader concepts, such as *B.2.1 Design Styles* is broader keyphrase of *Parallel*. The organization of keyphrases in broader and narrower levels forms the hierarchical structure of taxonomy.

(II) Agrovoc: Agrovoc[4] is a multi-lingual thesaurus developed for agriculture, forestry, fisheries, food and related domains (e.g. environment). It has been developed by the UN Food and Agriculture Organization (FAO). It is used by them to classify the documents in their large and well used online document repository. The English Agrovoc defines over 28,000 concepts; preferred term (descriptor) and associated non-descriptors, extends its size up to 40,000 terms. The concepts in Agrovoc are interconnected with each other through RT, NT and BT semantic relations. Number of such semantic links goes up to 83,000 in Agrovoc. Terms that appear similar but are semantically different are identified by parentheses for instance Vanilla (genus) and Vanilla (spice). If needed, scope note is also added with terms to make clear their intended meaning. Example snippet from Agrovoc is already shown in figure 2.1. Unlike CCS, concepts of General terms and Miscellaneous terms are not applied in Agrovoc. Also in

[4] http://aims.fao.org/website/AGROVOC-Thesaurus/sub

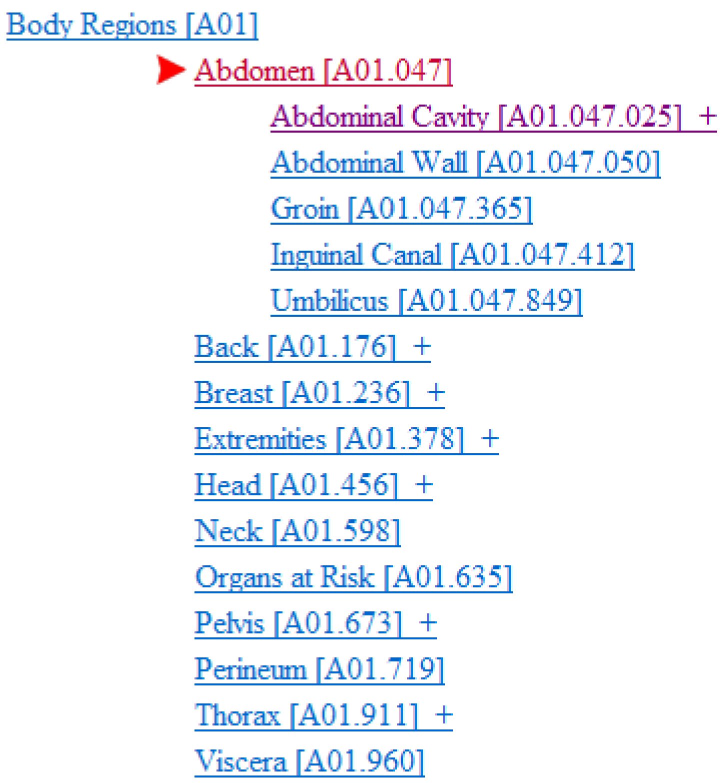
agrovoc semantically related or equivalent terms are considered as non-descriptors unlike ACM CCS. In ACM CCS equivalent terms are also descriptors rather than non-descriptors or non-preferred terms.

(III) Medical Subject Headings thesaurus (MeSH): MeSH[5] is the taxonomy for medicine domain used by the National Library of Medicine. In MeSH concepts are called *Descriptors* ; *Descriptors* contain *Concepts* and *Concepts* contain *Terms*. Exactly one concept is the preferred concept for a descriptor and exactly one term is a preferred term for a concept, which is usually not the case in other taxonomies and thesauri. Each concept has a name and a unique identifier. Also with each concept documentation is attached such as its date of introduction and historical notes. Descriptors are hierarchically related and each MeSH descriptor has one or more *TreeNumbers*. This number implicitly encode its position in the taxonomy hierarchy for instance A01.047 is a child of A01 as shown in figure 2.3. There are 26,142 descriptors in 2011 MeSH. Descriptors, concepts and terms are coded from top to the lowest level and they are being sorted in alphabetical order. MeSH taxonomy tree is deep up to twelve levels. MeSH concepts that appear within one descriptor can be related to each other with relations BT, NT and RT. MeSH has sixteen trees with top-concepts named e.g. “organisms” or “diseases”. These appear to be facets, but they are also used in indexing articles, so we interpret them as normal thesaurus concepts. MeSH contains 24,000 concepts organized into a hierarchy via 32,000 BT/NT links.

(IV) High Energy Physics-HEP: High Energy Physics (HEP)[6] is the taxonomy for physics domain. It is used by the European Organization for Nuclear Research to classify the contents of the CERN Document Server. It comprises approximately 16,000 concepts with very few non-descriptors and almost 500 BT, NT and RT semantic relations. In addition it also defines a semantic relation called Composite/CompositeOf. However the use of this semantic link has nothing to do with our main task at hand in this thesis.

[5] http://www.nlm.nih.gov/mesh/

[6] http://www-library.desy.de/schlagw2.html



- Body Regions [A01]
  - ► Abdomen [A01.047]
    - Abdominal Cavity [A01.047.025] +
    - Abdominal Wall [A01.047.050]
    - Groin [A01.047.365]
    - Inguinal Canal [A01.047.412]
    - Umbilicus [A01.047.849]
  - Back [A01.176] +
  - Breast [A01.236] +
  - Extremities [A01.378] +
  - Head [A01.456] +
  - Neck [A01.598]
  - Organs at Risk [A01.635]
  - Pelvis [A01.673] +
  - Perineum [A01.719]
  - Thorax [A01.911] +
  - Viscera [A01.960]

Figure 2.3: Snippet from MeSH [46]

2.3 SKOS (Simple Knowledge Organization System)

For making use of the taxonomies in different computing task, it is desirable to express them in machine readable form. So that they can not only be understandable by the computers, but can also be linked and shared among different domain users for performing their task. In order to meet this need W3C (World Wide Web Consortium) developed SKOS (Simple Knowledge Organization System)[7]. SKOS is a common data model for knowledge organization systems such as thesauri, classification schemes, subject heading systems and taxonomies. SKOS gets its basics from RDF (Resource Description Framework)[8], which was developed by W3C to represent the information about resources.

The SKOS data model views a knowledge organization system as a concept scheme comprising a set of concepts. These SKOS concept schemes and SKOS concepts are identified by URIs. Enabling anyone to refer to them unambiguously from any context, and making them a part of the World Wide Web. These concepts can be labeled with any number of lexical (UNICODE) strings.

[7] http://www.w3.org/2004/02/skos/

[8] http://www.w3.org/rdf/

They can be documented using different kinds of notes that can tell additional comments or details about the concept. They can be linked to each other through semantic relation properties. The SKOS data model also provides support for hierarchical and associative links between SKOS concepts.

In SKOS, the properties *skos:broader* and *skos:narrower* are used to assert a direct hierarchical link between two SKOS concepts. A triple <A> *skos:broader* <B> asserts that <B>, which is the object of the triple, is a broader concept than <A>, which is the subject of the triple. Similarly, a triple <C> *skos:narrower* <D> asserts that <D>, which is the object of the triple, is a narrower concept than <C>, which is the subject of the triple. Similarly *skos:related* is used to represent the associative kind of semantic relation among concepts and this relation is symmetric in nature means <A> *skos:related* <B> implies that <B> *skos:related* <A>. SKOS has provided a lot more support as far as the representation of taxonomy from the perspective of machine understandability, data linkage and sharing is there. But for part of our research, we are mainly dealing with hierarchical and associative linkage of concepts with each other.

It is not the case that SKOS which is based on RDF is the only option available for the representation of taxonomy. But SKOS exhibits certain advantages over other languages available for thesauri representation [37], like ZTHES [43], MeSH (Medical Subject Headings) [8] and XTM (XML Topic Maps) [39] which are based on XML (Extended Markup Language). SKOS in its true sense is meant for the representation of conceptual schemes, such as thesauri. It is appropriate to use it for information retrieval and the organization of knowledge on the Web and more specifically on the semantic Web. As mentioned it is based on RDF, which represents the true concept of semantic web. RDF extends XML by not only exchanging the data, but carries the true meaning and context of the concept with it. There are alternatives that employ RDF to represent thesauri, such as LIMBER (Language Independent Metadata Browsing of European Resources) [33] and CERES (California Environmental Resource Evaluation System)[9]. They involve their own developments in modeling thesaurus classes and relations [27]. They are not integrated within the W3C initiatives, because on occasions they have lexical units as their central elements. The adoption of SKOS as a common model to represent thesauri allows conceptual thesauri to be represented in a standardized manner. To an extent, OWL offers greater possibilities of representation and potential application than SKOS. OWL could be used directly to develop ontologies with which one can represent thesauri. However, the direct

[9] http://ceres.ca.gov/thesaurus/rdf.html

Table 2.2: Advantages of SKOS over other alternatives

Formats for Thesaurus representation	**Based on**	**Advantage of SKOS**
ZTHES, MESH	XML	Integration within Semantic Web at descriptive level using RDF
XTM	XML(Topic Maps)	Integration within Semantic Web at logical level using OWL
LIMBER, CERES, ILRT	RDF	Flexible, standardized development based on the conceptual paradigm
Ontologies	OWL	Simplification of representation and maintenance tasks

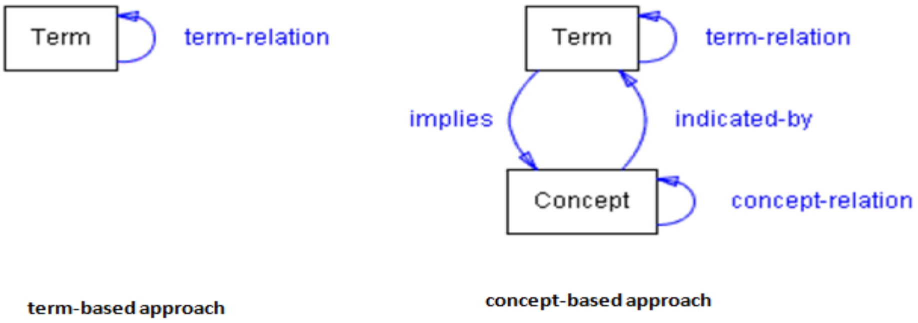


Figure 2.4: Term-based vs. Concept-based model [4]

use of ontologies raises the drawback of the complexity of thesaurus management tasks. This job is simplified with SKOS, while maintaining and expanding the scope of application. As this is a specialized OWL ontology and can be expanded in the future. Table 2.2 summarizes the advantage of SKOS over these available options for thesauri representation.

As we are suggesting SKOS as the preferred format for taxonomy representation. So it is important to mention here two different approaches that exist for taxonomy or thesauri representation; *term-based approach* and *concept-based approach*. In term-based approach, terms are the main unit. They are semantically related to each other through hierarchical, associative and equivalence relations. On the other hand in the concept-based approach, concepts are the main units. They possesses semantic relation with other concepts and terms are only the lexicalization of a concept [42] i.e. a label that are used to express concept. Figure 2.4 shows the difference between these approaches[10]. SKOS is a concept-based model. Concept-based models are the preferred models

[10] http://www.w3c.rl.ac.uk/SWAD/deliverables/8.2.html#2.1

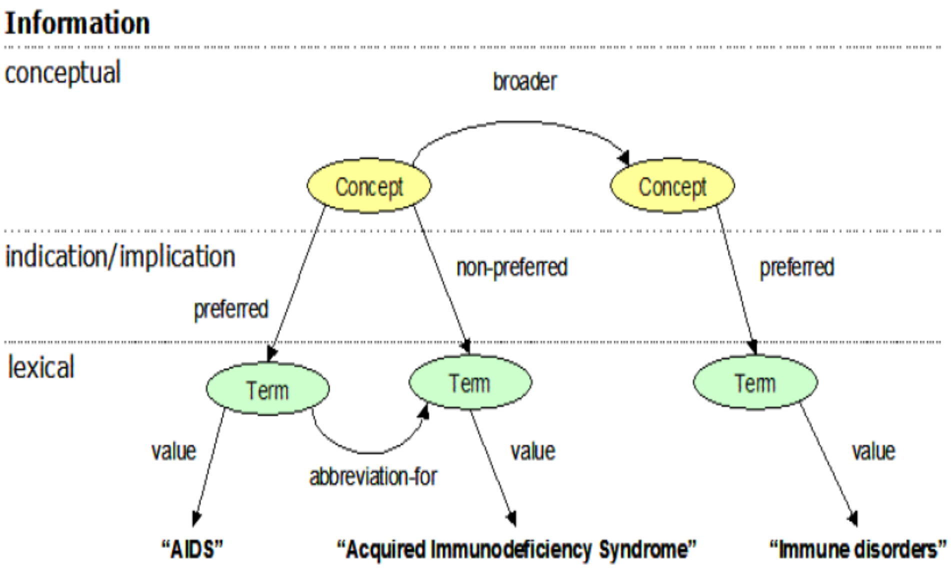


Figure 2.5: Example of Concept-based model [4]

for taxonomy or thesauri representation. As it gives improved clarity and easier maintenance, but nobody can ignore the number of available taxonomies that are in term-based format. Figure 2.5 shows the example of concept-based model taken from [4].

Usually taxonomies are available in SKOS format from their source site. But if not available then one can use the software ThManager [11] to build the SKOS format of any taxonomy. [5] proposed the method of converting available taxonomy into SKOS format. [5] is not the only method available but this work is better from the perspective of completeness and interoperability while conversion in contrast to others [6, 42]. It might be possible that while conversion a term-based taxonomy into concept-based model like SKOS some information get lost. Also some taxonomies have different and complicated structures that needs to be catered while converting into SKOS. These and many other issues are highlighted in their work. Case studies have been used to show how taxonomies having different structure are mapped into main SKOS unit.

In a nutshell, keyphrases are the means to represent the context of document precisely. Tasks like keyphrase assignment, extraction and tagging are the various ways to assign the keyphrases to any sort of document available. Keyphrases are restricted to be the part of the document, or can be assigned freely. Sometimes they are assigned to the documents from set of domain specific concepts, also called as taxonomy. Our task mainly deals with the assignment of

[11] http://sourceforge.net/projects/ThManager

keyphrases to the document from available taxonomy using the semantic relationship that exists among the concepts present in taxonomy. We have also looked at some of the example taxonomies belonging to various domains and analyzed their taxonomic structures. Lastly, we have explored SKOS in detail which is a language available for the representation of these taxonomies.

Chapter 3

RELATED WORK

In the previous chapter, we have defined keyphrases, their importance and application in great detail. As mentioned digital content is increasing drastically as more and more people are relying on digital world. When it's the matter of searching for suitable content, building digital library for an institution or making research content repositories, keyphrases become part and parcel for organizing the digital content. Lots of documents or texts are coming with manually supplied keyphrases. Still there is lot more left and coming that does not have user assigned keyphrases. In fact it is being observed that mostly users do not supply keyphrases until and unless they are bound to do that. The need for tool that can automatically generate keyphrase was felt many years ago and lot of work had been done in this regard. In this chapter we will mention various tools and techniques that have been adopted for facilitating the task of keyphrase generation.

3.1 Automatic Keyphrase Extraction and Assignment Approaches

Following are the different types of approaches used for performing keyphrase extraction and assignment task:

1. **Language dependent approach**

   One of such approach was discussed in [44]. In this work a language based approach was developed. They formed a language model based on statistical techniques. It checks the informativeness and phraseness of the keywords, which are later combined to form a single score. This score is listed in sorted order to pick the most suitable keywords. Higher the combine score, higher is the appropriateness of the keyword. Phraseness is a somewhat abstract notion which describes the degree to which a given word sequence is considered to be a phrase. Whereas, informativeness refers to how well a phrase captures or illustrates the key ideas in a set of documents. [44] have proved the effectiveness of the approach by applying the proposed method on sample data sets and get some quality keywords. But the method lacks simplicity of quantitative analysis of the results obtained. As this is not a machine learning approach this method does not built training model prior to actual keyphrase extraction. With the study we can say that this algorithm also needs modification as far as keyphrase assignment is concerned. So far they have only focused on keyphrase extraction task.

2. Language independent approach

In contrast to the work done by [44], [38] presented an approach that has light-weight preprocessing phase and does not require any prior parameter settings. They named their approach Language Independent Keyphrase Extraction (Likey). Keyphrase assignment and keyphrase extraction in major depends highly on the language used. The need for preprocessing task like including part-of-speech tagging, stemming, use of stop word lists and other language dependent filters is extensive. Likey was developed for keyphrase extraction task in major and it is based upon the ranking and frequency of occurrence of phrase; frequently occurring phrases are assigned higher ranks. The only language dependent factor was the use of Reference Corpora, which is a large collection of documents used to get the idea of language use. They have compared Likey with the tfxIdf (term frequency–inverse document frequency) factor and results of precision and recall was higher in comparison.

3. Machine learning approach

On the other hand Kea [48] and its later version, also the technique proposed by [45], named GenEx, are all supervised learning techniques. Two sets of documents are there; training dataset and testing dataset. Training dataset has been utilized to generate a statistical model, based on whose values results for test dataset are chosen. The advantage of Kea over GenEx is its simplicity of implementation. GenEx is a combination of two algorithms: Genitor, genetic algorithm and Extractor, keyphrase extraction algorithm. It works in a way that Extractor takes a document as an input and produces the list of keyphrases based on some parameters. Whereas Genitor is needed to tune the parameters of Extractor. It is only needed in the training process of GenEx. So the training process is termed as GenEx (includes Genitor and Extractor), whereas, the extraction process is called Extractor as it does not involve Genitor.

4. Natural language processing approach

Another approach that performs better than all the above mentioned techniques is the one proposed by [21]. The idea behind A.Hulth keyphrase extraction algorithm is to add the linguistic knowledge to the extraction process, rather than only using frequency and ngrams factors for extracting the keyphrases. This algorithm with the support of Natural language Processing (NLP) techniques performed better than the rest of the above mentioned algorithms.

Among all the approaches and techniques discussed so far, we can say that Kea and GenEx are the most comparable algorithms. In fact in their work [16, 25] have compared Kea with GenEx. [16] stated that Kea performed comparably to GenEx. They have also discussed factors that can boost Kea performance such as document size, number of keyphrase per document and length of the document. GenEx uses a specialized algorithm for training and extraction, but Kea utilized Naïve Bayes that makes it quicker and faster than GenEx. It has also been proved through experiments that Kea performs well when trained for about 50 documents. After that the improvement in result is very slow. Also it would be better to use Kea with training and testing documents taken from the same domain, rather than both sets belong from different subject areas. Among all the approaches that we have discussed so far, we are interested in Kea and its later versions which are supervised learning techniques. The next part of this chapter will be dedicated to the in depth study of Kea, and its evolution from simple keyphrase extraction algorithm to keyphrase indexing algorithm.

3.2 Kea

Kea was developed at University of Waikato, New Zealand. It was a tool meant to perform keyphrase extraction task automatically. In the work [48] describes Kea's working, it works in two main stages: Training and Extraction. During the initial stage, it creates a model using training documents and manually assigned

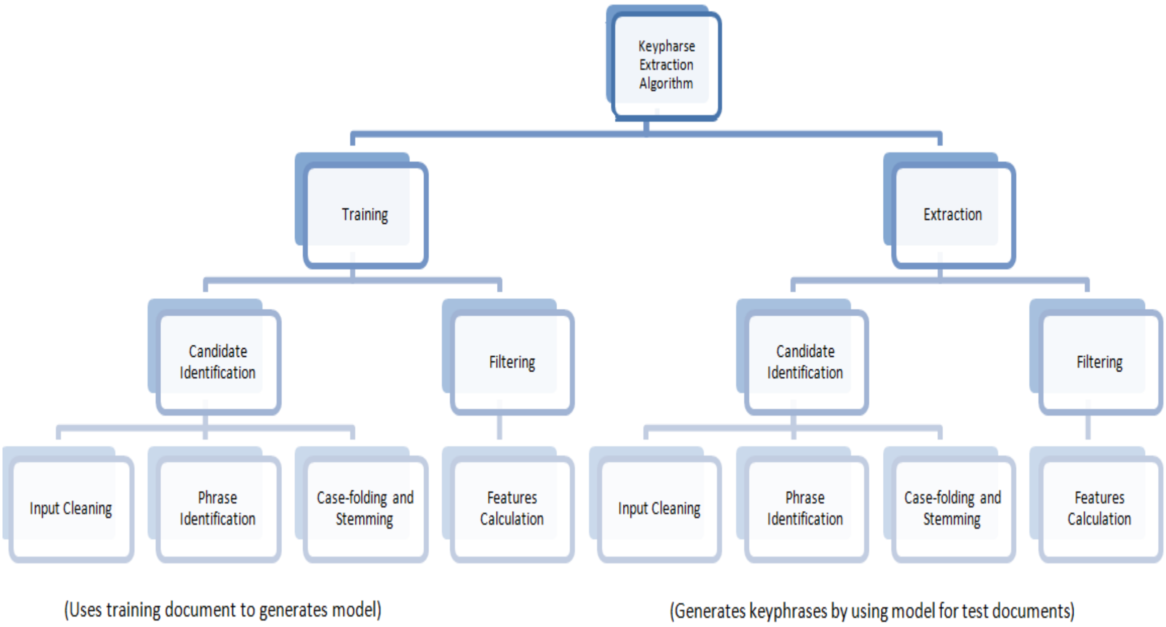


Figure 3.1: Steps performed by Kea

keyphrases. Whereas at the later stage, keyphrases for test documents are generated using the model created in the first stage. Figure 3.1 shows the steps taken to generate keyphrases in both stages.

Training Phase

Briefly, in training phase Kea identifies the textual sequences defined by orthogonal boundaries such as punctuation marks, numbers and new lines. It splits these sequences into tokens in order to extract the keyphrases. Candidate keyphrases consist of one word or concatenation of two or more words (tokens), that do not begin or end with a stopword. A Naïve Bayes learning scheme is used to create a statistical model from training data. For filtering, Kea uses two feature values for each candidate (a) tf x idf measure, (b) distance of the phrase first occurrence in the document from its beginning. It then calculates the overall probability for each candidate keyphrase. This value is then discretized using method described by [14]. Then by applying Naïve Bayes two sets of numeric weights are learned from the discretized feature values. One set is applied to positive examples (those which are considered as keyphrase) and the other is applied to negative examples (which are not considered as keyphrase).

Extraction Phase

During extraction stage when test set is given, same steps are applied to extract candidate phrases. Then their feature values are calculated as mentioned previously. The model developed during training stage is applied to calculate the overall probability. Based on probability values keyphrases are ranked in order. The resulting keyphrase set consists of k top ranked phrases, where k is the user defined parameter.

After this version, Kea came with improvements in code and support for various languages. Then in [28, 31], Kea was enhanced to extract keyphrases from controlled vocabulary for a document whose keyphrase is not known, instead of choosing it from the document itself. The idea given was different from the keyphrase assignment done by [10]. This approach chooses keyphrase from a controlled vocabulary of term, and documents are classified according to their content into classes that correspond to elements of the vocabulary. This new approach was termed as keyphrase indexing. This technique is claimed to avoid the short comings of keyphrase extraction and assignment but utilizing their advantages. This new keyphrase indexing algorithm was called Kea++. Kea++ is described in detail in the next section, as it is the main algorithm upon which this work is based on.

3.3 Kea++

Similar to Kea it works in two stages: Training and Extraction. However what it does in these two stages is somewhat different from that of conventional Kea approach. During each stage it works in two sub stages: Candidate Identification and Filtering.

Candidate Identification

Each document in a collection is segmented into individual tokens on the basis of white space and punctuation that comes under the task of Input Cleaning. All word ngrams that do not cross phrase boundaries are extracted, and stop words are removed that is the Phrase Identification step. To achieve the best possible matching and also to attain a high degree of conflation third phase of Case-folding and Stemming is performed. In this step words are stemmed and sorted in alphabetical order. This results in the formation of pseudo phrases as it was proposed by [36]. Comparison (exact match) has been made between these pseudo-phrases and controlled vocabulary whose terms are also converted into pseudo-phrases form before comparison. After performing these steps non-descriptors are replaced by their equivalent descriptors using links in the thesaurus. This involves the semantic relation between terms and that is what makes Kea++ different from Kea algorithm. This step extract those terms which are similar in meaning to the terms occurring in the actual document content. This step also includes occurrence count, which measures the sum of the counts of all associated full forms of the phrase in the document.

Filtering

This step involves the identification of those keyphrases which are the most suitable candidates based on four features:

a. Term frequency–inverse document frequency (tf×idf): This feature compares the frequency of a phrase used in a particular document with the frequency of that phrase used in general. Document frequency is used to represent the general usage. It represents the number of documents containing the phrase in some large corpus. A phrase's document frequency indicates how common it is. Generally rarer phrases are more likely to be the keyphrases.

b. Phrase's first occurrence: This measure is calculated as the number of words that are before the phrase's first appearance, divided by the number of words in the document. The result ranges between 0 and 1, which represents how much of the document precedes the phrase's first appearance.

c. Length of a candidate phrase in words: This represents how much long the phrase is. It can be controlled by means of the parameter setting done by the user of Kea++. This feature is not used in Kea.

d. Node degree: It is the number of thesaurus link that a candidate phrase has, more are the links more is the probability of the phrase to be the appropriate candidate. This feature is also not included in Kea.

Now based upon these features, decision has been made. Candidate phrases are divided into two classes “is an index term “and “not an index term”.

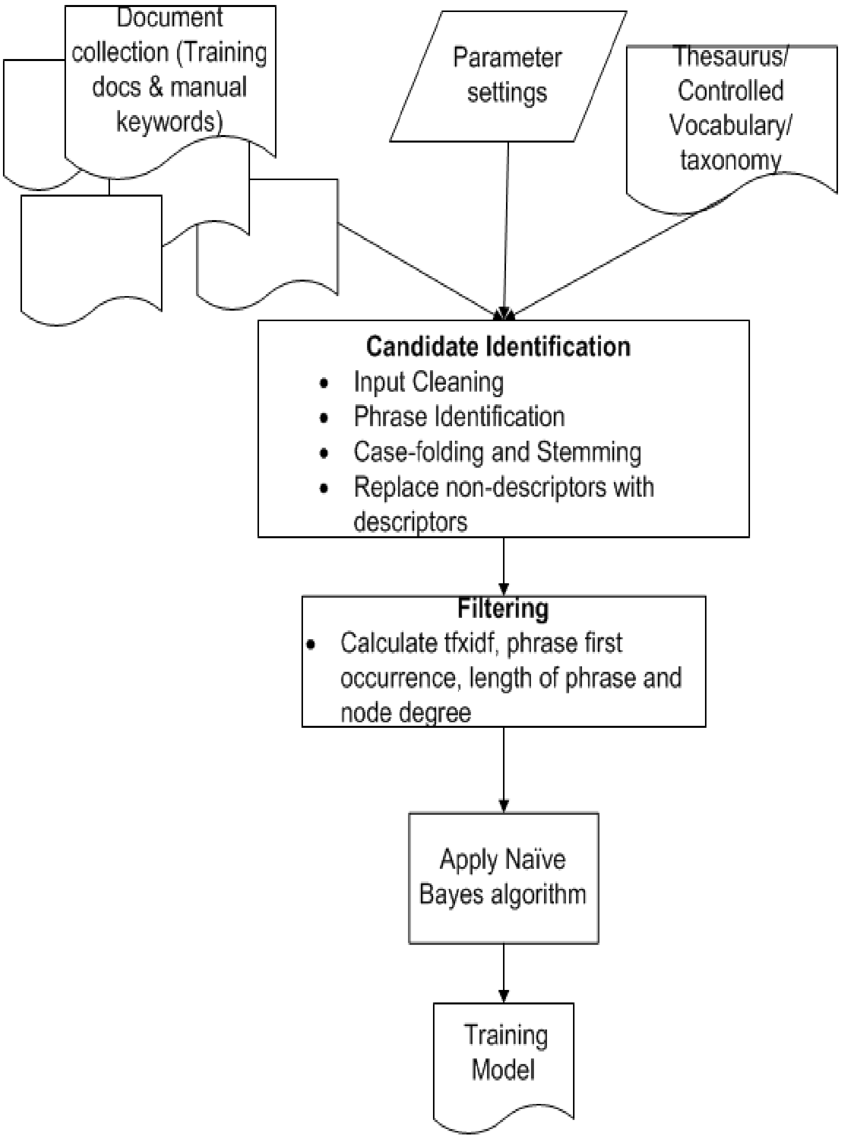

Document collection (Training docs & manual keywords)
Parameter settings
Thesaurus/ Controlled Vocabulary/ taxonomy
Candidate Identification
Input Cleaning
Phrase Identification
Case-folding and Stemming
Replace non-descriptors with descriptors
Filtering
Calculate tfxidf, phrase first occurrence, length of phrase and node degree
Apply Naïve Bayes algorithm
Training Model

Figure 3.2: Kea++ Training Phase

Then by application of Naïve Bayes algorithm phrases are assigned to the respective classes. This results in the generation of a training model that can be used in the Extraction phase. The Training phase takes document collection along with the manually generated keyphrases as input. The Extraction phase is similar in terms of the first two steps of candidate identification and filtering but differs after this. It uses the model to calculate the overall probability of each candidate to be a phrase. It then applies post processing steps like ranking and sorting and picks the best candidates. Figure 3.2 and figure 3.3 illustrates Kea++ training and extraction phases.

Furthermore in [28] extension of Kea++ has also been proposed and evaluated. Extension of Kea++ involves deeper semantic. It extends the idea

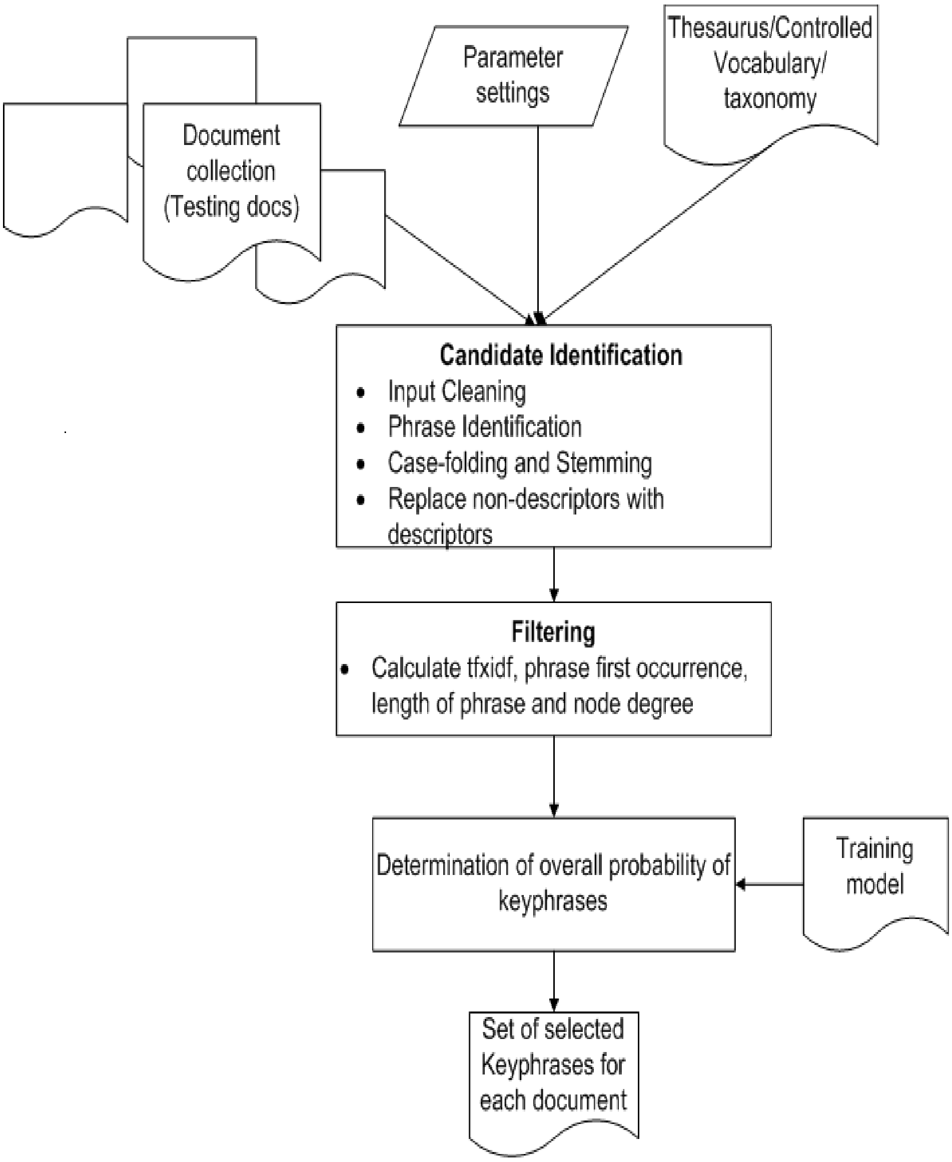

Document collection (Testing docs)
Parameter settings
Thesaurus/Controlled Vocabulary/ taxonomy
Candidate Identification
Input Cleaning
Phrase Identification
Case-folding and Stemming
Replace non-descriptors with descriptors
Filtering
Calculate tfxidf, phrase first occurrence, length of phrase and node degree
Determination of overall probability of keyphrases
Training model
Set of selected Keyphrases for each document

Figure 3.3: Kea++Extracting Phase

of not only matching the exact candidate phrases, but to include all terms that are related to the candidate terms. Even though they may not correspond to pseudo-phrases that appear in the document. Each candidate's one-path related terms, i.e. hierarchical neighbors (BT and NT in Agrovoc) and associatively related terms (RT) are included. This technique helps to cover the entire semantic scope of the document. It increases the frequency of the original candidate phrases based on their relations to other candidates. Along with other evaluation strategies used in [28], semantic based evaluation was performed and three levels of evaluation has been defined:

Level I: keyphrases have equal pseudo-phrases, e.g. epidermis and epidermal.
Level II: keyphrases have equal pseudo-phrases or are one-path related, e.g. epidermis and peel, or plant hairs and root hairs as shown in figure 2.1.
Level III: keyphrases have equal pseudo-phrases or are one- or two-path related, e.g. plant cuticles and root hairs as can be seen in figure 2.1.

Evaluation shows that at Level I matches, Kea++ performs two times better than that of Kea as well as extension of Kea++. But when testing has been performed to check for Level II and Level III matching, extension of Kea++ shows minor improvement over Kea++. However, Kea results still remain lower than that produced by Kea++. Though overall precision and recall measures improved two to three times to that of Kea still its not sufficient. This need of improvement was addressed in [13, 12] by developing the refinement methodology based on refinement rules. It was built by examining the behavior of keyphrase assignment task and Kea++ working. Refinement methodology will be discussed in detail in the next section.

3.4 Refinement of Kea++

The goal of refinement methodology was to improve the semantic alignment by exploiting the hierarchical structure of the taxonomy, and ultimately to eliminate noise from the relevant information [12, 13]. They have found out that the hierarchical levels of the taxonomy and their generalization and specialization play significant role in the training and extraction process of Kea++. In the work refinement rules were developed. Initially the parameters like maximum number of keyphrase, minimum and maximum occurrences of phrases etc were set depending upon the controlled vocabulary in use. Then refinement rules were applied on the keyphrases returned by Kea++. The refinement rules are as follows:

Rule I: Adopting Training-Level
Rule II: Preserving Training-Level Keyphrases
Rule III: Stemming Lower Level General Keyphrases
Rule IV: Preserving Lower Level Keyphrases
Rule V: Identifying and Preserving Training-Level Equivalent Keyphrase
Rule VI: Removing Redundant Keyphrase

The most important factor for these rules is that of training-level. It is the hierarchical level of taxonomy adjusted for manually extracted keyphrases in documents. It is the deciding factor for the selection of keyphrases. The need for carefully selecting the training-level is mentioned in the work as its impact will be on the final result set. Based on these rules refinement algorithm has been developed. This algorithm controls the application of rules based on the level label of the keyphrase contained in result produced by Kea++. If level label returned for a keyphrase is equal to training-level which is found as a result of execution of rule I, then it is preserved in the final keyphrases result set. If lower level keyphrases are there they will be discarded except those that belong to the general category of lower level keyphrases. Lower level keyphrases belonging to the general category are stemmed to training-level and will be added to the final result set. For upper level keyphrases equivalent training-level keyphrases are being searched for. If found then they will be replaced by that training-level keyphrase in the final result set otherwise discarded. Also if no keyphrase in the result produced by Kea++ contains training-level keyphrases then rule IV is applied and preserve lower level keyphrases in the final result set. Figure 3.4 shows the working of Kea++ refinement algorithm and figure 3.5 shows the flow of refinement algorithm.

The refinement algorithm is given in Appendix A. The refinement algorithm when evaluated by performing tests on four separate datasets showed obvious improvement over the performance of Kea++. Precision had been increased in all four tests. Recall either remained same or decreased in some of the tests. The number of irrelevant keyphrases produced in the result set given by Kea++ has been reduced when refinement algorithm was run on that result set. It is important to mention that work done by [13] is an upper layer over Kea++. It can be used after running Kea++ separately to fine tune its results. The work is the foundation of this thesis and will be explored in detail in the next section.

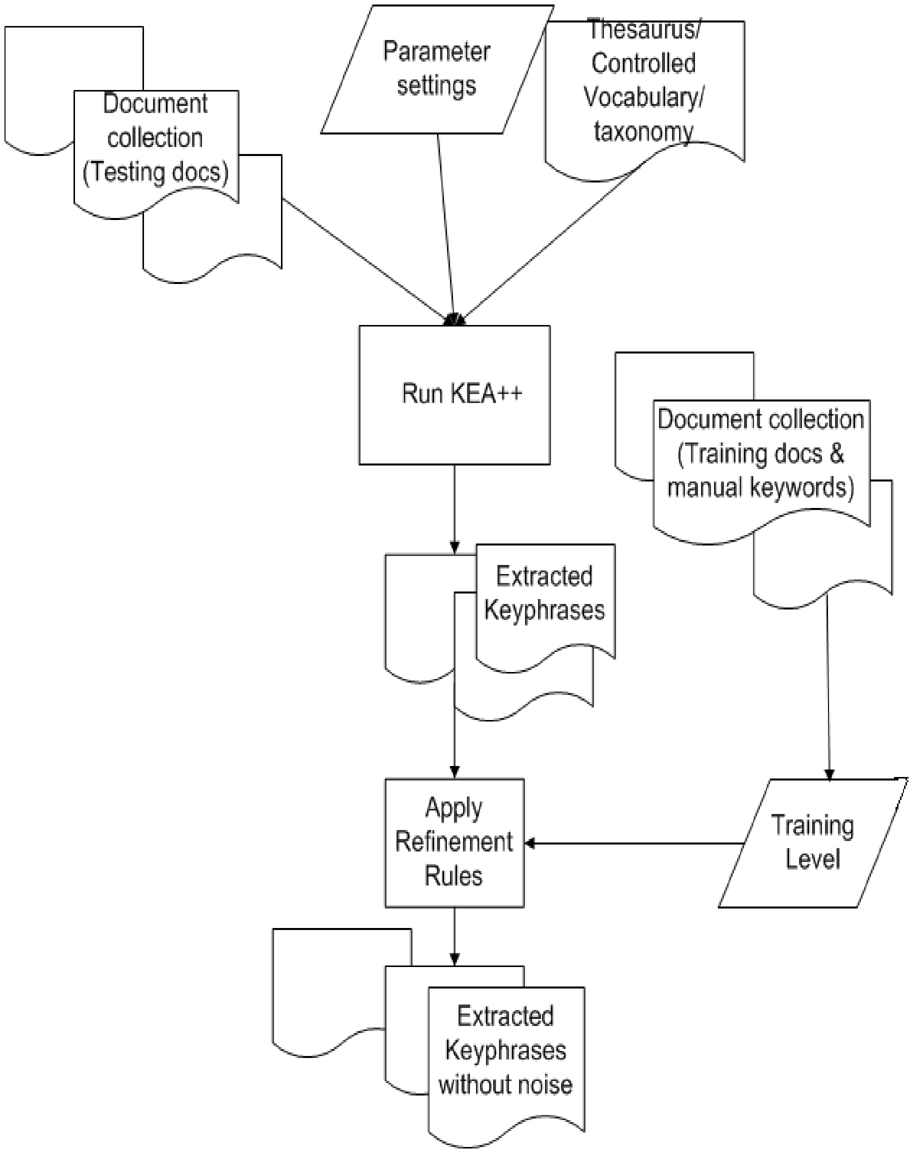

Parameter settings
Thesaurus/ Controlled Vocabulary/ taxonomy
Document collection (Testing docs)
Run KEA++
Document collection (Training docs & manual keywords)
Extracted Keyphrases
Apply Refinement Rules
Training Level
Extracted Keyphrases without noise

Figure 3.4: Refinement of Kea++

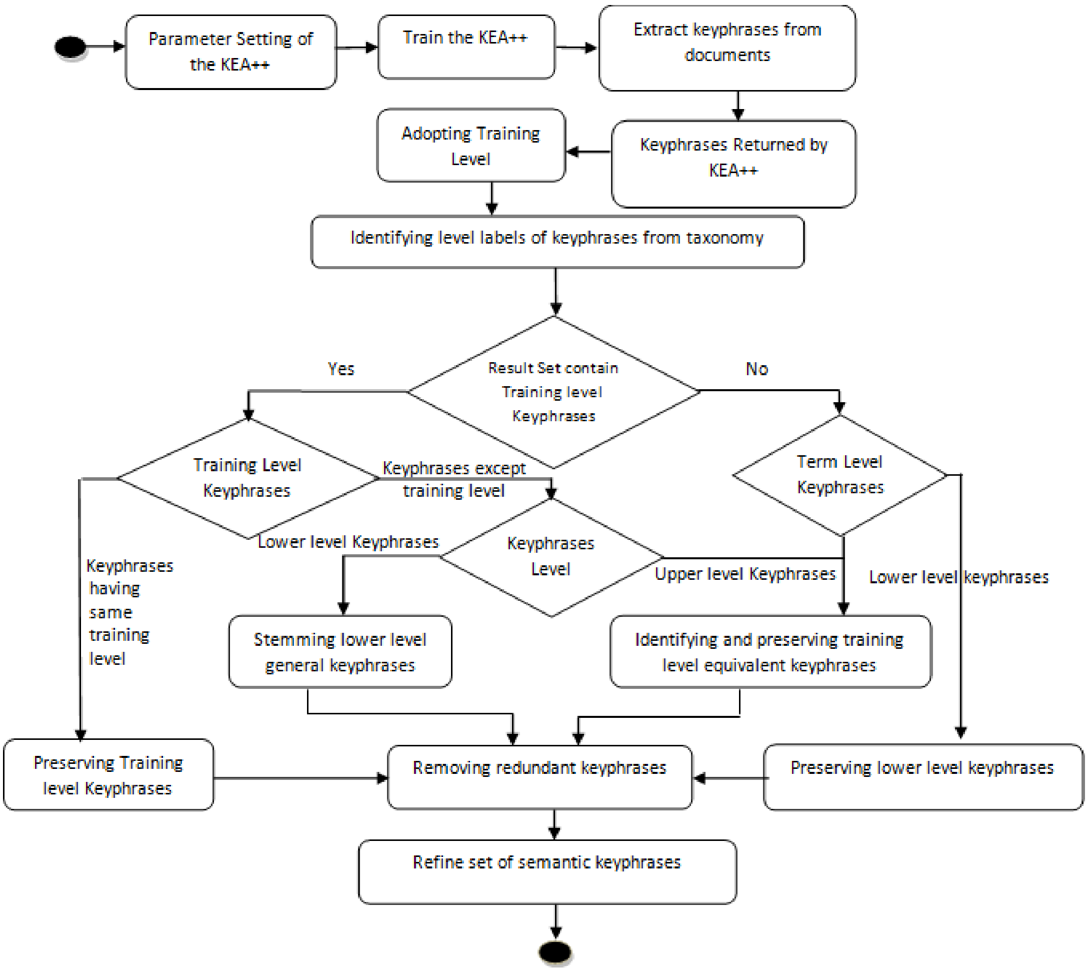

Parameter Setting of the KEA++
Train the KEA++
Extract keyphrases from documents
Keyphrases Returned by KEA++
Adopting Training Level
Identifying level labels of keyphrases from taxonomy
Result Set contain Training level Keyphrases
Yes
No
Training Level Keyphrases
Keyphrases except training level
Term Level Keyphrases
Lower level Keyphrases
Keyphrases Level
Upper level Keyphrases
Lower level keyphrases
Keyphrases having same training level
Stemming lower level general keyphrases
Identifying and preserving training level equivalent keyphrases
Preserving Training level Keyphrases
Removing redundant keyphrases
Preserving lower level keyphrases
Refine set of semantic keyphrases

Figure 3.5: Flow chart for Refinement Algorithm [12]

3.5 Maui

Keyphrase assignment task can also be performed by Maui [29] along with keyphrase extraction and tagging. Similar steps like that of Kea++ were followed to assign keyphrases to the document from taxonomy. Assignment task starts with the extraction of all n-grams up to a certain length, which should match the length of the longest term in the taxonomy. Then normalization has been performed on both n-grams and taxonomy terms in order to ensure good coverage. Normalization includes steps performed to convert n-grams to pseudo-phrases as it was explained in section 3.3. These pseudo phrases are then matched with the taxonomy terms (exact matching). Then semantic conflation is ensured by replacing any pseudo-phrase that matches a non-descriptor with the linked descriptor. However filtering stage utilizes additional features along with those explained in section 3.3. Like Kea++ it first builds model based on training documents and based on this model apply extraction and filtering on the test documents to give related keyphrases. However the real contribution of Maui is its assignment of terms from Wikipedia in the absence of domain specific controlled vocabulary. Also one can perform the desired task of assignment, extraction or tagging based on the given input through one single algorithm.

3.6 Critical Analysis

All the above mentioned approaches attempt to extract or assign keyphrases to documents but none has achieved results closer to that of professional human indexer. Though Maui performs better than the volunteer taggers and students but does not reach the performance level of professional human indexer. A. Hulth technique gives better results than the rest as it uses NLP techniques, but it also cannot replace the accuracy of professional human indexing. All the above discussed methods are used to give keyphrases option to the users, who can pick the words according to their understanding of the document from the result set given. Also there are very few tools performing keyphrase assignment. Kea++ is a good effort in this regard but the result set produced by Kea++ contains irrelevant terms and noise. Refinement of Kea++ improves the accuracy of the result set but its application on taxonomy other than the computing domain is yet to be evaluated. This work will try to evaluate and extend the refinement methodology, aiming to make it generalized and applicable to any domain taxonomy in hand and also try to improve its performance.

In this chapter, we highlighted the major tools and techniques developed to perform the task of keyphrase generation automatically. Their working methodology and problems associated with them are also analyzed. The refinement methodology is the foundation of this research work. It was developed to fine tune the results of famous keyphrases generation tool: Kea++. This methodology was also explained in detail in this chapter. Now in the next chapter proposed methodology will be presented.

Chapter 4

PROPOSED METHODOLOGY

The focus of this work is to generalize, improve and evaluate the refinement methodology developed in [13]. In this chapter, the extended refinement methodology is proposed to accomplish this goal. But before doing that, it is important to formalize the problem areas that will be addressed in the proposed solution. Initially refinement rules developed in refinement methodology will be explained and those needing some improvement will be discussed at the same time. Then an attempt will be made to generalize and improve the refinement rules. Also refinement rules will be improved based on the observations made during the analysis of the taxonomic structure of various domain specific taxonomies.

4.1 Problem Identification

4.1.1 Problem in using different taxonomies

In our work, we are trying to make the refinement rules generalized in order to make them applicable to any domain specific taxonomy at hand. There are some minor and specific implementation and structural details associated with each and every taxonomy. In fact these details vary if we consider different taxonomies according to the specific requirement of the domain to which they belong. So we should select the important and common relations between terms in the taxonomy before extending the refinement rules.

4.1.2 Problem with refinement rules

Now we will explain each and every refinement rule [13] and if there exist any problem with the particular rule, it will be discussed simultaneously as well.

- **Rule I: Adopting training-level:** Training-level in the existing refinement methodology is adopted as the level up to which manually assigned keyphrases in the training dataset are aligned. For example, consider a training dataset which comprises 50 documents and their manually assigned keyphrases. For calculating the training-level, hierarchical level of each keyphrase in the training dataset is found out from the taxonomy. Suppose taxonomy has five hierarchical levels at maximum. The number of keyphrases aligned at level 1, 2, 3, 4 and 5 are 25, 67, 89, 26 and 0 respectively, then as per the selection rule the fourth level is adopted as the training-level, since the keyphrases in the training dataset are aligned up to level four. The training-level selection here is not giving priority to the level upon which majority of the keyphrases are aligned and hence not the true representation of the training dataset.

- **Rule II: Preserving training-level keyphrases:** In refined result set all those keyphrases are preserved that are aligned at training-level. If for instance the training-level adopted as a result of application of rule I is four, then all those keyphrases that are aligned at level four will be preserved in the refined result set.

- **Rule III: Stemming lower level general keyphrases:** For ACM taxonomy, keyphrase that is aligned on the general node and whose level is lower than the training-level will be stemmed to its training-level keyphrase. In the final result the stemmed training-level keyphrase will be preserved. This rule is specifically defined for ACM CCS or we can say that this can only be applied to those taxonomies in which general node exists. There are many taxonomies that do not define general node, in that particular case this rule cannot be applied. Also for general node mostly the term *General* is used but this term appears at multiple places in ACM CCS like *General* is aligned at *B.2.0, B.3.0, B.4.0* etc, so to which parent node this should be stemmed is also not explained in this rule.

- **Rule IV: Preserving lower level keyphrases:** For situation in which no training-level keyphrase exist in the result set, this rule can be applied. In that scenario keyphrases that are aligned at lower level than the training-level will be preserved in the final result set. This rule works fine and has got no implementation issue.

- **Rule V: Identifying and preserving training-level equivalent keyphrases:** In Kea++ result set, if a keyphrase is aligned at upper level than the training-level and has equivalent training-level keyphrase associated with it in the taxonomy, then replace that keyphrase with its respective equivalent training-level keyphrase, and preserve it in the refined result set. For instance the keyphrase *Control Structures and Microprogramming (B.1)* appears in Kea++ result set and the training-level is three then training-level equivalent keyphrase is looked for this keyphrase in the taxonomy. As *Language Classifications (D.3.2)* exists in ACM CCS which is training-level equivalent to *B.1*, so we will preserve that in the final result set. However this rule does not consider keyphrases that have multiple equivalent keyphrases associated with them in the taxonomy. For instance, the keyphrase: *Files* having identifier *E.5* has three equivalents keyphrases i.e. *D.4.3, F.2.2* and *H.2.* How to deal with this scenario is not clearly explained in this rule.

- **Rule VI: Removing redundant keyphrases:** As an application of rule III and V there might exist some repeating keyphrases in the refined result set so they may be removed by applying this rule.

After identifying the problem areas we will move towards the extended refinement rules. Then based on these rules, the extended refinement algorithm will be developed.

4.2 Proposed Methodology

In this section, we will first select some relations that are common in different taxonomies. Then extended refinement rules and algorithm will be proposed.

4.2.1 Dealing with Different Taxonomies

It should be noted that while assigning keyphrases to document from taxonomy, Kea++ uses either the SKOS version or the text version of the taxonomy. SKOS is the preferred version when performing the keyphrase assignment task, because semantic relations among terms are better represented in SKOS than text format. In their work, [5] selected relations that are being important in SKOS conversion of any taxonomy and we will use these relations to convert our refinement rules to generalized refinement rules. For any vocabulary the main building blocks are *Concept/Preferred Terms/Terms,* which are identified by preflabel in SKOS and two relations between them: *Broader Term (BT) and Narrower Term (NT).* Because of structural differences between different domain taxonomies, these relations are at times not easy to identify. In case if the domain specific taxonomy structure is complicated and not compliant with the standards, it has been suggested to convert it according to the method proposed in [5] and guidelines and standards mentioned in [35]. So before formulating extended refinement rules it should be made clear that the word keyphrase is the concept or preferred term of the taxonomy and we are dealing with the BTand NT relations that exist between the concept or preferred terms .

We have also studied and analyzed various example taxonomies from [2]. These taxonomies belong to different domains including computing [1], agriculture [2], alcohols and drugs [3], art and architecture [4], forestry [5], health sciences [6], engineering [7], mathematics [8], management and business [9]. Properties and structural behavior of these taxonomies differ from each other. We tried to find the commonality between them and accordingly categorized the taxonomies based on the following properties:

- Main categories/Top nodes
- Total terms count
- Hierarchical level

Based on the range of values possessed by these properties, we divide the tax-

[1] http://www.acm.org/about/class/1998/ ; Last viewed : 31 October, 2011

[2] http://aims.fao.org/website/AGROVOC-Thesaurus/sub ; Last viewed : 31 October, 2011

[3] http://etoh.niaaa.nih.gov/AODVol1/Aodthome.htm ; Last viewed : 31 October, 2011

[4] http://www.getty.edu/research/tools/vocabularies/aat/ ; Last viewed : 31 October, 2011

[5] http://ec.europa.eu/eurostat/ramon/nomenclatures/index.cfm?TargetUrl=LST_NOM_DTL &StrNom=FORESTPROD& StrLanguageCode=EN&IntPcKey=&StrLayoutCode=HIERARCHIC ; Last viewed : 31 October, 2011

[6] http://decs.bvs.br/I/homepagei.htm

[7] http://www.csa.com/factsheets/mechtrans-set-c.php ; Last viewed : 31 October, 2011

[8] http://www.ams.org/mathscinet/msc/msc2010.html ; Last viewed : 31 October, 2011

[9] http://libraryds.grenoble-em.com/en/training/Pages/Thesaurus_Management.aspx ; Last viewed : 31 October, 2011

Table 4.1: Categorization of taxonomies

	Small scale	**Medium scale**	**Large scale**
Total terms count	1-999	1000-5000	>5000
Main categories/Top nodes	1-4	5-15	>15
Hierarchical level	1-2	3-5	>5

onomies into three categories ; *small scale*, *medium scale* and *large scale* taxonomies as shown in table 4.1.

4.2.2 Extended Refinement Rules

Following are the proposed extended refinement rules:

- **Rule I Adopting training-level:** "The level in the taxonomy, upon which maximum numbers of keyphrases in the training dataset are aligned, should be adopted as the training-level for the dataset. If a keyphrase is aligned at multiple levels in the taxonomy, then it is considered to be appearing only once at each level, despite its occurrence at a level more than one in the taxonomy."

This proposed rule gives priority to the level upon which most of the keyphrases are aligned in the training dataset. Consider a training dataset comprising of 50 documents and their manually assigned keyphrases. Hierarchical level of each keyphrase in the training dataset is found out from the taxonomy for calculating the training-level. The number of keyphrases aligned at level 1, 2, 3, 4 and 5 are 25, 67, 89, 26 and 0 respectively. We adopted training-level as three since the maximum number of keyphrases in the training dataset are aligned at level three. Moreover the keyphrase *Distributed Systems* that is aligned at *C.2.4* having level 3, aligned under *D.4.7* at level 4 and aligned under *H.3.4* at level 4. In this case, the keyphrase is considered to be appearing at level 3 and 4 once, despite its occurrence at level 4 twice in the taxonomy. This is because if multiple occurrence of the keyphrase at a level is considered more than once, then this could result into misleading value of training-level in the end. Finally when all the keyphrases in the training dataset are checked for the level upon which they are aligned, the level having maximum number of keyphrases aligned to it is opted as the training-level.

- **Rule II: Preserving training-level keyphrases:** “Keyphrases that are aligned at training-level will be preserved in the final result set.”

Like the refinement rules, the extended refinement rules will also preserve the training-level keyphrases.

- **Rule III: Discarding narrower level keyphrases:** “If the result set contains training-level keyphrases, then all those keyphrases that are aligned at narrower level than the training-level will be discarded despite their existence at the general node.”

During the analysis of the taxonomic structure of various taxonomies, it was observed that rule III of the refinement rules is only applicable to some taxonomies. Also more commonly the term *General* is used to represent general node in the taxonomies that implement general node, for instance, ACM CCS. In ACM CCS the term *General* is present at multiple places like *B.2.0, B.3.0, B.4.0* etc. Assume that the training-level is two then situation like this will ultimately lead to a confusing state of which training-level parent keyphrase it should be stemmed to. That is why this rule is simplified to discard all the narrower level keyphrases despite their alignment at the general node or not.

- **Rule IV: Preserving keyphrases narrower than training-level:** “If the result set does not contain training-level keyphrases, then keyphrases aligned at narrower level than the training-level will be preserved in the final result set.”

This rule is similar to rule IV of the refinement rules and enables the selection of appropriate keyphrases in the absence of training-level keyphrase. However for generalizing purpose, the word: lower level is replaced with narrower, which is more commonly used for referring to the child relation in the hierarchical parent-child semantic relation that exist between terms in the taxonomy.

- **Rule V: Identifying and preserving keyphrases equivalent to broader level keyphrase:** “In the result set, if a keyphrase is broader than the training-level and has single or multiple equivalent training-level keyphrases associated with it in the taxonomy, then replace that broader keyphrase with its respective equivalent training-level keyphrases, and preserve them in the refined result set.”

The execution of this rule follows the same procedure, in which keyphrases broader than the training-level are replaced by the keyphrases that are equivalent to the broader keyphrases, but are aligned at the training-level. In the final result set equivalent training-level keyphrase will be preserved. It is important to mention that one keyphrase can have multiple equivalent keyphrases and we will preserve all of those equivalent keyphrases which are aligned at training-level.

- **Rule VI: Selecting the training-level common parent as keyphrase:** "If the result set contains no training-level keyphrases and if the taxonomic structure has large number of hierarchical levels (greater than five), then rule VI is applied. According to this rule, keyphrases having training-level common parent are replaced by the training-level common parent. In the final result set training-level common parent will be preserved as keyphrase."

Consider an example agricultural result set which contains two terms *Deforestation* and *Reforestation*. Their hierarchies in Agrovoc are *activities/management/ resource_management/Natural_resources_management/Land_management/ Land_clearing/Deforestation* and *activities/management/Forest_management /Forestation/Reforestation* respectively. Suppose that the training-level is two. Then Management is the training-level common parent between Deforestation and Reforestation. So in the final result set keyphrases Deforestation and Reforestation are replaced by the keyphrase *management*.

- **Rule VII: Removing redundant keyphrases:** "Keyphrases that are redundant will be removed from the final set."

After the execution of rules there is a possibility of repetition of some keyphrases in the result set produced by extended refinement algorithm. In that case preserve only one keyphrase and discard the remaining if they are repeated. This step eliminates the redundancy in the keyphrase final result set.

4.3 Extended Refinement Algorithm (ERA)

The rules have been proposed in the previous section. In this section we describe the proposed algorithm for the flow of these rules. Extended refinement algorithm describes a set of steps that are performed to get the refined set of semantic keyphrases.

Table 4.2: Kea++ parameters setting

Parameter	Value	Comment
Vocabulary name	acm/agrovoc /mesh etc	specifies the name of the taxonomy
Vocabulary format	SKOS	txt format can also be supported
Vocabulary encoding	UTF-8	UTF-8 documents are easy to process
Document language	en	en is the code for English language
Maximum length of phrases	5	usually 5-6 is the sufficient length
Minimum length of phrases	2	
Minimum occurrence	2	this value produce better results
Number of extracted keyphrases	10	

4.3.1 Steps of Extended Refinement Algorithm

1. **Kea++ parameters setting:** There are some parameters that need to be set for better results before running Kea++ on the dataset. The parameters setting as it was done in [13] is applicable for extended refinement algorithm as well. Table 4.2 lists these parameters and their respective values. Details of what these parameters are and why they are assigned these values is available in Appendix B.

2. **Apply Kea++:** Kea++ is applied on the test dataset giving train dataset and domain specific taxonomy as input.

3. **Adopt the training level:** Read each file having manually assigned keyphrases in the training dataset and find out the level(s) associated with each keyphrase. There is a counter variable associated with each hierarchical level of the taxonomy, taking care of the number of keyphrases aligned at each level. Each keyphrase is read and its level is found out from the taxonomy. Respective counter associated to the level gets incremented. If the keyphrase is aligned at multiple levels then respective counters will get incremented only once. The algorithm for adopting the training-level is given in algorithm 4.1.

4. **Find the final result set:** Each test document containing Kea++ as-

Algorithm 4.1 Algorithm for adopting the training-level

```
input: Training dataset, Manual keyphrases, Taxonomy & Taxonomy maximum hierarchical level
1. Declare 1,....i number of counters and initialize each counter with 0 // i = Maximum number of levels in the taxonomy
2. Find level of each keyphrase in Kea++ training dataset repeat
{
If (Keyphrase is aligned at single level) then increment the respective counter
Else If (Keyphrase is aligned at multiple levels) then increment respective counters only once
} for each keyphrase
3. Adopt counter having maximum value as training-level
output: Training-level
```

signed keyphrases in the test dataset will be read and refinement algorithm will be applied on them. It is shown in algorithm 4.2.

4.3.2 Description of Extended Refinement Algorithm

After Kea++ installation and parameters setting according to the values provided in table 4.2, Kea++ is applied to produce keyphrases for test dataset. Kea++ produces keyphrases for each test document but these keyphrases contain noise and irrelevant terms. The objective of refinement algorithm (RA) and its extension is to eliminate these irrelevant terms from the result set and produce accurate and relevant keyphrases as output. Before looking into the details of the extended refinement algorithm it is important to mention here that both refinement algorithm and its extension by no means influence the working of Kea++, rather they serve as an upper layer over Kea++. They are used to fine tune the results of Kea++, moreover it can be used for any tool that can perform keyphrase alignment from the domain specific taxonomy and is based on machine learning approach.

The manually assigned keyphrases in the training dataset are read by extended refinement algorithm in order to calculate the training-level which is the fundamental parameter for the preservation or elimination of the keyphrase. Counter variables are used for each level of domain specific taxonomy. For instance if the domain taxonomy comprises four levels then there will be four counter variables associated with each level. One by one each and every manually assigned keyphrase is read, its level in the taxonomy is noted and the associated counter variable is incremented. If the keyphrase is aligned at multiple levels like

Algorithm 4.2 Algorithm for finding final refined result set

```
input: Test dataset, Training-level & Taxonomy
1. If test document contains (keyphrases narrower OR broader than training-level) AND contains (keyphrases equal to training-level) then
    a) If (Levels of keyphrases are equal to training-level) then preserve training-level keyphrases in the refined result set
    b) Else If (Levels of keyphrases are broader than training-level) then identify and preserve their equivalent training-level keyphrases (single or multiple) in the refined result set
    c) Else If (Levels of keyphrases are narrower than training-level) then discard narrower keyphrases
2. Else If test document contains (keyphrases narrower OR broader than training-level) AND NOT contain (keyphrases equal to training-level) then
    a) If (Levels of keyphrases are narrower than training-level) then preserve narrower keyphrases in the refined result set
    b) Else If (Levels of keyphrases are broader than training-level) then identify and preserve their equivalent training-level keyphrases (single or multiple) in the refined result set
    c) If (taxonomy hierarchical level $> 5$) then replace keyphrases having training-level common parent with the training-level common parent and preserve training-level common parent in the refined result set
4. Remove redundant keyphrases from the refined result set
5. Return the final result set of keyphrases
output: Refined keyphrases result set
```

keyphrase: *Distributed Systems*, that exists at *C.2.4* having level 3, under *D.4.7* at level 4 and under *H.3.4* at level 4 then respective counters are incremented only once. As here in this example, counters associated with level 3 and level 4 are incremented only once despite the occurrence of this keyphrase at level 4 twice in the taxonomy. Finally the level associated with the counter having maximum value is selected as the training-level. The algorithm then reads keyphrases result set produced by Kea++, extended refinement algorithm processes the keyphrase result set based on the presence or absence of the training-level keyphrases. If the result set contains training-level keyphrases then all those keyphrases that are aligned at training-level are preserved in the final result set. For keyphrases aligned at broader level than training-level equivalent keyphrases are checked in the taxonomy. Those equivalent keyphrases are selected which are aligned at the training-level and the rest of them are discarded. All those keyphrases that are aligned at narrower level than the training-level are simply discarded, and will not be included in the final refined result set.

The next part of the algorithm deals with the situation when the result set contains no keyphrases aligned at training-level. In that case all keyphrases aligned at narrower level are preserved in the final result set. For the keyphrases aligned at broader level than the training-level, same strategy follows as it was followed in the presence of the training-level aligned keyphrases. For broader level keyphrases associated equivalent keyphrases in the taxonomy are looked for. All those equivalent keyphrases are preserved that are aligned at training-level and the rest of them are discarded. Now if the taxonomy hierarchical level is greater than five then the keyphrases possessing training-level common parents are replaced with the training-level common parent keyphrase. In the refined result set, the training-level common parent will be preserved. Finally all redundant keyphrases are removed from the final result set.

Flowchart for extended refinement algorithm can be seen in figure 4.1. Kea++ processes the test documents and assigns keyphrases to them. The results produced by Kea++ for each test document are taken as input by the extended refinement algorithm. It reads each and every keyphrase, finds out the level for the keyphrase from the taxonomy. Then treats the keyphrase according to the extended refinement rules.

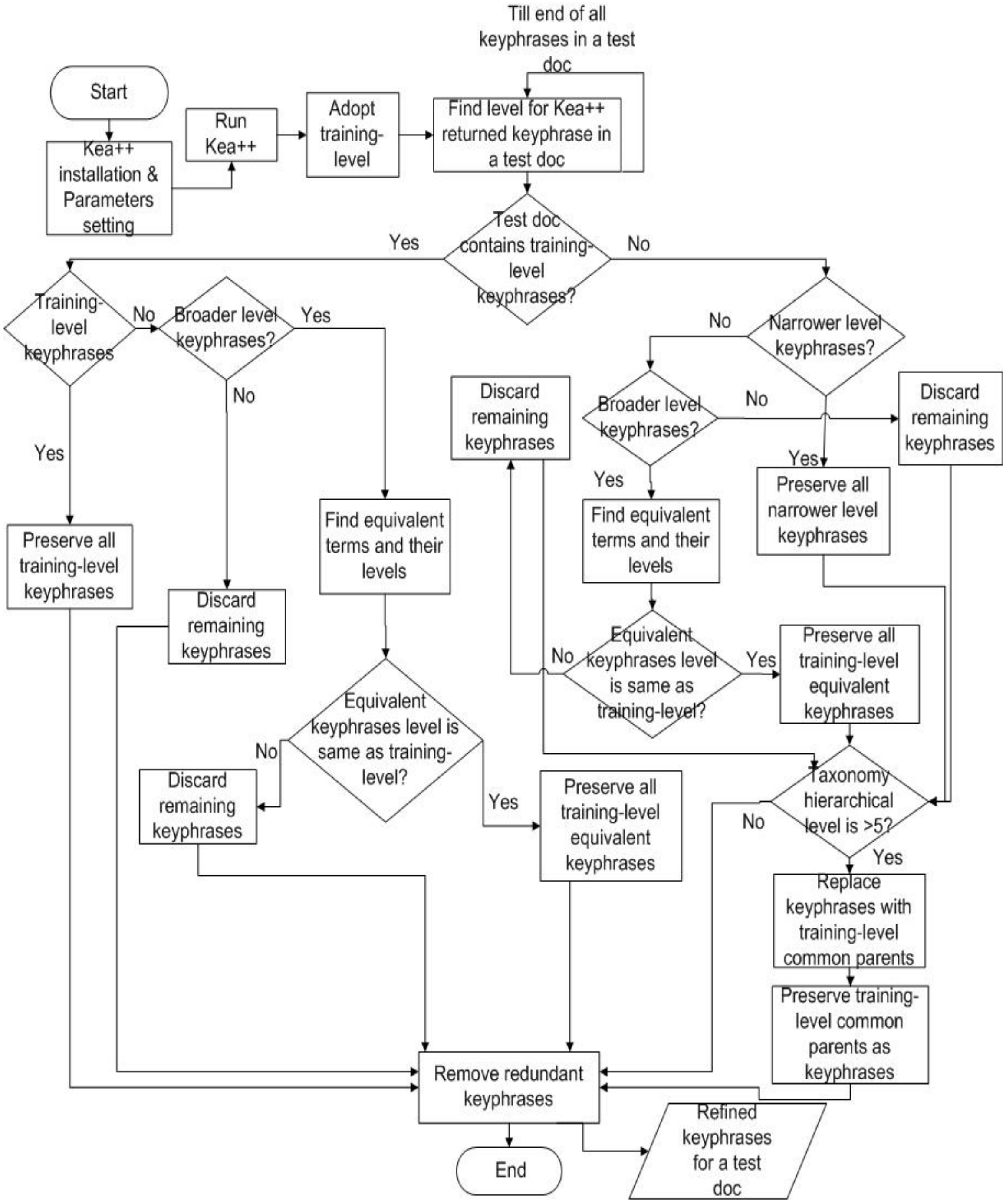

Till end of all keyphrases in a test doc
Start
Kea++ installation & Parameters setting
Run Kea++
Adopt training-level
Find level for Kea++ returned keyphrase in a test doc
Test doc contains training-level keyphrases?
Yes
No
Training-level keyphrases
No
Broader level keyphrases?
Yes
No
Yes
Preserve all training-level keyphrases
Discard remaining keyphrases
Find equivalent terms and their levels
Equivalent keyphrases level is same as training-level?
No
Yes
Discard remaining keyphrases
Preserve all training-level equivalent keyphrases
Narrower level keyphrases?
No
Yes
Discard remaining keyphrases
Broader level keyphrases?
No
Yes
Discard remaining keyphrases
Preserve all narrower level keyphrases
Find equivalent terms and their levels
Equivalent keyphrases level is same as training-level?
No
Yes
Preserve all training-level equivalent keyphrases
Taxonomy hierarchical level is >5?
No
Yes
Replace keyphrases with training-level common parents
Preserve training-level common parents as keyphrases
Remove redundant keyphrases
Refined keyphrases for a test doc
End

Figure 4.1: Flow chart for Extended Refinement Algorithm

Table 4.3: Extended Refinement Algorithm example 1

Title	A Survey on the Design, Applications, and Enhancements of Application-Layer Overlay Networks	
Source	ACM Computing Survey	
Manual keyphrases	Network Architecture and Design ; Network Operations	
Training level	3	
Kea++ keyphrases	**Keyphrases Level id (level)**	**Selected keyphrases**
Network Operations	C.2.3 (3)	Network Operations
Routing Protocols	under C.2.2 (4)	
Network Protocols	C.2.2 (3)	Network Protocols
Distributed Systems	C.2.4 (3) ; under D.4.7 (4); under H.3.4 (4)	Distributed Networks
Distributed Networks	under C.2.1 (4)	
Operating Systems	D.4 (2)	
Network Topology	under C.2.1 (4)	
Network Communication	under D.4.4 (4)	

4.4 Walk-through Examples

Three walk-through examples will be discussed; two examples are taken from computing domain and the other is taken from agricultural domain.

4.4.1 Walk-through example 1

The document used in this example belongs to computing domain. The extended refinement algorithm first calculates the training-level. For the dataset to which this document belongs, most keyphrases in the training set of documents are aligned at level 3, so the value of training-level is 3. On the basis of this extended refinement algorithm, apply the rest of the rules. Level calculated by the algorithm for each keyphrase is shown in parenthesis in the second column along with the level identifier in table 4.3. We can see that Kea++ result set for the document contains training-level keyphrases. So from the keyphrases assigned by Kea++, the extended refinement algorithm picks *Network Operations*, *Network Protocols* and *Distributed Systems,* as they are aligned on the training-level. Narrower than training-level keyphrases:*Routing Protocols, Distributed Networks, Network Topology* and *Network Communication* are simply discarded. *Operating*

Table 4.4: Extended Refinement Algorithm example 2

Title	A Taxonomy of Sequential Pattern Mining Algorithms	
Source	ACM Computing Survey	
Manual keyphrases	Data Mining ; Marketing	
Training-level	3	
Kea++ keyphrases	**Keyphrases Level id (level)**	**Selected keyphrases**
Data Mining	under H.2.8 (4)	Data Mining
Main Memory	under D.4.2 (4)	Main Memory
Sampling	under I.4.1 (4)	Sampling
Data Structures	E.1 (2)	
Analysis of Algorithms	under I.1.2 (4)	Analysis of Algorithms
Information Systems	H (1)	

Systems is a keyphrase that belongs to broader level than the training level, algorithm looked for its equivalent keyphrase in the taxonomy but couldn't find it so it is also discarded. None of the keyphrases are being repeated in the final result set so rule VII is not applicable here. Now if we compare the results with the manually assigned keyphrases we can see that one match occur i.e *Network Operations*. It is also very obvious that extended refinement algorithm eliminates the irrelevant terms from the Kea++ result set just like the refinement algorithm do.

4.4.2 Walk-through example 2

Document used in this example also belongs to computing domain. Training-level value is also three here. But this examples shows the scenario when the result set produced by Kea++ possesses no keyphrase aligned at training-level. So all the keyphrases aligned at narrower level than the training-level are preserved. Here *Data Mining, Main Memory, Sampling* and *Analysis of Algorithms* are preserved in the result set of extended refinement algorithm. Now the keyphrases that are aligned at broader level are looked for, we can see that two keyphrases, *Data Structures* and *Information Systems* are aligned at level 2 and 1 respectively which is broader than the training-level. So the algorithm will look for the existence of any equivalent terms for these keyphrases. As none exist, so they are discarded. At the final step, like refinement algorithm, extension of refinement algorithm will also check for the existence of the repeated keyphrases. As none exist here so this rule makes no change in the final result set.

We can see in table 4.4 that one manual match occurs and the number of irrelevant terms from the result set of Kea++ are also reduced.

4.4.3 Walk-through example 3

In order to test the extended refinement methodology on domain other than computing we have taken an example document from agricultural domain. When the extended refinement algorithm was run on the training dataset, the training-level value calculated was three. As it is the level upon which most of the keyphrases in the training dataset are aligned. Next the algorithm will check for the availability of training-level keyphrase in the Kea++ result set. *Harbours* and *fishes* are the two keyphrases that are aligned at training-level so they are being preserved in the final result set. Keyphrases aligned at narrower level are simply discarded. Finally for the keyphrases that occurs at broader level in the taxonomy extension of refinement algorithm will check for their equivalent keyphrases. For two keyphrases: *Smoked fish* and *Retail marketing*, equivalent keyphrases exist. For *Smoked fish* there is only one equivalent keyphrase which is *Smoking* and for *Retail marketing* there is one too which is *Shops*. Both of them are aligned on the training-level so in the final result set we will preserve them. Note that if more than one equivalent keyphrases exist that are aligned at training-level, we preserve them all in case of the extended refinement algorithm. Rule related to the elimination of redundant terms from the final result set is not applicable in this example. We know that Agrovoc is the taxonomy with hierarchical levels greater than five, but the particular example used contains training-level keyphrases in Kea++ result set. So replacing the keyphrases having training-level common parent with the training-level common parent will not be applied here. When final result set are compared with the manual keyphrase in table 4.5, we can see that one manual match occur and number of irrelevant terms are also less as a result of extended refinement algorithm.

4.5 Implementation

In this section we will briefly look into the implementation details of the extended refinement algorithm. System architecture and tools used in the implementation will be explored.

Table 4.5: Extended Refinement Algorithm example 3

Title	A Study of the trade in smoked-dried fish from West Africa to the United kingdom	
Source	FAO	
Manual keyphrases	Fish products ; Processed products ; Smoked fish ; Smoking ; West Africa	
Training level	3	
Kea++ keyphrases	**Keyphrases Level id (level)**	**Selected keyphrases**
Smoked fish	7115 (1)	Smoking
Smokes	7117 (5)	
fishes	2943 (3) ; 2943 (4) ; 2943 (4)	fishes
Certification	35702 (2)	
Retail marketing	6536 (1)	Shops
Nigeria	5182 (7)	
Ghana	3253 (7)	
Cold stores	1744 (1)	
Harbours	3492 (3)	Harbours
Infestation	3855 (4)	

4.5.1 System Architecture

The system architecture used is somewhat similar to the one used in [13]. The system architecture is shown in figure 4.2. The semantic keyphrase alignment module is the main module used which has two sub modules: Kea++ and Extended Refinement Module. Kea++ sub module needs set of training documents, test documents and domain specific taxonomy to perform its task. The output produced by this sub module is provided as input to Extended Refinement Module. Extended Refinement sub module needs train and test datasets as input. Also the information stored in the database related to the taxonomy will be accessed by Extended Refinement Module to produce set of refined keyphrases. The final result set produced by the Extended Refinement Module for each document is stored in the database, so that it can be used later on to facilitate the searching process.

4.5.2 System Requirements

The system specifications used for the implementation of the extended refinement algorithm were:

Processor: 2.6 GHz Intel Pentium IV or equivalent

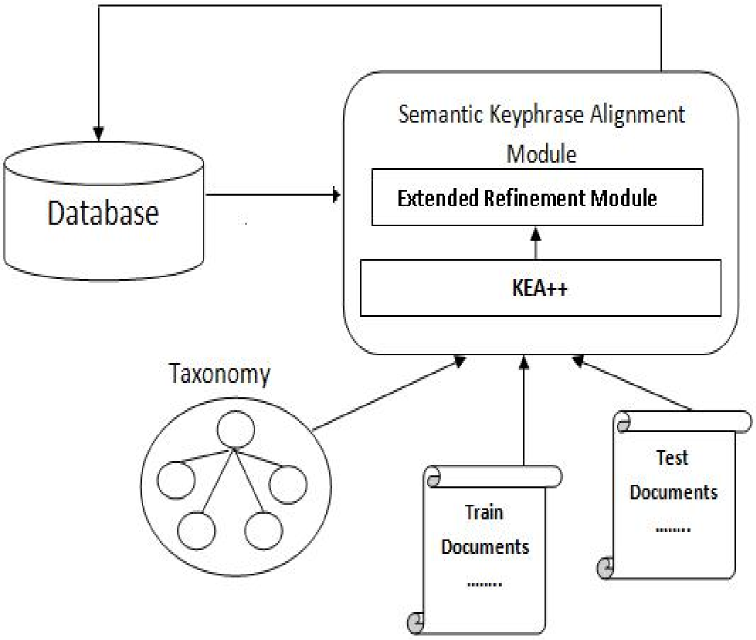


Figure 4.2: System Architecture for Extended Refinement Algorithm [13]

Memory: 2 GB
Disk space: 4 GB of free disk space

4.5.3 Tools and Installation

Tools used were:

1. **Kea5.0:** Kea++ can be easily downloaded from Kea home page[10]. Installation instructions can also be found from Kea home page[11]. Kea5.0 was the tool used to implement the first sub module of semantic keyphrase alignment module shown in figure 4.2.

2. **Java Netbeans 6.9.1:** Java Netbeans 6.9.1 was used for the implementation of the extended refinement algorithm. Netbeans can be downloaded from *http://netbeans.org/community/releases/69/* and the instruction for installation is also available from this link.

3. **MySQL:** MySQL is an optional tool used for storing information related to taxonomy that are used by the extended refinement algorithm. Refined keyphrases are also stored in the database for their use in future. This

[10] http://www.nzdl.org/Kea/download.html
[11] http://www.nzdl.org/Kea/Download/Kea-5.0-Readme.txt

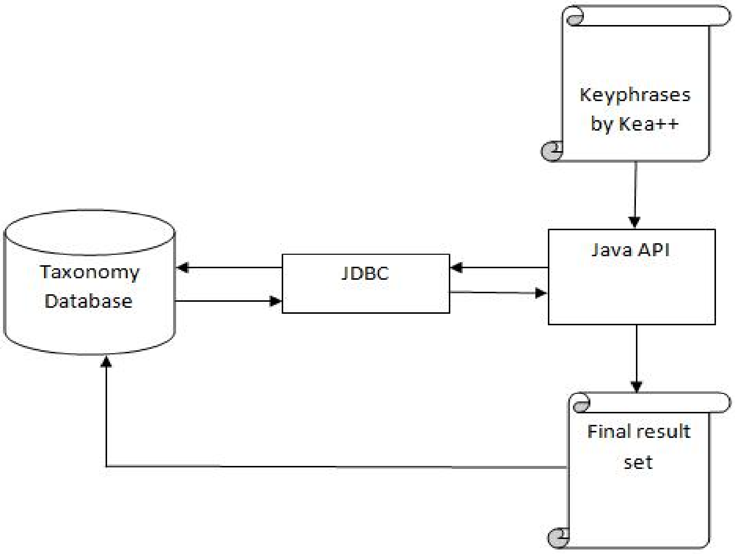


Figure 4.3: Implementation of Extended Refinement Algorithm (option 1)

can be available from *http://dev.mysql.com/downloads/* and can be easily downloaded as well.

4.5.4 Implementation of Extended Refinement Algorithm

There are two options available for the implementation of the extended refinement algorithm.

For the implementation of the extended refinement algorithm we have adopted option 1 as shown in figure 4.3 . A small database comprising the necessary information related to taxonomy has been developed for the taxonomies used in the testing of the algorithm. This information comprises the level, broader, narrower and equivalent terms for each term in the taxonomy. Mostly taxonomy is available in many formats like .rdf, .mdb, .sql etc. So the development of the small and specific database comprising such information is not very tedious. This information is accessed by Java API through JDBC. Keyphrases produced by Kea++ is taken as input. They are being refined by extended refinement algorithm to produced final result set.

Another option shown in figure 4.4 access the information from the .rdf file that has been used by Kea++ for assigning keyphrase to the documents. But sometimes the information needed by the extended refinement algorithm is not easily derivable from the rdf format. So we consider it better to develop the specific database.

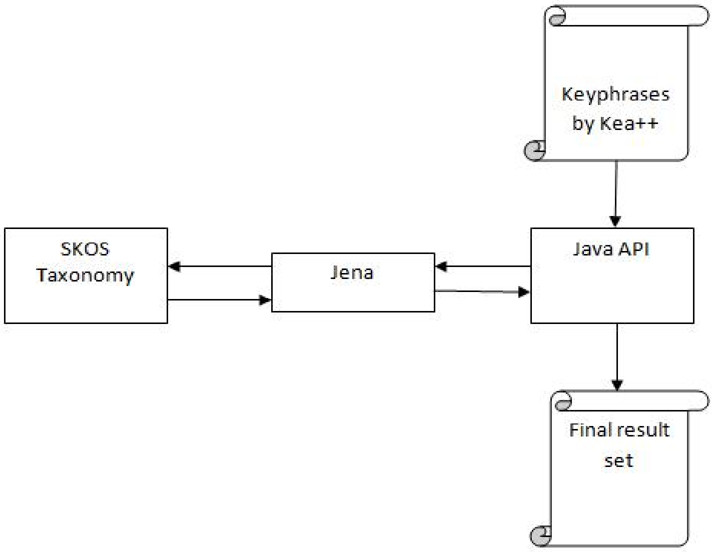


Figure 4.4: Implementation of Extended Refinement Algorithm (option 2)

Briefly, in order to generalize and extend the already existing refinement methodology, different taxonomies were analyzed. Based on this analysis problems were identified in the refinement rules. Extended refinement rules were formulated. Different approach has been adopted for the selection of training-level. Based on these rules the extended refinement algorithm was developed. The extended refinement algorithm is a simple program developed in Java Netbeans 6.9.1. MySQL database has been used for storing the information related to taxonomy. The keyphrases were preserved and discarded according to the level of their occurrence in the taxonomy. This way the extended refinement algorithm removes irrelevant terms from the result set produced by Kea++. We have also discussed few walk-through examples that have made the methodology working more clear. In the next chapter we will evaluate the results produced by Kea++, refinement algorithm and extended refinement algorithm to check for any improvement made by the proposed methodology.

Chapter 5

EVALUATION

In the previous chapter we proposed the extension of refinement methodology and discussed the implementation details. In this chapter we will finally test the proposed methodology and analyze the results obtained. Various tests will be performed on datasets from three different domains: computing, agriculture and mathematics. Refinement methodology was already tested on computing domain in [12] showing good results but its application on dataset of domain other than computing is yet to be tested. We will also present the comparative analysis of all three approaches i.e Kea++, refinement methodology and extension of refinement methodology.

5.1 Test dataset description

The datasets used in this testing belong to three separate domains, one from the computing domain, other from agriculture domain and the last one from mathematics domain. For computing domain, dataset was taken form ACM Computing Survey[1]. Four hundred pdf documents were randomly picked from ACM Computing Survey archive along with manually assigned keyphrases placed in separate key file. After converting them into text format they were divided randomly into three sets. First set Set-A comprised 175 train documents and 25 test documents. Second set, Set-B comprised 175 train documents and 50 test documents and the third set Set-C comprised 300 train documents and 50 test documents. The taxonomy used for testing was ACM CCS. Same testing was

[1] http://portal.acm.org/citation.cfm?id=J204&picked=prox

performed on all three datasets separately.

For agricultural domain 400 documents were downloaded from FAO[2] along with manually assigned keyphrases placed in separate key files. They were randomly divided into five sets. First set comprised 100 train and 15 test documents (Set-A). Second set comprised 175 train and 25 test documents (Set-B). Third set comprised 175 train documents and 50 test documents (Set-C). For fourth set number of training documents increased to 300 and test documents remained the same as previous test (Set-D). For fifth test 300 train and 100 test documents were used (Set-E). Taxonomy used in these tests was Agrovoc. The testing for agriculture domain dataset was done in two phases. Initially datasets were tested without including rule VI. Then rule VI was tested to check for any improvement obtained.

For testing the methodology on mathematics domain, dataset was taken from the Journal of the American Mathematical Society [3]. 115 documents along with manually assigned keyphrases were downloaded. Randomly 100 documents were selected to make the train dataset while 15 documents were selected as test dataset (Set-A).

Each of the datasets belonging to any domain will be reshuffled thrice and will undergone the same test three times to get the accurate value of the evaluation metrics.

5.2 Evaluation Metrics

The evaluation metrics used for evaluating the extended refinement algorithm are *precision,recall* and *f-measure*. As they are the most commonly used metrics in the field of information retrieval. There are many other reasons for opting them as suitable metrics for performance evaluation which will be discussed in the later section. Its important to look at the result from the other perspective so we will also compare the average number of keyphrases assigned to test documents by Kea++, refinement algorithm and extended refinement algorithm to manually assigned keyphrases. In order to check for the elimination of noise and irrelevant terms. Finally for testing the significance of the results obtained t-test has been performed on dataset of all three domains.

[2] http://www.fao.org/Documents/index.asp?lang=en

[3] http://www.ams.org/publications/journals/journalsframework/jams

5.2.1 Precision, Recall and F-measure

The evaluation metrics used to measure the performance of the proposed methodology are very commonly used and widely understood metrics of *Precision (P)*, *Recall (R)* and *F-measure (F)*. They are basically used to compare the expected result and the effective result of the system under evaluation [11]. Generally precision can be defined as the share of real matches among all found ones [9] while recall shows the share of real matches that is found [9]. Despite their flexibility and understandability they also have certain disadvantages. It is claimed that it is easy to maximize precision at the expense of recall by returning the single most promising match [48]. Similarly recall can be maximized at the expense of precision by returning all the matches [48].

However there are two reasons of their usage in our work. First of all, as stated that in the field of information retrieval they are the most commonly used metrics. So by using them we can very clearly state the effectiveness of the result obtained. Secondly they have been used by the work [12] with which we will compare the result to check for any improvement achieved.

As mentioned these measures are flexible enough to be mold according to the requirement of the particular situation in information retrieval. They have been used by [9] for ontology alignment evaluation. [26] have adapted them to evaluate various approaches of automated indexing technique. In our case we will use them to compare the keyphrases returned by Kea, refinement and its extension with the manual assignment. We can define them as follows:

Precision is the ratio of relevant keyphrases retrieved out of total keyphrases retrieved. We compute precision as follows:

$$Precision = \frac{TruePositive}{(TruePositive + FalsePositive)}$$

Recall is the ratio of relevant keyphrases retrieved out of all relevant keyphrases related to a document. We compute recall as follows:

$$Recall = \frac{TruePositive}{(TruePositive + FalseNegative)}$$

TruePositives (TP) are the terms that are extracted and relevant to the document. FalsePositives (FP) are the terms that are extracted but not relevant

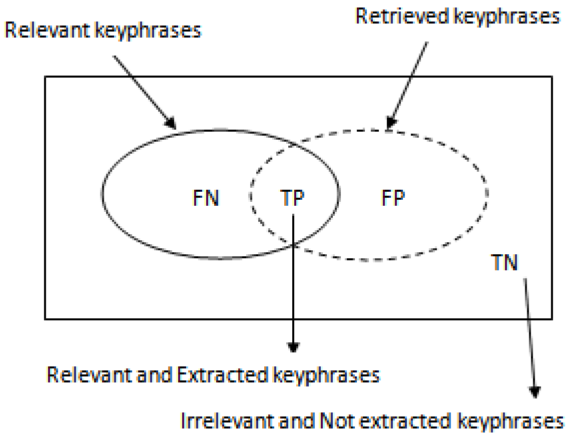

Relevant keyphrases
Retrieved keyphrases
FN
TP
FP
TN
Relevant and Extracted keyphrases
Irrelevant and Not extracted keyphrases

Figure 5.1: Figure showing TP, FP, FN and TN

to the document. FalseNegatives (FN) are the terms that are not extracted but are relevant to the document. They can be shown in figure 5.1.

Finally f-measure is one single measure to check the accuracy and relevancy of result. F-measure is defined as the harmonic mean of precision and recall. We compute f-measure as follows:

$$F - measure = 2 * \frac{(Recall * Precision)}{(Recall + Precision)}$$

5.2.2 Average number of assigned keyphrases

The other side of looking at the result is to check whether refinement and its extension has reduced the noise and irrelevant terms from Kea++ result set or not. For this reason we will consider the average number of keyphrases that are produced by Kea++, refinement algorithm and extended refinement algorithm for test documents in comparison with manually assigned keyphrases for checking the elimination of noise and irrelevant terms.

5.2.3 t-test

Despite the usage of precision and recall very commonly in the evaluation of information retrieval systems, they also posses certain drawbacks which are already mentioned in section 5.2.1. Even if the results obtained are showing improvement it might be possible that it has occurred by chance for the sample under test. Various statistical tests are there for determining that the differences in performance between retrieval methods are significant or not [20]. The t-test is the statistical test for the mean of the population. It is applied in the situation where the population curve is normally distributed or approximately normally distributed, population standard deviation is unknown and the sample size is less than 30 [7]. There are many flavors for t-test depending upon the property of the sample under test. We use the t-test for dependent sample. Dependent sample is the sample which has undergone some new test or procedure in order to increase or decrease certain desirable property [7]. For instance in our case we are giving the same dataset to Kea++, refinement and its extension to check for the improvement in the relevancy of the keyphrases obtained. Statistical t-test is based upon the null hypothesis which assert the true state of the sample in which it exist before applying any new procedure or technique [40]. There exist a claim which can be rejected or accepted based on the value obtained from t-test in comparison to the critical value. Critical value is the value that test statistics should yield if the null hypothesis is indeed true [40]. There is also a concept of level of significance which indicates the maximum probability that null hypothesis is rejected despite the fact that its true. It is indicated by Greek letter alpha. In all the tests we are performing t-test on two levels of significance i.e. 0.05 and .005. For 0.05 we can say that the probability of rejecting the null hypothesis despite the fact that it is true is 10% and 90% chances are there that null hypothesis is accepted as it is true [7]. At 0.005 we can say that the probability of rejecting the null hypothesis despite the fact that it is true is 1% and 99% chances are there that null hypothesis is accepted as it is true. The t Distribution table used to find the critical value can be found in Appendix C. The formula for the t-test is as follows:

$$t = D_M - \mu/(s/\sqrt{n})$$

where,
D= difference of two samples
$\mathrm{D_M}$= mean of the difference of two samples
µ= hypothesized difference
s= sample standard deviation
n= size of the sample
The formula for calculating the standard deviation is given as under:

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

5.3 Test Results

Kea++ is applied on every domain datasets. Later on refinement algorithm (RA) and extended refinement algorithm (ERA) are applied to fine tune the results. In this section we will explore whether the refinement extension has resulted in any improvement as compared to Kea++ and RA or not.

5.3.1 Test results for Computing domain datasets

In this section we will evaluate the proposed methodology using computing domain datasets. This section is divided into three subsections; first section will present the results for precision, recall and f-measure. Next section will evaluate irrelevant terms elimination by calculating average number of assigned keyphrases and lastly we will perform t-test to check the significance of the results obtained.

5.3.1.1 Precision, Recall and F-measure

The precision, recall and f-measure values for all three computing domain datasets are shown in table 5.1. For each set of documents the difference in training-level value in case of refinement algorithm and its extension can be seen from the table. The value for training-level in case of refinement algorithm is four and it remains the same in all three datasets. For extended refinement algorithm this value is three for all three datasets.

We can see in figure 5.2 that the precision value for Set-A is considerably higher in case of ERA as compared to Kea++ and RA. The same pattern has been followed for Set-B and Set-C. RA precision value for Set-A and Set-B is minutely lesser than Kea++ but for Set-C this is equal to Kea++. In short we can say that the precision value obtained as a result of application of ERA on computing domain datasets has improved as compared to not only Kea++ but also the founding refinement algorithm. This interprets that the number of extracted keyphrases that are not relevant to the document is least in case of extended refinement algorithm as compared to Kea++ and even to the founding refinement algorithm.

The recall value in case of RA and ERA are lower when compared with Kea++ as seen in figure 5.2. However the extended refinement algorithm has

Table 5.1: Test results for computing datasets

	Kea++	**Refinement Algorithm**	**Extended Refinement Algorithm**
Set-A (175 train docs & 25 test docs)			
Training-level		4	3
Precision	0.15	0.13	0.38
Recall	0.27	0.14	0.17
F-measure	0.19	0.13	0.23
Set-B (175 train docs & 50 test docs)			
Training-level		4	3
Precision	0.17	0.16	0.36
Recall	0.34	0.16	0.2
F-measure	0.23	0.16	0.26
Set-C (300 train docs & 50 test docs)			
Training level		4	3
Precision	0.16	0.16	0.35
Recall	0.3	0.15	0.18
F-measure	0.21	0.15	0.24

resulted in the improvement in recall value as compared to former refinement algorithm. We can see that for Set-A, Set-B and Set-C recall is higher in case of ERA as compared to RA. This interprets that the keyphrases that are relevant but not extracted is greater in number in case of refinement and its extension when compared with Kea++. This is the improvement area where more work is needed. However precision and recall always counter one another. One measure can be increased or decreased at the expense of the other. That is why we will calculate the f-measure to get the overall picture.

The f-measure value for Set-A, Set-B and Set-C is higher in case of extended refinement algorithm as compared to Kea++ and refinement algorithm as seen from figure 5.2. The f-measure value comes out to be lower for refinement algorithm in comparison to both Kea++ and extended refinement algorithm in all three sets. This shows that the extended refinement algorithm is helpful in improving the overall performance of the system in case of computing domain datasets.

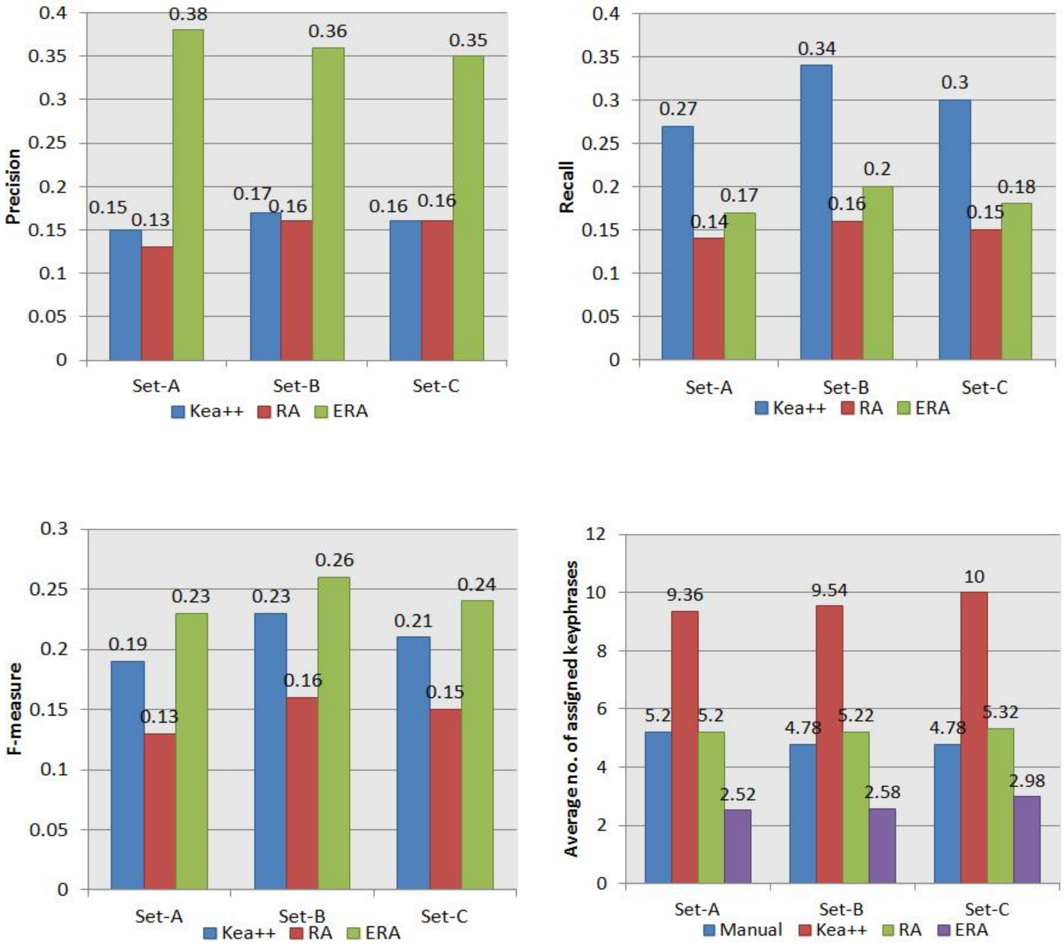


Figure 5.2: Results for computing datasets

5.3.1.2 Average number of assigned keyphrases

Figure 5.2 shows average number of keyphrases produced by Kea++, refinement algorithm and extended refinement algorithm for the test datasets in comparison with manually assigned keyphrases. We can see that average number of keyphrases for the test dataset is lowest in case of extended refinement algorithm. Kea++ produced greater noise and the value for average number of keyphrases is much larger than manually assigned keyphrases. The result produced by refinement algorithm in all three tests is closer to the number of keyphrases produced manually.

5.3.1.3 t-test

We perform t-test on documents of Set-A (175 train docs & 25 test docs). Precision, recall and f-measure value obtained are used by statistical t-test to check

Table 5.2: t-test for computing domain

Metric	SysA	SysB	t-test	Result(0.05; 1.711)	Result(0.005; 2.797)
Precision	Kea++	ERA	-3.82	<<	<<
Precision	RA	ERA	-3.20	<<	<<
Recall	Kea++	ERA	2.68	>>	>>
Recall	RA	ERA	-0.91	>>	>>
F-measure	Kea++	ERA	-1.85	<<	>>
F-measure	RA	ERA	-1.847	<<	>>

"<<" indicates SysB has caused significant improvement over SysA

">>" indicates SysB has not caused significant improvement over SysA

that whether the improvement produced in case of the extended refinement algorithm is by chance or is it the significant improvement? Before applying the t-test on the given sample we assume that the sample under test is approximately normally distributed. In order for ERA to be effective the before values must be significantly less than the after values. So the mean of the differences must be less than zero. This shows that this is a left tailed t-test. The critical value obtained from the t Distribution table at level of significance 0.05 and at n-1=24 for left tail test is -1.711. The critical value obtained from the t Distribution table at level of significance 0.005 and at n-1=24 for left tail test is -2.797.

The calculations for t-test related to computing domain dataset are presented in Appendix D. Table 5.2 shows the results of t-test performed on computing domain dataset at level of significance 0.05 & 0.005.

5.3.2 Test results for Agriculture domain dataset (excluding rule VI)

In this section we will evaluate the proposed methodology using agricultural domain datasets. Like previous section, this section is also divided into three subsections; first section will present the results for precision, recall and f-measure. Next section will evaluate irrelevant terms elimination by calculating average number of assigned keyphrases. In the end we will perform t-test. But it is important here to mention that testing has been done without including rule VI of the extended refinement methodology, which has been introduced for the taxonomies having hierarchical level greater than 5.

Table 5.3: Test results for agriculture datasets (excluding rule VI)

	Kea++	**Refinement Algorithm**	**Extended Refinement Algorithm**
Set-B (175 train docs & 25 test docs)			
Training-level		11	3
Precision	0.21	0.0	0.33
Recall	0.21	0.0	0.15
F-measure	0.21	0.0	0.2
Set-C (175 train docs & 50 test docs)			
Training-level		11	3
Precision	0.17	0.0	0.23
Recall	0.21	0.0	0.11
F-measure	0.19	0.0	0.15
Set-D (300 train docs & 50 test docs)			
Training level		12	2
Precision	0.15	0.0	0.16
Recall	0.23	0.0	0.08
F-measure	0.18	0.0	0.11
Set-E (300 train docs & 100 test docs)			
Training-level		12	2
Precision	0.15	0.0	0.17
Recall	0.16	0.0	0.08
F-measure	0.16	0.0	0.11

5.3.2.1 Precision, Recall and F-measure

The precision, recall and f-measure values for four agricultural domain datasets are shown in table 5.3. For each set of documents the difference in training-level values in case of refinement algorithm and its extension can be seen from the table. The value for training-level in case of refinement algorithm is 11 for Set-B and Set-C. For Set-D and Set-E this value is 12. However the value of training-level in case of ERA is 3 for Set-B and Set-C. For Set-D and Set-E this value becomes 2.

We can find in figure 5.3 that the precision value has been increased in case of ERA in all four tests as compared to Kea++. However the difference in Set-B and Set-C are considerable but it gets less in Set-D and Set-E. Because of very large training-level value in case of RA, we find no results produced. So

RA needed modifications in order to work for other domain taxonomies as it is not showing any result, and that is the very reason of extending the refinement methodology developed earlier. After looking at the results we can say that the number of extracted keyphrases that are not relevant to the document is least in case of the extended refinement algorithm as compared to Kea++.

Recall in case of ERA is lower when compared with Kea++ as shown in figure 5.3. As concluded in computing domain dataset that this part of the work needs improvement. RA is not giving any result for recall as well. This is obvious that the keyphrases that are relevant but not extracted is greater in number in case of the extended refinement algorithm when compared with Kea++. Now we will have a look at the f-measure values to make the final conclusion.

We can see in figure 5.3 for all four tests the f-measure value for the extended refinement methodology is lower to that of Kea++. This shows that unlike computing domain datasets, the extended refinement algorithm is not very helpful in improving the overall performance of the system for agricultural domain datasets. We can say that what is different about this taxonomy is the hierarchical depth which is greater in Agrovoc as compared to computing domain taxonomy. Agrovoc is a large scale taxonomy according to the categorization made earlier in section 4.2.1. So we need to modify extended refinement methodology to be effective for large scale taxonomies as well. For this reason rule VI was included in the extended refinement methodology. Next section will evaluate the results for agricultural domain datasets including rule VI. However we can conclude that the extended refinement methodology at least made the refinement methodology workable for domain other than computing.

5.3.2.2 Average number of assigned keyphrases

The average number of keyphrases assigned to test documents by ERA is lower when compared with manually assigned keyphrases and Kea++, as shown in figure 5.3. We have already concluded that refinement methodology developed earlier needed modification before getting applied on taxonomy other than computing domain, so we will not consider the negligibly small results produced by that for average number of assigned keyphrases to test documents. We can say that ERA resulted in the elimination of noise from the result set, but since this value is too low as compared to manually assigned keyphrases, that is why the overall results produced are not very satisfactory.

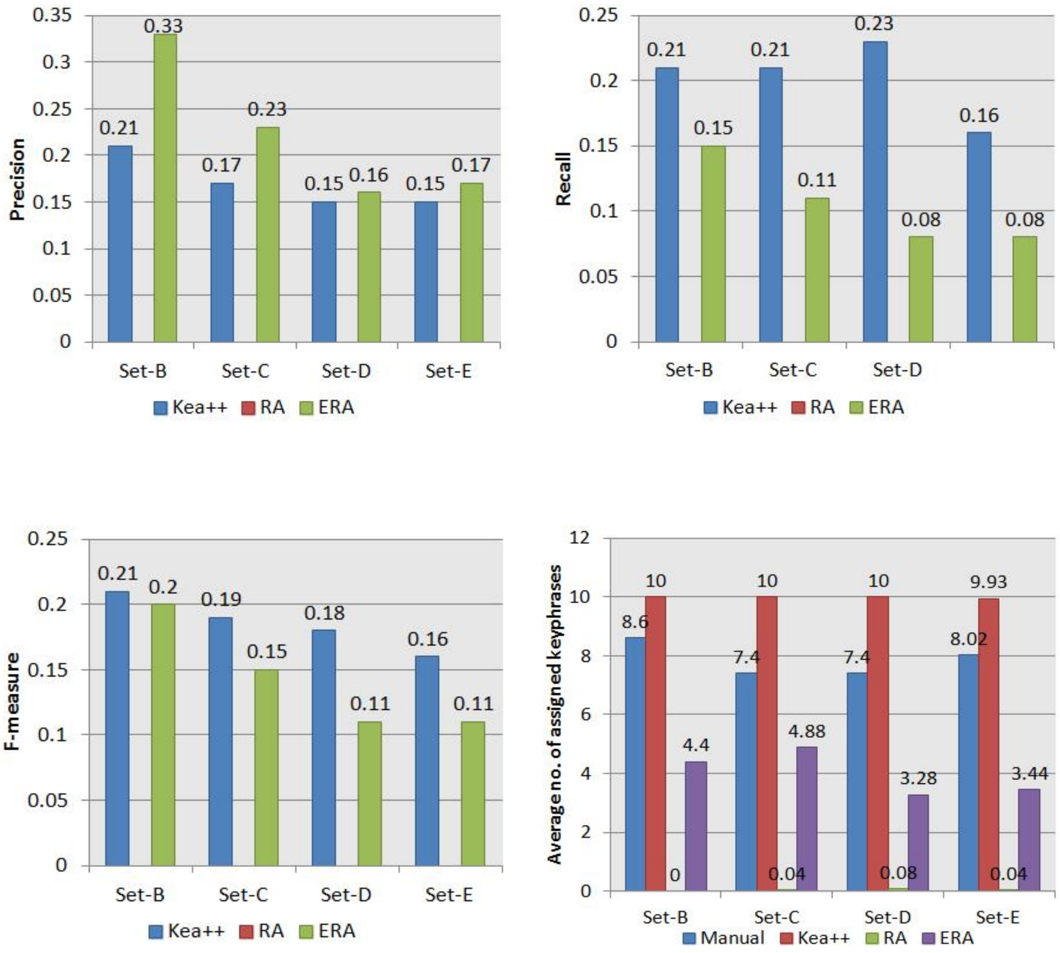


Figure 5.3: Results for agricultural datasets (excluding rule VI)

Table 5.4: t-test for agricultural domain (excluding Rule VI)

Metric	SysA	SysB	t-test	Result(0.05; -1.711)	Result(0.005; -2.797)
Precision	Kea++	ERA	-1.96	<<	>>
Recall	Kea++	ERA	2.78	>>	>>
F-measure	Kea++	ERA	0.77	>>	>>

"<<" indicates SysB has caused significant improvement over SysA
"\>\>" indicates SysB has not caused significant improvement over SysA

5.3.2.3 t-test

We perform t-test on documents of Set-B (175 train docs & 25 test docs). Precision, recall and f-measure values obtained without including Rule VI are used by statistical t-test. Here also we assume that the sample under test is approximately normally distributed. In order for ERA to be effective the before values must be significantly less than the after values. So the mean of the differences must be less than zero. This shows that this is a left tailed t-test.The critical value obtained from the t Distribution table at level of significance 0.05 and at n-1=24 for left tail test is -1.711. The critical value obtained from the t Distribution table at level of significance 0.005 and at n-1=24 for left tail test is -2.797.

The calculations for t-test related to agricultural domain dataset without including Rule VI are presented in Appendix D. Table 5.4 shows the results of t-test performed on the dataset at level of significance 0.05 & 0.005.

We do not apply t-test to check any significant improvement between RA and ERA because RA is not producing any result for agricultural domain datasets.

5.3.3 Test results for Agriculture domain dataset (including rule VI)

The extended refinement methodology has added a new rule in already existing methodology especially for taxonomies having greater hierarchical depth. Two agricultural domain datasets have been used to check any improvement achieved by the addition of this new rule. First dataset comprised of 100 test documents and 15 train documents (Set-A). The other set comprised of 175 train documents and 25 test documents (Set-B). As done before precision, recall and f-measure values will be calculated for each of the dataset. Next average number of assigned keyphrases to test documents will be compared to find the elimination of irrelevant terms from the result set. Finally t-test will be performed on documents of Set-

Table 5.5: Test results for agriculture datasets (including rule VI)

	Kea++	**Refinement Algorithm**	**Extended Refinement Algorithm (including rule VI)**
Set-A (100 train docs & 15 test docs)			
Training-level		11	3
Precision	0.27	0.0	0.33
Recall	0.2	0.0	0.28
F-measure	0.23	0.0	0.3
Set-B (175 train docs & 25 test doc			
Training-level		11	3
Precision	0.23	0.0	0.34
Recall	0.16	0.0	0.23
F-measure	0.19	0.0	0.28

B.

5.3.3.1 Precision, Recall and F-measure

The precision, recall and f-measure values for both the datasets are shown in table 5.5. Training-level value for Set-A is 3 and 11 for ERA and RA respectively. Training-level value for Set-B is also 3 and 11 for ERA and RA respectively.

We can find in figure 5.4 that the precision value has increased when ERA was applied on both the datasets as compared to Kea++. RA like before is not giving any result for the dataset under test. So we can say that the number of extracted keyphrases that are not relevant to the document is least in case of extended refinement algorithm as compared to Kea++.

We can find in figure 5.4 the recall value for both datasets showed improvement when ERA has been applied on the dataset after applying Kea++. This is the point where we can say that the earlier approach without rule VI was not producing the result. This interprets that the keyphrases that are relevant but not extracted is greater in number in case of Kea++ in comparison to ERA. Now we will check the overall improvement by looking at the f-measure value.

F-measure values for both of the datasets in case of ERA are higher when compared with Kea++, as can be seen in the figure 5.4. We can say that inclusion

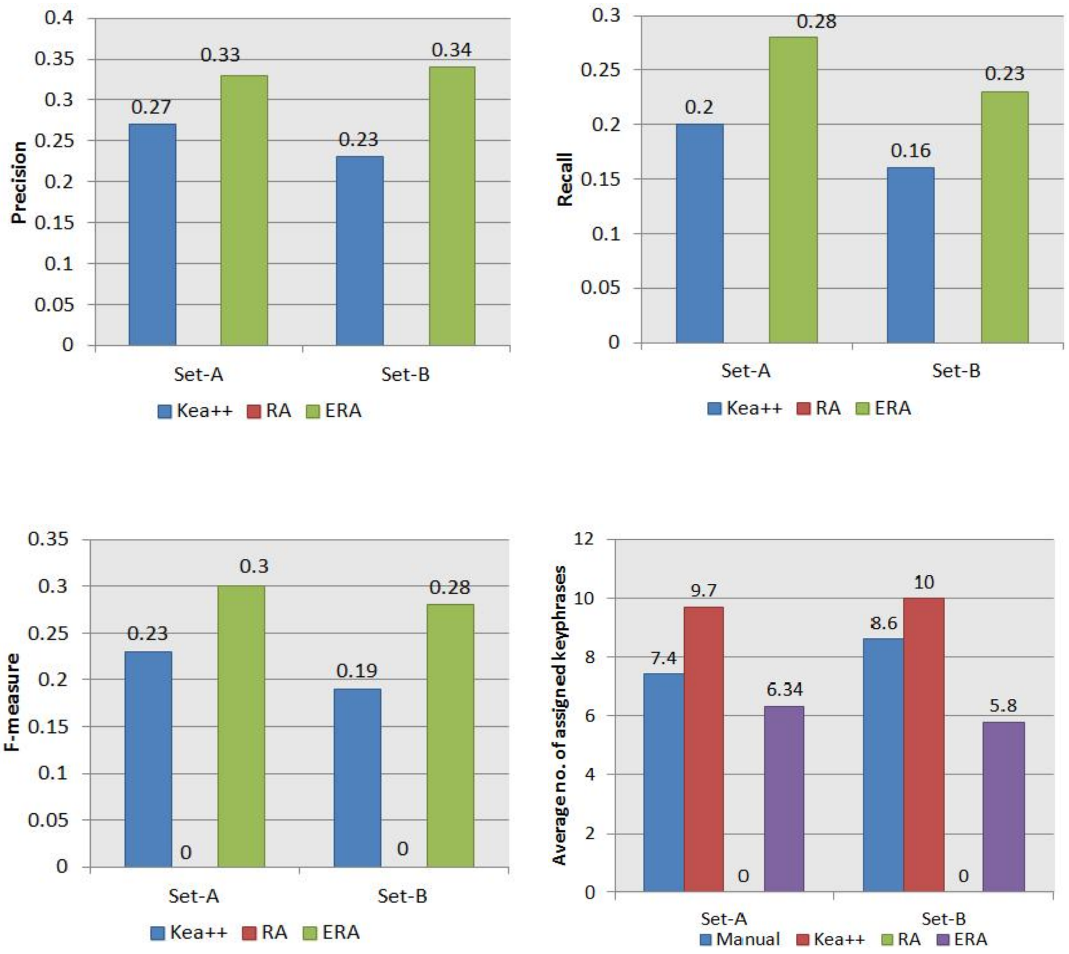


Figure 5.4: Results for agricultural datasets (including rule VI)

of rule VI for taxonomies having hierarchical level greater than five has resulted in the improvement in the results obtained earlier with out including rule VI in the testing.

5.3.3.2 Average number of assigned keyphrases

Average number of keyphrases to test documents is lowest in case of ERA when compared with manually assigned keyphrases, as can be seen from figure 5.4. This has shown the elimination of irrelevant terms from the result set produced by Kea++.

5.3.3.3 t-test

We perform t-test on documents of Set-B (175 train docs & 25 test docs). Precision, recall and f-measure values obtained after including Rule VI are used by

Table 5.6: t-test for agricultural domain (including Rule VI)

Metric	SysA	SysB	t-test	Result(0.05; -1.711)	Result(0.005; -2.797)
Precision	Kea++	ERA	-2.01	<<	>>
Recall	Kea++	ERA	-1.796	<<	>>
F-measure	Kea++	ERA	-1.813	<<	>>

"<<" indicates SysB has caused significant improvement over SysA

">>" indicates SysB has not caused significant improvement over SysA

statistical t-test. Here also we assume that the sample under test is approximately normally distributed. In order for ERA to be effective the before values must be significantly less than the after values. So the mean of the differences must be less than zero. This shows that this is a left tailed t-test.The critical value obtained from the t Distribution table at level of significance 0.05 and at n-1=24 for left tail test is -1.711. The critical value obtained from the t Distribution table at level of significance 0.005 and at n-1=24 for left tail test is -2.797.

The calculations for t-test related to agricultural domain dataset including Rule VI are presented in Appendix D. Table 5.6 shows the results of t-test performed on the dataset at level of significance 0.05 & 0.005.

We do not apply t-test to check any significant improvement between RA and ERA because RA is not producing any result for agricultural domain datasets.

5.3.4 Test results for Mathematics domain dataset

For mathematics domain only one document set was used. The set comprised of 100 train documents and 15 test documents (Set-A). Precision, recall and f-measure values will be calculated. Average number of assigned keyphrases to test documents by Kea++, refinement and its extension will also be calculated. In the end t-test will be performed to check the significance of the result obtained. The training-level value is three for both refinement and extension despite the change in the selection process of training-level as shown in table 5.7. MCS is a taxonomy that resembles ACM CCS in way that it implements general and miscellaneous node concepts and its hierarchical level is less than five as well. The only difference is that total term count and number of top nodes are much higher as compared to that of ACM CCS.

Table 5.7: Test results for mathematics dataset

	Kea++	**Refinement Algorithm**	**Extended Refinement Algorithm**
Set-A (100 train docs & 15 test docs)			
Training-level		3	3
Precision	0.14	0.17	0.19
Recall	0.4	0.39	0.39
F-measure	0.21	0.24	0.26

5.3.4.1 Precision, Recall and F-measure

Table 5.7 shows the value of precision, recall and f-measure.

Precision is higher for refinement and its extension as compared to Kea++ as shown in figure 5.5. This shows that the number of extracted keyphrases that are not relevant to the document is least in case of refinement and extension as compared to Kea++. Recall is slightly lower in case of refinement and its extension when compared with Kea++ as can be seem in figure 5.5. This interprets that the keyphrases that are relevant but not extracted is more in case of refinement and its extension when compared with Kea++.

We can find in figure 5.5 that overall f-measure is showing good results for refinement and extended refinement methodology. Finally t-test will be applied on the f-measure value to check the significance of the result obtained.

5.3.4.2 Average number of assigned keyphrases

Average number of assigned keyphrases is lowest for manually assigned keyphrases. From figure 5.5 we can see that refinement methodology and its extension are reducing the average number of assigned keyphrases to test documents by Kea++ from 8.6 to 6. This shows that our methodology has resulted in reducing irrelevant keyphrases from the result set produced by Kea++.

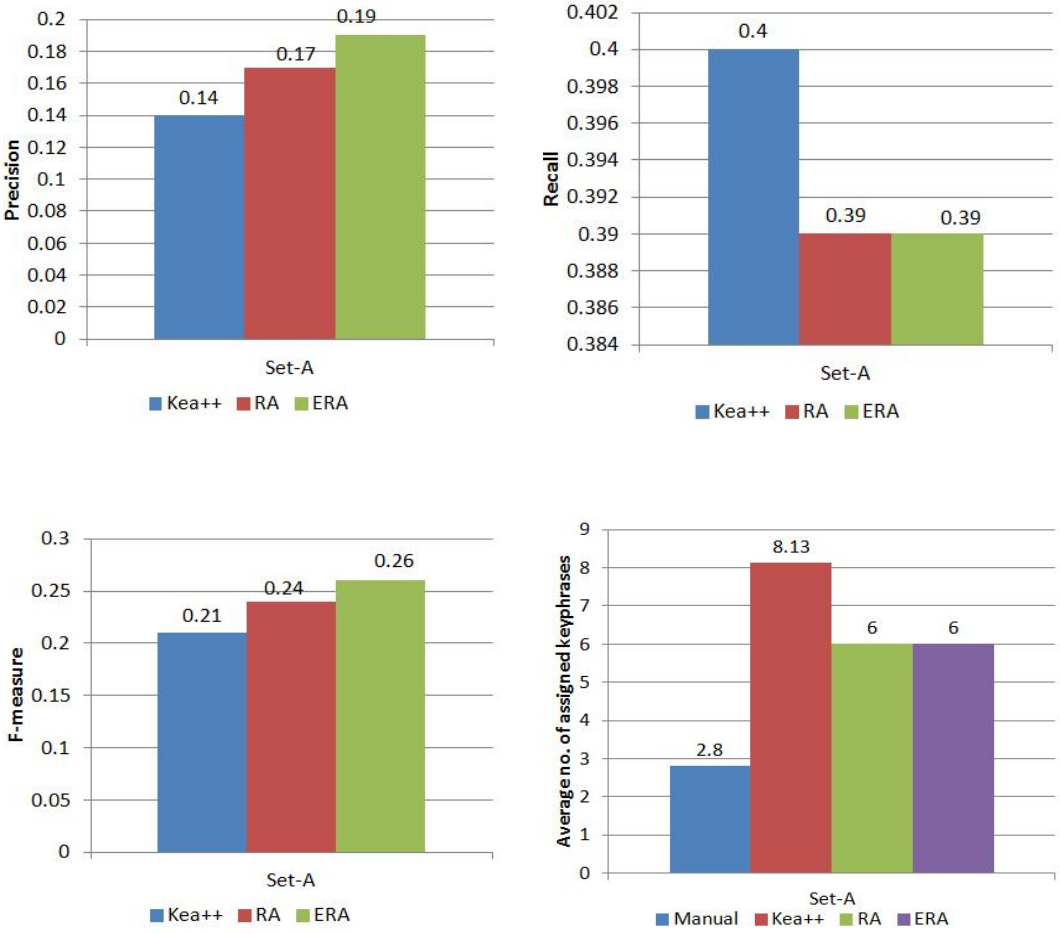


Figure 5.5: Precision for mathematics datasets

Table 5.8: t-test for mathematics domain

Metric	SysA	SysB	t-test	Result(0.05; 1.761)	Result(0.005; 2.977)
Precision	Kea++	ERA	-1.819	<<	>>
Precision	RA	ERA	-1.934	<<	>>
Recall	Kea++	ERA	-1.328	>>	>>
F-measure	Kea++	ERA	-1.82	<<	>>
F-measure	RA	ERA	-2.127	<<	>>

"<<" indicates SysB has caused significant improvement over SysA
"<<" indicates SysB has not caused significant improvement over SysA

5.3.4.3 t-test

We perform t-test on documents of Set-A (100 train docs & 15 test docs). Precision, recall and f-measure value obtained are used by statistical t-test to check that whether the improvement produced in case of the extended refinement algorithm is by chance or is it the significant improvement? Before applying the t-test on the given sample we assume that the sample under test is approximately normally distributed. In order for ERA to be effective the before values must be significantly less than the after values. So the mean of the differences must be less than zero. This shows that this is a left tailed t-test. The critical value obtained from the t Distribution table at level of significance 0.05 and at n-1=14 for left tail test is -1.761. The critical value obtained from the t Distribution table at level of significance 0.005 and at n-1=14 for left tail test is -2.977.

The calculations for t-test related to mathematics domain dataset are presented in Appendix D. Table 5.8 shows the results of t-test performed on the dataset at level of significance 0.05 & 0.005.

We do not apply t-test to check any significant improvement between RA and ERA in case of recall, because the recall values produced by RA and ERA are same for all test documents.

In short we have tested the extended refinement algorithm on three domain datasets; computing, agriculture and mathematics. We have analyzed the working of the algorithm by performing different tests on each domain datasets. These tests include finding the precision, recall and f-measure values, calculating the average number of keyphrases assigned to test documents and performing the statistical t-test. For computing domain the algorithm is giving good results. F-measure value has increased for each of the datasets belonging to computing domain. This was even proved by performing the t-test that ERA has resulted in significant increase in the results obtained earlier. Then agricultural domain dataset has been tested using two different approaches. Initially rule VI of the extended refinement algorithm was excluded and result was calculated. The result obtained was not showing any improvement. Then agricultural dataset was tested by including rule VI of the extended refinement methodology. With its inclusion, results were greatly improved as compared to the previous approach. This was even proved with the help of statistical t-test that rule VI which has been particularly introduced for taxonomies having hierarchical level greater than five is showing better results. Finally mathematics domain dataset was tested. F-measure value obtained was better when compared with Kea++. Significant improvement was even checked by performing the statistical t-test . In the next chapter we will draw some conclusions based on the results that we have obtained in this chapter.

Chapter 6

CONCLUSION

This chapter finally concludes the actual findings of this work. First of all a brief summary of the whole work is presented. Then we look at the contributions of this work and also draw the conclusion. In the end future directions are explored.

6.1 Summary

The refinement methodology developed to overcome the drawbacks of keyphrase assignment tool: Kea++, was no doubt a valuable contribution. This methodology eliminated number of irrelevant keyphrases from the result set produced by Kea++. Initially we started off with the aim of testing the methodology on domains other than computing. In attempt of doing this, different taxonomies were considered. It was realized that the actual structure of the refinement rules need to be altered before getting it work for different domain specific taxonomies. Based on the observations, the extended refinement rules were developed. The main difference between extended refinement rules and founding refinement rules was the selection process of the main parameter: training-level. Earlier training-level was adopted as the level up to which keyphrases in the training dataset are aligned. Now it is adopted as the level upon which maximum number of keyphrases in the training dataset is aligned. Also extended refinement rules are written by keeping in mind the standard taxonomy language: SKOS. Relations that are commonly used to express link between terms in the taxonomy are used, in order to get better understandability of the rules. A new rule has also been added to cater the need of taxonomy having deep hierarchical tree.

After altering the rules, the extended refinement algorithm was developed accordingly. This algorithm was tested on three domains dataset: computing, agriculture and mathematics. For computing domain three separate datasets were used. For agriculture domain five independent datasets were used. One dataset had been there for mathematics domain. Precision, recall and f-measure were calculated. Not only this average number of assigned keyphrases to test documents by Kea++, refinement and extended refinement methodology had also been compared with the manually assigned keyphrases.

We found out that precision and f-measure for all three computing domain datasets were not only higher than Kea++ but also higher than the founding refinement algorithm. Recall was still lower, but good point was the increase of recall from the refinement algorithm. For agriculture domain, testing was done in two phases. Initially testing was done by excluding the new rule that was added for deep hierarchical tree taxonomies. Then testing was performed by including this newly added rule. We saw that phase one was not resulted in any better overall results. But phase two produced better precision, recall and f-measure. For mathematics domain same pattern followed as it was there in computing domain. Increase in precision and f-measure, recall was lower. Average number of assigned keyphrases to the test document was lowest in case of the extended refinement methodology in almost all cases. Showing that the aim of reducing the irrelevant terms from the result set produced by Kea++ was still not ignored.

In the end of testing for each domain, statistical t-test was performed. To find out whether the results obtained were showing considerable improvement or not. We observed that t-test on f-measure value obtained by one of the computing dataset showed that the improvement was significant. Similarly when the t-test was performed on agricultural domain dataset without the inclusion of newly added rule, we found no improvement. T-test on agricultural dataset after including new rule showed that the improvement achieved this time was really significant. Finally mathematics dataset had also undergone statistical t-test and showed that the improvement obtained was significant.

6.2 Contributions

We have drawn the following conclusions, after undergoing all the testing and evaluation of the proposed methodology on the datasets belonging to computing, agriculture and mathematics domain:

(I) Improvement of results:

(a) The extended refinement methodology has improved the overall results obtained earlier by refinement methodology.

The first contribution of this work is of course the improvement of the refinement methodology developed earlier. We have seen that by making change in training-level selection process we have improved the working of the refinement methodology. Most of the time precision value obtained after applying the extended refinement algorithm has increased as compared to refinement algorithm. Not only this recall has been improved in case of the extended refinement algorithm. Overall results shown are also satisfactory.

(b) Significance of the result proved through statistical t-test

The improvement in result values is also proved through statistical t-test. Dataset changes can change the result, significance of result was not confirmed in [13] and that is the reason of performing the t-test. In this work we have strengthen our claim of improvement by performing statistical t-test on different domain datasets. This is not the first time some one has used t-test to prove the significance of the result in information retrieval field. But certainly very few work has been done that has used statistical testing techniques to prove the efficiency of the automatic keyphrase assignment and result refinement techniques.

(II) Methodology is domain independent:

(a) The extended refinement methodology has also extended the founding refinement methodology and meet the requirement of taxonomies having deep hierarchies

A new rule has been added in the extended refinement rules, especially for taxonomies having deep hierarchies. This rule replaces the keyphrases having training-level common parent with the training-level common parent term. Agricultural domain dataset when tested with out including this rule, no improvement was obtained. However when the testing was performed by adding this rule, results have shown considerable improvement.

(b) The extended refinement methodology has defined new conditions in refinement rules so that it can be applied to different domain taxonomy other than computing.

There are some rules that are applicable to one taxonomy and not applicable to other. We have defined new conditions in the rules so that they can be applied when the appropriate conditions are met. This has given greater flexibility to the already existing refinement rules. Also the extended refinement rules are defined using terminologies from standard taxonomy language, SKOS for better understandability and easier applicability.

A part from this, taxonomies are categorized based on three properties: hierarchical level, top nodes and total terms count. The taxonomies are classified as large scale, medium scale and small scale on the basis of these properties. They can be used by some other work dealing with the taxonomies.

The usage of keyphrases has extended beyond the data organization and retrieval task. The extension of refinement methodology is contributing in making the automatic keyphrase assignment process more accurate, refined and domain independent. The task was challenging, as assignment of keyphrases to the document by understanding the true context has not achieved yet the level of accuracy that is needed. Also taxonomy structure and implementation differs with each other from domain to domain. The extended refinement methodology is an upper layer, moreover involvement of deeper semantics to the assignment process can even improve the assignment task further. That can give extended refinement methodology more accurate result set to work on.

6.3 Future Works

There is always need of improvement in any work done. Likewise, this work can be extended and further improved as well. As mentioned the extended refinement methodology is an upper layer over keyphrase assignment tool: Kea++. In future, an attempt can be made to change the background algorithm of keyphrase assignment. Ultimately to develop an independent keyphrase assignment tool based on the learning from different taxonomic structures. So that we can also control the assignment done before refinement. Also there are variety of taxonomies implemented and organized in different manners. Extended refinement rules can further be extended and improved in order to deal with varying taxonomies. Multiple training-level options can also be opted to improve the refinement process further. The idea could be the use of existing training-level selection process, along with average assigned training-level or second maximum assigned training-level.

References

[1] ACM. ACM Computing Classification System: http://www.acm.org/about/class/1998/. Last viewed: 21 March, 2012.

[2] Anonymous. Taxonomy warehouse (everything you need to know about taxonomy): http://www.taxonomywarehouse.com. Last viewed : 31 October, 2011.

[3] Arampatzis, A. T., Tsoris, T., Koster, C. H. A., and van der Weide, T. P. Phrase-based information retrieval. In *Proceedings of Information Processing and Management* (1998), vol. 34(6), pp. 693–707.

[4] Arockiasamy, S. Using the SKOS model for standardizing semantic similarity and relatedness measures for ontological terminologies. Master's thesis, Computer Science and Systems Analysis, Miami University, 2009.

[5] Assem, M., Malaisé, V., Miles, A., and Schreiber, G. A method to convert thesauri to SKOS. In *In volume 4011 of Lecture Notes in Computer Science* (2006), Springer, pp. 95–109.

[6] Assem, M., Menken, M., Schreiber, G., Wielemaker, J., and Wielinga, B. A method for converting thesauri to RDF/OWL. In *Proceedings of the 3rd International Semantic Web Conference* (2004), Springer-Verlag, pp. 17–31.

[7] Bluman, A. *Elementary Statistics A Step by Step Approach.* WCB McGraw-Hill, Third Edition, 1998.

[8] Coletti, M. H., and Bleich, H. L. Medical subject headings used to search the biomedical literature. *Journal of the American Medical Informat-*

ics Association 8(4) (2001), 317–323.

[9] Do, H., Melnik, S., and Rahm, E. Comparison of schema matching evaluations. In *2nd International Workshop on Web Databases (German Informatics Society)* (2002), pp. 221–237.

[10] Dumais, S., Platt, P., Sahami, M., and Heckerman, D. Inductive learning algorithms and representations for text categorization. In *Proceedings of ACM Conference on Information and Knowledge Management* (1998), pp. 148–155.

[11] Euzenat, J. Semantic precision and recall for ontology alignment evaluation. In *International Joint Conferences on Artificial Intelligence* (2007), pp. 348–353.

[12] Fatima, I. Refinement methodology for automatic document alignment using taxonomy in digital libraries. Master's thesis, School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan, 2009.

[13] Fatima, I., Khan, S., and Latif, K. Refinement methodology for automatic document alignment using taxonomy in digital libraries. In *IEEE International Conference on Semantic Computing* (2009), pp. 281–286.

[14] Fayyad, U., and Irani, K. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of the International Joint Conference on Uncertainty in Artificial Intelligence* (1993), pp. 1022–1027.

[15] Feather, J., and Sturges, P. International encyclopedia of information and library science, London & New York, 1996.

[16] Frank, E., Paynter, G. W., Witten, I. H., and Gutwin, C.and Nevill-Manning, C. G. Domain specific keyphrase extraction. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence* (1999), Morgan Kaufmann, pp. 668–673.

[17] Greg Oxton, G., Chmaj, J., and Kay, D. Perspectives on taxonomy, classification, structure and find-ability. Written for Consortium for Service Innovation.

[18] Gutwin, C., Paynter, G., Witten, I. H., Nevill-Manning, C., and Frank, E. Improving browsing in digital libraries with keyphrase indexes. Tech. rep., Department of Computer Science, University of Saskatchewan, Canada, 1998.

[19] Hitzler, P.and Kršotzsch, M., Ehrig, M., and Sure, Y. What is ontology merging?-a category-theoretical perspective using pushouts. In *Workshop on Contexts and ontologies: theory, practice and applications* (2006).

[20] Hull, D. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of the 16th annual international ACM SIGIR Conference on Research and Development in Information Retrieval* (1993), pp. 329–338.

[21] Hulth, A. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the Conference on Empirical Methods in NLP* (2003), pp. 216–223.

[22] Hulth, A. *Combining Machine Learning and Natural Language Processing for Automatic Keyword Extraction.* PhD thesis, Computer and Systems Sciences, Stockholm University, Sweden, 2004.

[23] Jiang, J., and Conrath, D. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference Research on Computational Linguistics (ROCLING X), Taiwan* (1997).

[24] Jones, S., and Mahoui, M. Hierarchical document clustering using automatically extracted keyphrases. In *Proceedings of the 3rd International Asian Conference on Digital Libraries* (2000), pp. 113–120.

[25] Jones, S., and Paynter, G. W. An evaluation of document keyphrase sets. *Journal of Digital Information 4(1)* (2003).

[26] Marko, K., Daumke, P., Schulz, S., and Hahn, U. Cross-language mesh indexing using morphosemantic normalization. In *Proceedings of the American Medical Informatics Association Symposium (AMIA 2003)* (2003), pp. 425–429.

[27] Matthews, B., Miles, A., and Wilson, M. Modelling thesauri for the semantic web. SWAD-Europe Deliverable, July 2003.

[28] Medelyan, O. Automatic keyphrase indexing with a domain-specific thesaurus. Master's thesis, University of Freiburg, Germany, 2005.

[29] Medelyan, O. *Human-competitive automatic topic indexing.* PhD thesis, The University of Waikato, New Zealand, July, 2009.

[30] Medelyan, O., Frank, E., and Witten, I. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing* (2009), pp. 1318–1327.

[31] Medelyan, O., and Witten, I. Thesaurus based automatic keyphrase indexing. In *Proceedings of 6th Agricultural Ontology Service (AOS), workshop at EFITA/WCA* (2005).

[32] Medelyan, O., and Witten, I. H. Thesaurus based automatic keyphrase indexing. In *Proceedings of the Joint Conference on Digital Libraries* (2006), pp. 296–297.

[33] Miller, K., and Matthews, B. Having the right connections the limber project. *Journal of Digital Information 1(8)* (2001).

[34] Motta, E., Buckingham Shum, S., and Domingue, J. Ontology-driven document enrichment: principles, tools and applications. *International Journal of Human Computer Studies 52 (6)* (2000), 1071–1110.

[35] NISO-ANSI. ANSI/NISO Z39.19 - Guidelines for the Construction, Format, and Management of Monolingual Controlled Vocabularies, 2005.

[36] Paice, C. D., and Black, W. J. A three pronged approach to the extraction of key terms and semantic roles. In *Proceedings of the Conference on Recent Advances in Natural Language Processing* (2003).

[37] Pastor, J., Martinez, F., and Rodriguez, J. Advantages of thesaurus representation using the simple knowledge organization system (SKOS) compared with proposed alternatives. *Information Research (International e-Journal) 14(4) paper 422* (December, 2009).

[38] Paukkeri, M., Nieminen, I., Pöllä, M., and Honkela, T. A language-independent approach to keyphrase extraction and evaluation. In

Proceedings of the International Conference on Computational Linguistics (COLING) (2008), pp. 83–86.

[39] Pepper, S., and Moore, G. Xml topic maps (xtm) 1.0 topicmaps.org specification:http://www.topicmaps.org/xtm/1.0/xtm1-20010806.html, 2001. Last viewed : 25 January, 2012.

[40] Salkind, N. *Statistics for People Who (Think They) Hate Statistics*. SAGE Publications, Second Edition, 2004.

[41] Sebastiani, F. Machine learning in automated text categorization. *ACM Computing Survey 34(1)* (March, 2002), 1–47.

[42] Soergel, D., Lauser, B., Liang, A., Fisseha, F., Keizer, J., and Katz, S. Reengineering thesauri for new applications: the agrovoc example. *Journal of Digital Information 4(4)* (March 2004).

[43] Taylor, M. Zthes: a z39.50 profile for thesaurus navigation, March 2001.

[44] Tomokiyo, T., and Hurst, M. A language model approach to keyphrase extraction. In *Proceedings of ACL Workshop on Multiword Expressions* (2003), pp. 33–40.

[45] Turney, P. Learning to extract keyphrases from text. Tech. rep., National Research Council Canada, 1999.

[46] USNLM. Medical Subject Headings: http://www.nlm.nih.gov/mesh/. Last viewed: 21 March, 2012.

[47] Vuorikari, R., Ayre, J., Hartinger, S., Garcia, R., Cecilia, M., Demaurissens, I., Fozo, A., Panzavolta, S., Sillaots, M., and Winer, D. Audit report on melt content - version 2 part ii folksonomies state of the art, requirements and use cases for the melt social tagging tool. Deliverable for MELT Project, March 2007.

[48] Witten, I. H., Paynter, G. W., Frank, E., Gutwin, C., and Nevill-Manning, C. G. Kea: Practical automatic keyphrase extraction. In *Proceedings of the 4th ACM Conference on Digital Libraries* (1999), pp. 254–255.

[49] Zhang, S., and Bodenreider, O. Comparing associative relationships among equivalent concepts across ontologies. *Studies in Health Technology and Informatics 107(1)* (2004), 459–466.

[50] Zuniga, G. Ontology: its transformation from philosophy to information systems. In *The 2nd international conference on Formal Ontology in Information* (2001).

Appendix A

Refinement Algorithm

Refinement algorithm as it was given in [13, 12]

1. Customize the parameters of the KEA++.

2. Train the KEA++ on documents and taxonomy.

3. Generate keyphrase result set with KEA++ for unknown documents.

4. Adopt the Training level from the training data set

5. Identify the labels of keyphrases from taxonomy in KEA++ result set.

6. Initialize refined result set

7. If KEA++ result set contain (Levels of keyphrases are narrower OR broader than Training level) AND contain (Levels of keyphrases are equivalent to Training level) then

    a) If (Levels of keyphrases are equivalent to Training level) then preserve Training level keyphrases in Refined result set

    b) Else If (Levels of keyphrases are narrower OR broader than Training level) then

        i. If (Levels of Keyphrases are broader than Training level) then identify and preserve their Equivalent Training level keyphrases

   ii. Else If (Levels of keyphrases are narrower than Training level and aligned on General node) then stem narrower keyphrases to their respective Training level keyphrases and preserve them in Refined result set

8. Else If KEA++ result set contain (Levels of keyphrases are narrower OR broader than Training level) AND NOT contain (Levels of keyphrases are equivalent to Training level) then

   a) If (Level of keyphrases are narrower than Training level) then preserve narrower keyphrases in Refined result set

   b) Else If (Level of keyphrases are broader than Training level) then identify and preserve their Equivalent Training level keyphrases in Refined result set

9. Remove redundant keyphrases from the Refined result set

10. Return the Refined result set of keyphrases

Appendix B

Parameters Setting for KEA++

KEA++ provides both free and the controlled indexing. The results of controlled indexing are highly affected by the parameter settings while executing the KEA++ . The major parameters that effects the results are:

1. **Vocabulary Name:** It is the name of the taxonomy which is used to restrict the document scope during the keyphrase extraction process. For example ACM vocabulary restricts the document scope to computer science domain while AGROVOC vocabulary restricts the document scope to agriculture domain. Vocabulary name is an optional parameter while executing the KEA++. If a vocabulary is provided, KEA++ matches the document's phrases against vocabulary name and if this parameter is not set then extracted phrases are selected from open domain. So users should select the appropriate taxonomy if they want to extract the semantic keyphrases bounded by a particular information domain.

2. **Vocabulary Format:** It is the format of the taxonomy. KEA++ facilitates only two vocabularies formats either SKOS format or text file format. SKOS format is mostly used because relations and hierarchy of the vocabulary are very clear and easy to process. The software, ThManager[1] , is available to build the SKOS format of any ontology. Text files are difficult to manage while changing relationships among terms and introducing the new concepts [18]. The loading of text files during the model creation of the KEA++ creates more errors as compare to SKOS format.

[1] http://sourceforge.net/projects/ThManager

3. **Vocabulary Encoding:** This parameter is about the encoding format of the taxonomy. KEA++ supports the UTF8.

4. **Document Language:** KEA support different languages such as English, French and Spanish. A user has to set this parameter according to the language of a document and taxonomy.

5. **Max. Length of Phrases:** This parameter sets the maximum length of the keyphrase returned as a semantic term. Phrases are actually consists of one or more words (i.e. Geometric Algorithms, Languages, and Systems and Special Functions Approximations) this parameter sets the value of the returned phrases according to the required number of words in semantic phrase. The returned phrase length cannot exceed the limit set by this parameter. The parameter value should be set according to the maximum available length of keyphrases in the taxonomy. If the value of the parameter is not set after analyzing the phrases length in the taxonomy then the results are badly affected.

6. **Min. length of Phrase:** This parameter sets the minimum length of the keyphrases returned as a semantic term. This parameter is the most important parameter during the extraction of the semantic keyphrases. For example if the max. length of the phrase is 5 in taxonomy and min. length is 2. This parameter should be set between 2 and 5, the minimum length value is set by considering the most occurring phrases' length in the taxonomy. If the minimum length of this parameter is set by considering the minimum length of the keyphrases in the taxonomy then results might be affected.

7. **Min. Occurrence:** Minimum occurrence of a phrase in a document means that the document must have the minimum number of times the extracted keyphrases as a set by this parameter. This parameter should be set after considering the required percentage of relevant information in the document. Higher the value of this parameter might affect the results badly. Documents might have the relevant information but the percentage might not present as mentioned in this parameter. This parameter should be set on some average value like for long documents the recommended value is 2 for this parameter.

8. **No of Extracted Keyphrases:** This parameter limits the maximum number of returned keyphrases. This parameter limits the maximum number of the returned keyphrases. Mostly other parameters have more effect on

results as compared to this parameter. A number of returned keyphrases might not according to this parameter if they are more affected by other parameters. If the number of extracted keyphrases, after applying the other parameters, are high as set by this parameter then KEA++ selects the top most terms as mention in this parameter value.

Appendix C

The t Distribution table

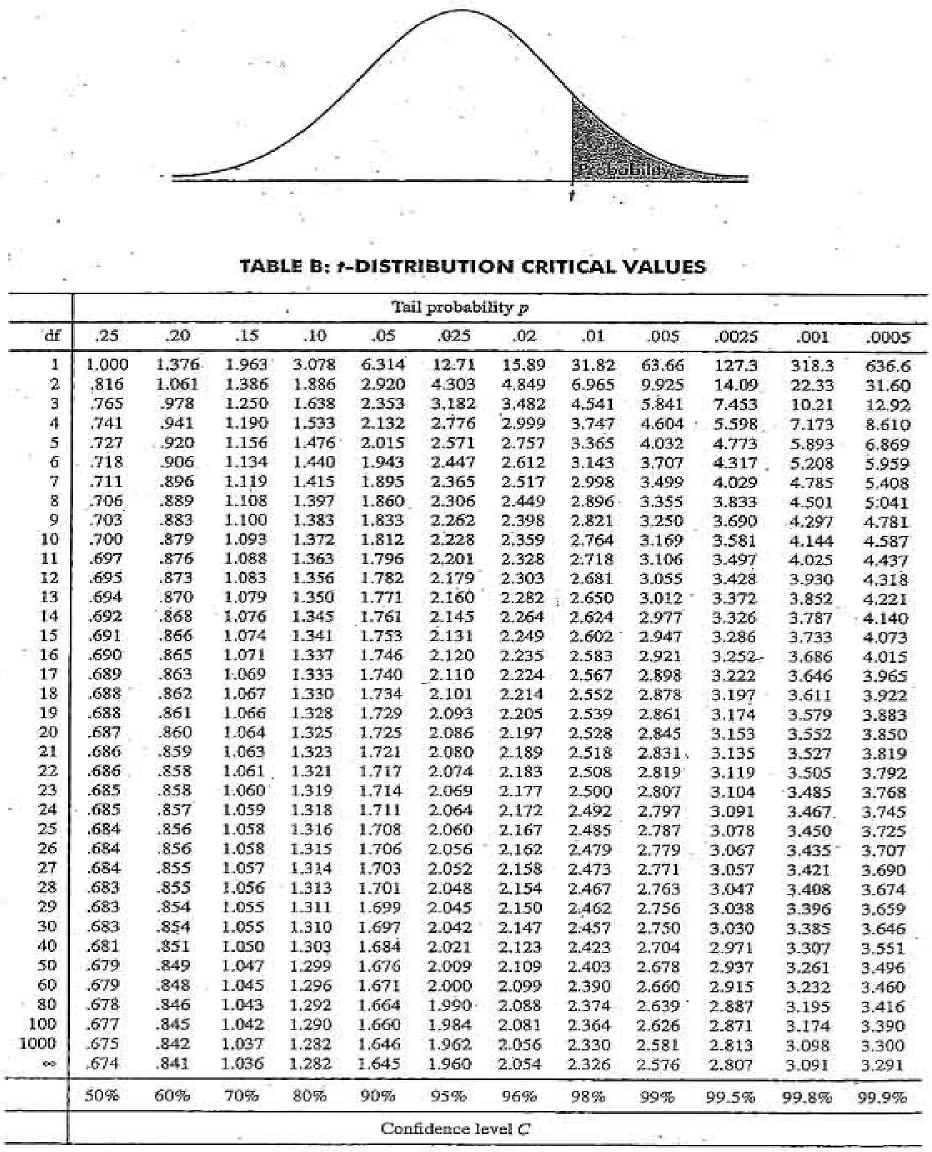

Probability p
t

TABLE B: t-DISTRIBUTION CRITICAL VALUES

	Tail probability p											
df	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	.688	.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	.687	.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	.686	.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	.686	.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792
23	.685	.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104	3.485	3.768
24	.685	.857	1.059	1.318	1.711	2.064	2.172	2.492	2.797	3.091	3.467	3.745
25	.684	.856	1.058	1.316	1.708	2.060	2.167	2.485	2.787	3.078	3.450	3.725
26	.684	.856	1.058	1.315	1.706	2.056	2.162	2.479	2.779	3.067	3.435	3.707
27	.684	.855	1.057	1.314	1.703	2.052	2.158	2.473	2.771	3.057	3.421	3.690
28	.683	.855	1.056	1.313	1.701	2.048	2.154	2.467	2.763	3.047	3.408	3.674
29	.683	.854	1.055	1.311	1.699	2.045	2.150	2.462	2.756	3.038	3.396	3.659
30	.683	.854	1.055	1.310	1.697	2.042	2.147	2.457	2.750	3.030	3.385	3.646
40	.681	.851	1.050	1.303	1.684	2.021	2.123	2.423	2.704	2.971	3.307	3.551
50	.679	.849	1.047	1.299	1.676	2.009	2.109	2.403	2.678	2.937	3.261	3.496
60	.679	.848	1.045	1.296	1.671	2.000	2.099	2.390	2.660	2.915	3.232	3.460
80	.678	.846	1.043	1.292	1.664	1.990	2.088	2.374	2.639	2.887	3.195	3.416
100	.677	.845	1.042	1.290	1.660	1.984	2.081	2.364	2.626	2.871	3.174	3.390
1000	.675	.842	1.037	1.282	1.646	1.962	2.056	2.330	2.581	2.813	3.098	3.300
∞	.674	.841	1.036	1.282	1.645	1.960	2.054	2.326	2.576	2.807	3.091	3.291
	50%	60%	70%	80%	90%	95%	96%	98%	99%	99.5%	99.8%	99.9%
	Confidence level C											

Figure C.1: t Distribution table

Appendix D

Statistical t-test Calculations

D.1 Statistical t-test on computing domain dataset

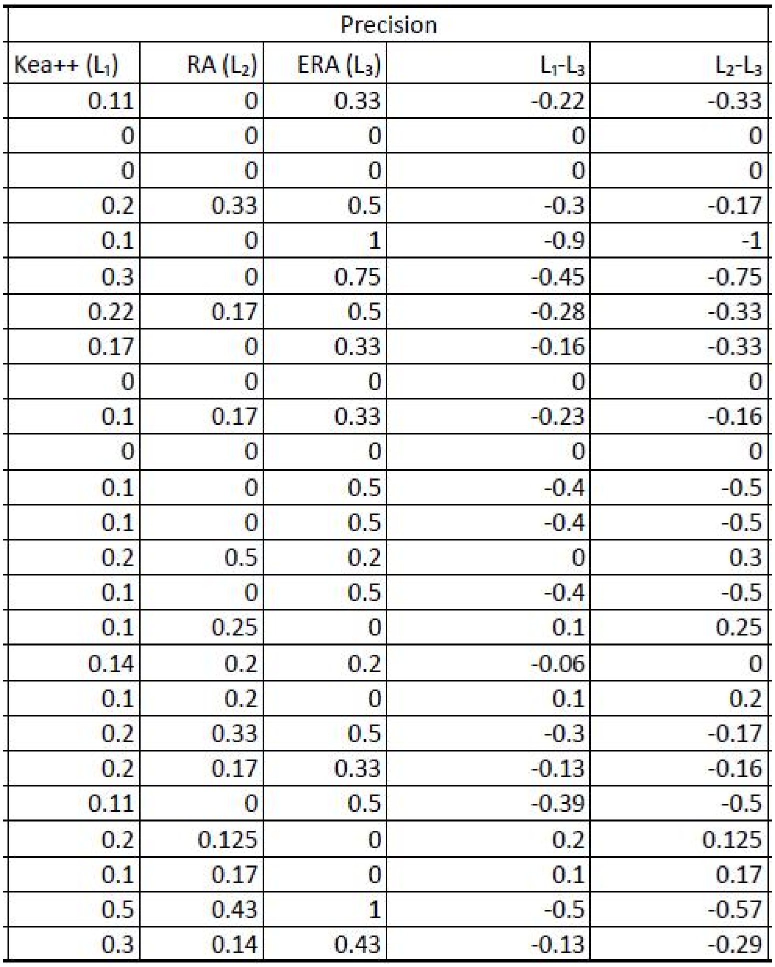
(I) Precision

Figure D.1 shows the precision value for each test document after applying Kea++, refinement and extended refinement algorithm.

(a) t-test to check significant improvement between Kea++ and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from Kea++.
For the sample under test, following are the values of the variables used in t-test:
D_M = -0.19
µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 25
$\sum$D = -4.75
$\sum D^2$=2.3873

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{2.3873 - (-4.75)^2/25\}/24} = 0.2487$$



Precision				
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3	L_2-L_3
0.11	0	0.33	-0.22	-0.33
0	0	0	0	0
0	0	0	0	0
0.2	0.33	0.5	-0.3	-0.17
0.1	0	1	-0.9	-1
0.3	0	0.75	-0.45	-0.75
0.22	0.17	0.5	-0.28	-0.33
0.17	0	0.33	-0.16	-0.33
0	0	0	0	0
0.1	0.17	0.33	-0.23	-0.16
0	0	0	0	0
0.1	0	0.5	-0.4	-0.5
0.1	0	0.5	-0.4	-0.5
0.2	0.5	0.2	0	0.3
0.1	0	0.5	-0.4	-0.5
0.1	0.25	0	0.1	0.25
0.14	0.2	0.2	-0.06	0
0.1	0.2	0	0.1	0.2
0.2	0.33	0.5	-0.3	-0.17
0.2	0.17	0.33	-0.13	-0.16
0.11	0	0.5	-0.39	-0.5
0.2	0.125	0	0.2	0.125
0.1	0.17	0	0.1	0.17
0.5	0.43	1	-0.5	-0.57
0.3	0.14	0.43	-0.13	-0.29

Figure D.1: Precision values (computing dataset)

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.19 - 0)/(0.2487/\sqrt{25}) = -3.82$$

At level of significance 0.05 the critical value is -1.711. Since, -3.82 <-1.711, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.2(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.797. Since, -3.82 <-2.797, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.2(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from Kea++.

b) t-test to check significant improvement between RA and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from RA.
For the sample under test, following are the values of the variables used in t-test:
D_M = -0.2086
µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 25
$\sum$D = -5.215
$\sum D^2$=3.644

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{3.644 - (-5.215)/25\}/24} = 0.3264$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.2086 - 0)/(0.3264/\sqrt{25}) = -3.20$$

At level of significance 0.05 the critical value is -1.711. Since, -3.20 <-1.711, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.2(c) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from RA.

At level of significance 0.005 the critical value is -2.797. Since, -3.20 <-2.797, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.2(d) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from RA.

(II) Recall

Figure D.3 shows the recall value for each test document after applying Kea++, refinement and extended refinement algorithm.

(a) t-test to check significant improvement between Kea++ and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from Kea++.
For the sample under test, following are the values of the variables used in t-test:
$D_M = 0.0709$
$\mu = 0$ (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 25
$\sum D = 1.773$
$\sum D^2$=0.5469

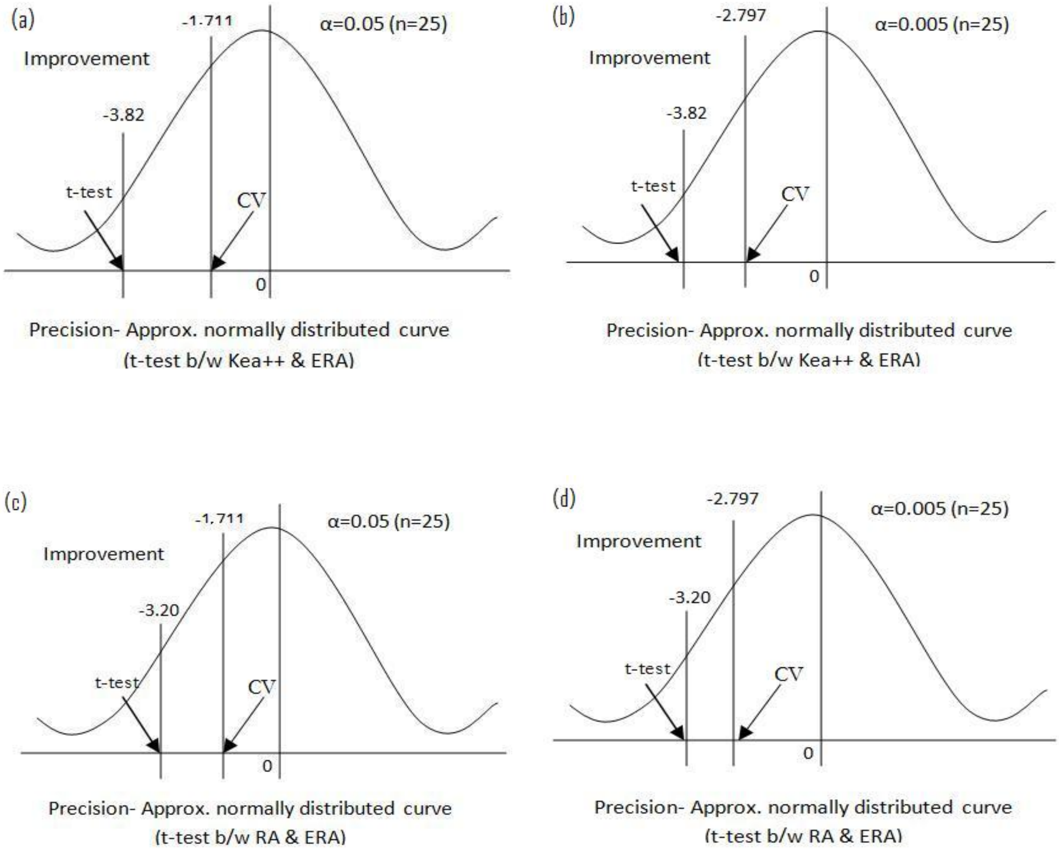
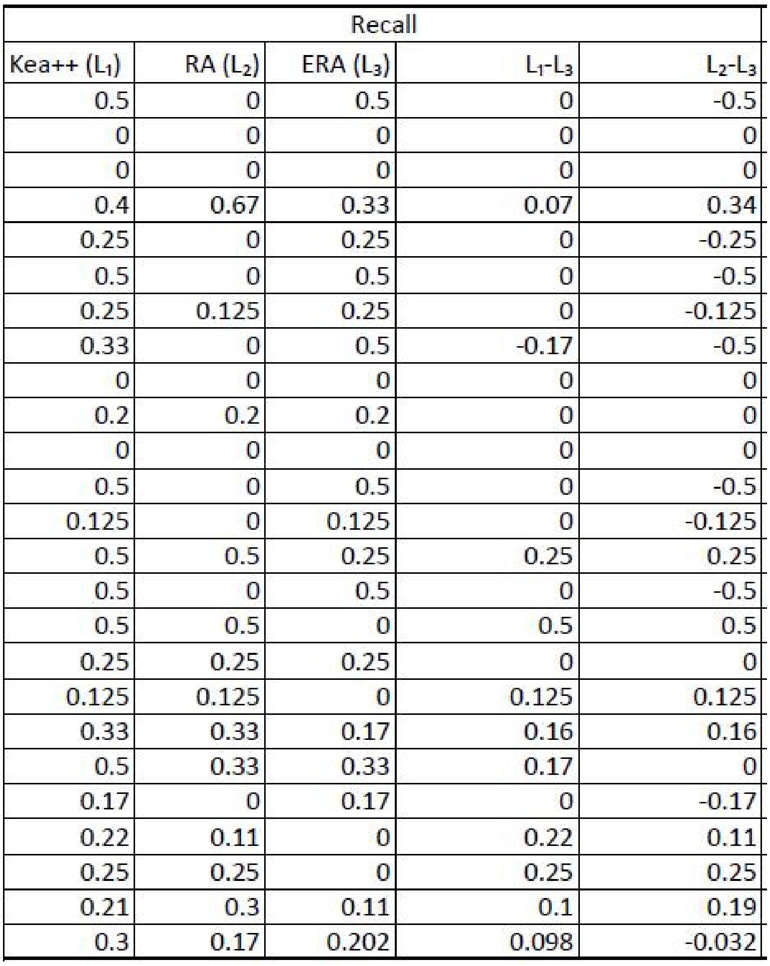


Figure D.2: Precision t-test (computing dataset)



Recall				
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3	L_2-L_3
0.5	0	0.5	0	-0.5
0	0	0	0	0
0	0	0	0	0
0.4	0.67	0.33	0.07	0.34
0.25	0	0.25	0	-0.25
0.5	0	0.5	0	-0.5
0.25	0.125	0.25	0	-0.125
0.33	0	0.5	-0.17	-0.5
0	0	0	0	0
0.2	0.2	0.2	0	0
0	0	0	0	0
0.5	0	0.5	0	-0.5
0.125	0	0.125	0	-0.125
0.5	0.5	0.25	0.25	0.25
0.5	0	0.5	0	-0.5
0.5	0.5	0	0.5	0.5
0.25	0.25	0.25	0	0
0.125	0.125	0	0.125	0.125
0.33	0.33	0.17	0.16	0.16
0.5	0.33	0.33	0.17	0
0.17	0	0.17	0	-0.17
0.22	0.11	0	0.22	0.11
0.25	0.25	0	0.25	0.25
0.21	0.3	0.11	0.1	0.19
0.3	0.17	0.202	0.098	-0.032

Figure D.3: Recall values (computing dataset)

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{0.5469 - (1.773)^2/25\}/24} = 0.1325$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (0.0709 - 0)/(0.1325/\sqrt{25}) = 2.68$$

At level of significance 0.05 the critical value is -1.711. Since, 2.68 >-1.711, so we can say that null hypothesis can not be rejected and our claim comes out to be false. Figure D.4(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.797. Since, 2.68 >-2.797, so we can say that null hypothesis can not be rejected and our claim comes out to be false. Figure D.4(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

b) t-test to check significant improvement between RA and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from RA.
For the sample under test, following are the values of the variables used in t-test:
D_M = -0.0511
µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 25
$\sum$D = -1.277
$\sum D^2$=1.9537

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{1.9537 - (-1.277)/25\}/24} = 0.2805$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.0511 - 0)/(0.2805/\sqrt{25}) = -0.91$$

At level of significance 0.05 the critical value is -1.711. Since, -0.91 > -1.711, so we can say that null hypothesis can not be rejected and our claim comes out to be false. Figure D.4(c) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from RA.

At level of significance 0.005 the critical value is -2.797. Since, -0.91 >-2.797, so we can say that null hypothesis can not be rejected and our claim comes out to be false. Figure D.4(d) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from RA.

(III) F-measure

Figure D.5 shows the f-measure value for each test document after applying Kea++, refinement and extended refinement algorithm.

(a) t-test to check significant improvement between Kea++ and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from Kea++.
For the sample under test, following are the values of the variables used in t-test:
D_M = -0.0548

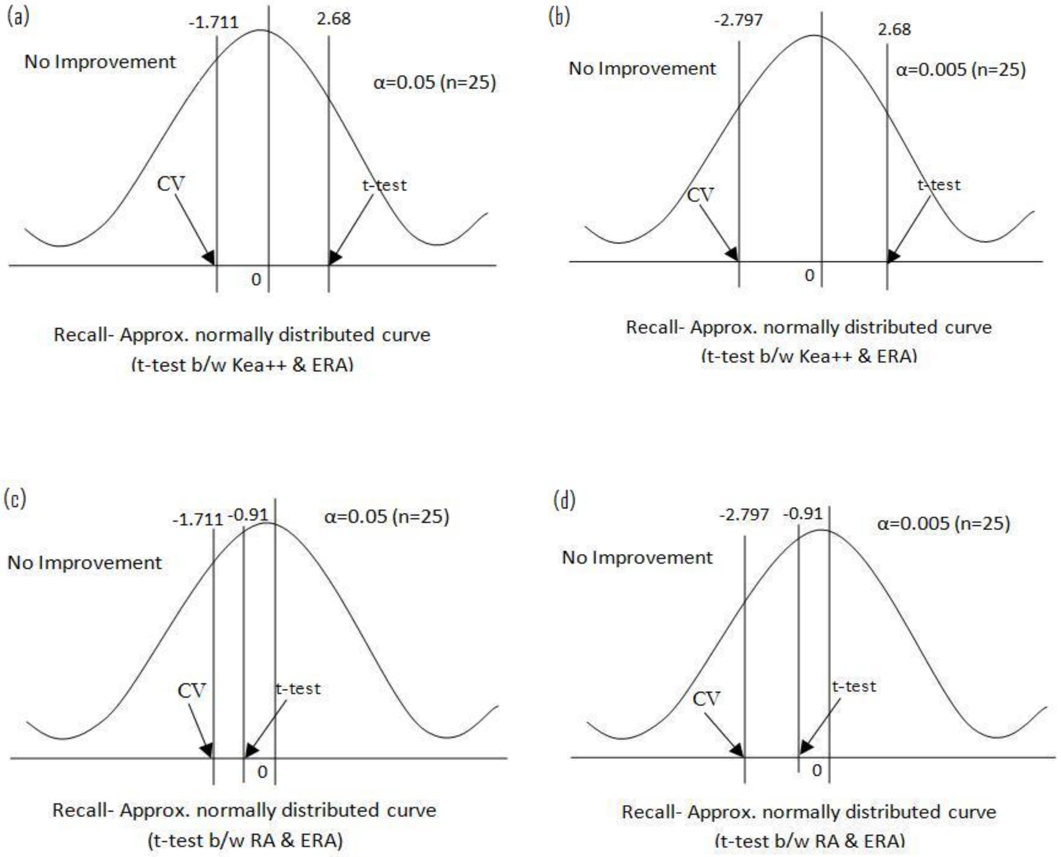
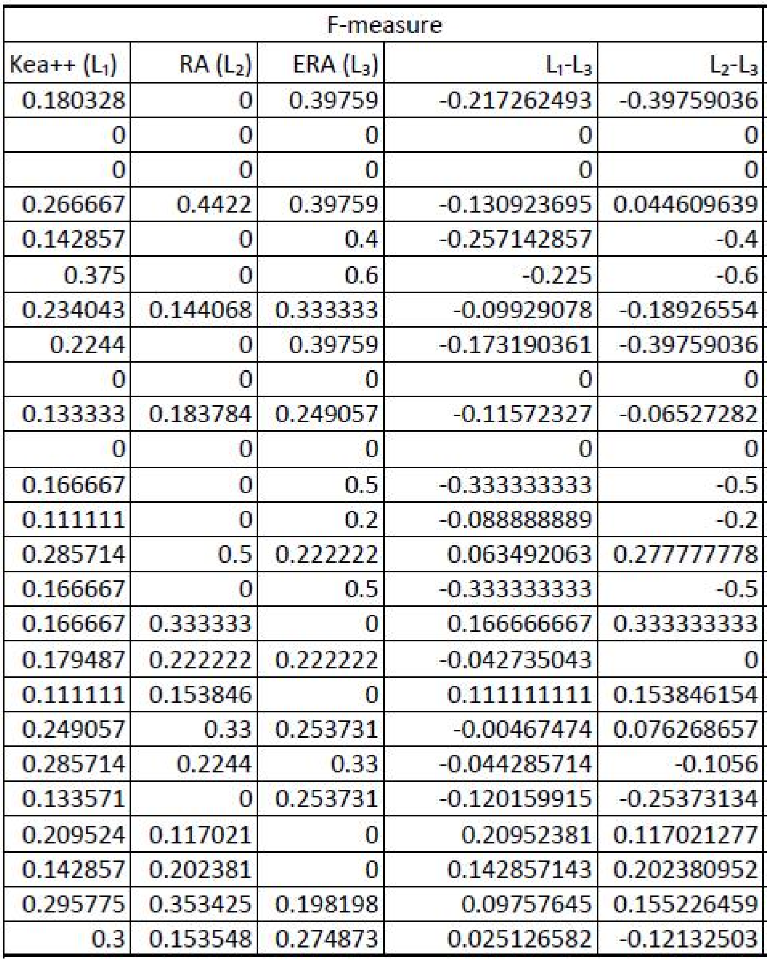


Figure D.4: Recall t-test (computing dataset)



F-measure				
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3	L_2-L_3
0.180328	0	0.39759	-0.217262493	-0.39759036
0	0	0	0	0
0	0	0	0	0
0.266667	0.4422	0.39759	-0.130923695	0.044609639
0.142857	0	0.4	-0.257142857	-0.4
0.375	0	0.6	-0.225	-0.6
0.234043	0.144068	0.333333	-0.09929078	-0.18926554
0.2244	0	0.39759	-0.173190361	-0.39759036
0	0	0	0	0
0.133333	0.183784	0.249057	-0.11572327	-0.06527282
0	0	0	0	0
0.166667	0	0.5	-0.333333333	-0.5
0.111111	0	0.2	-0.088888889	-0.2
0.285714	0.5	0.222222	0.063492063	0.277777778
0.166667	0	0.5	-0.333333333	-0.5
0.166667	0.333333	0	0.166666667	0.333333333
0.179487	0.222222	0.222222	-0.042735043	0
0.111111	0.153846	0	0.111111111	0.153846154
0.249057	0.33	0.253731	-0.00467474	0.076268657
0.285714	0.2244	0.33	-0.044285714	-0.1056
0.133571	0	0.253731	-0.120159915	-0.25373134
0.209524	0.117021	0	0.20952381	0.117021277
0.142857	0.202381	0	0.142857143	0.202380952
0.295775	0.353425	0.198198	0.09757645	0.155226459
0.3	0.153548	0.274873	0.025126582	-0.12132503

Figure D.5: F-measure values (computing dataset)

µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 25
$\sum$D = -1.3696
$\sum D^2$=0.6013

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{0.6013 - (-1.3696)^2/25\}/24} = 0.1481$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.0548 - 0)/(0.1481/\sqrt{25}) = -1.85$$

At level of significance 0.05 the critical value is -1.711. Since, -1.85 <-1.711, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.6(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.797. Since, -1.85 >-2.797, so we can say that null hypothesis can not be rejected and our claim comes out to be false. Figure D.6(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

b) t-test to check significant improvement between RA and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from RA.
For the sample under test, following are the values of the variables used in t-test:

D_M = -0.0948
µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 25
$\sum$D = -2.3699
$\sum D^2$=1.805

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{1.805 - (-2.3699)/25\}/24} = 0.2566$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.0948 - 0)/(0.2566/\sqrt{25}) = -1.847$$

At level of significance 0.05 the critical value is -1.711. Since, -1.847 <-1.711, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.6(c) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from RA.

At level of significance 0.005 the critical value is -2.797. Since, -1.847 > -2.797, so we can say that null hypothesis can not be rejected and our claim comes out to be false. Figure D.6(d) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from RA.

D.2 Statistical t-test on agricultural domain dataset (excluding Rule VI)

(I) Precision

Figure D.7 shows the precision value for each test document after applying Kea++, refinement and extended refinement algorithm.

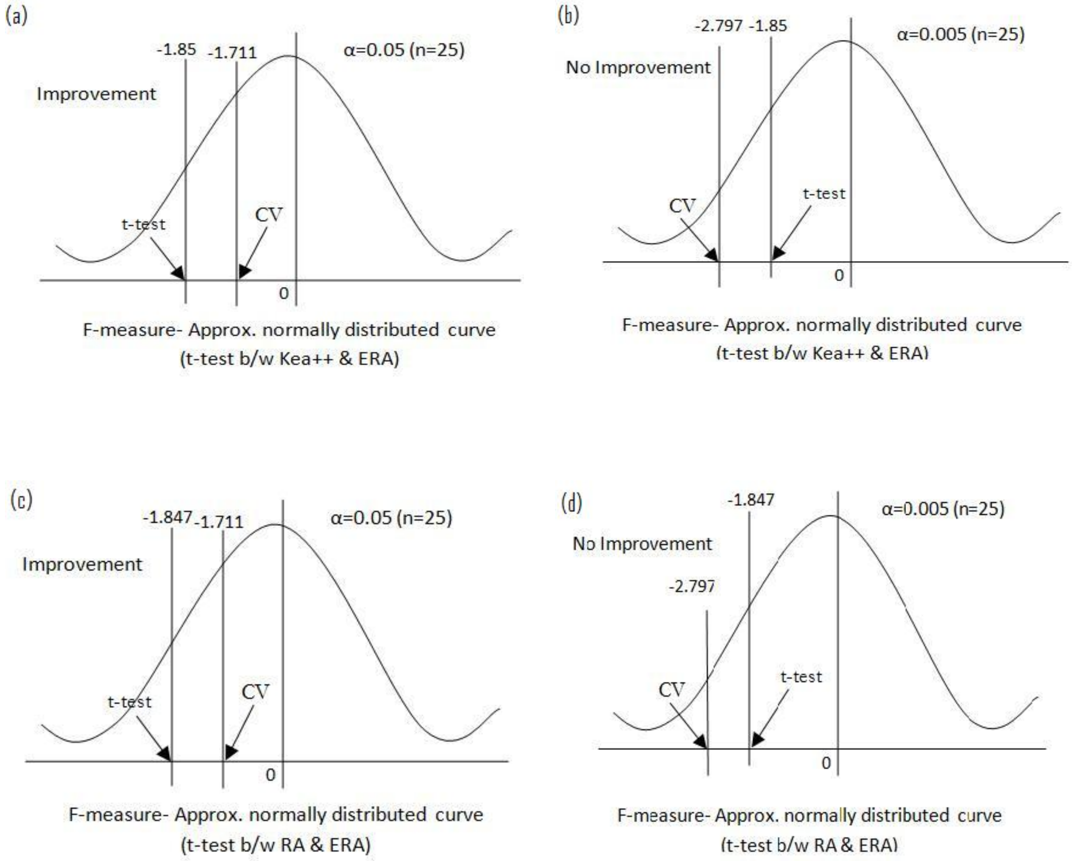
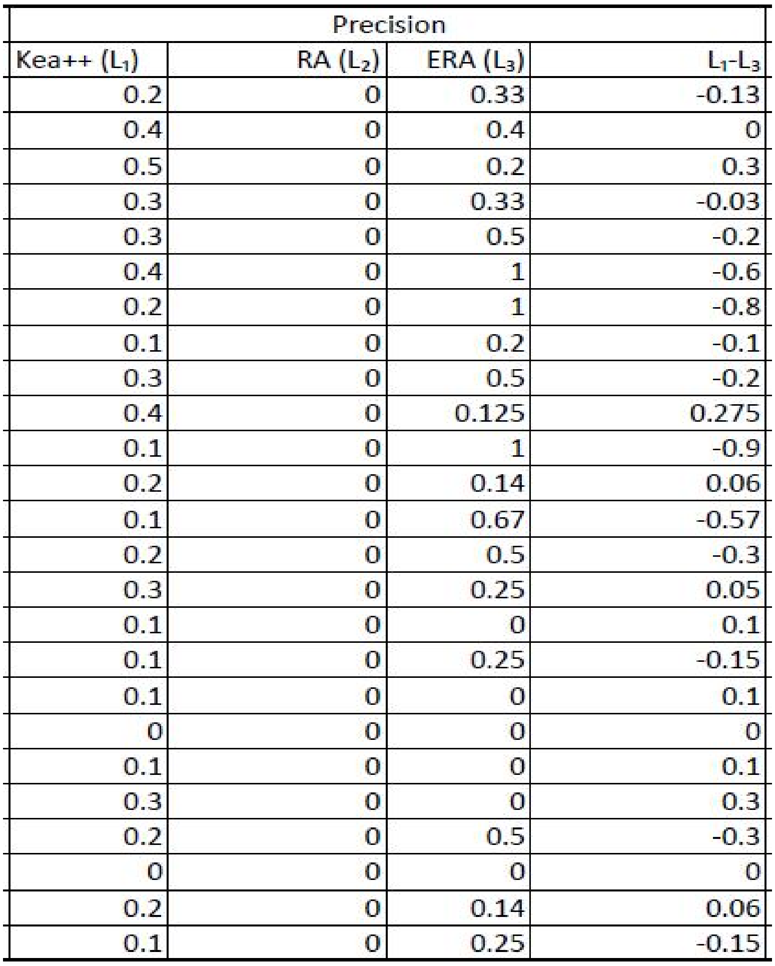


Figure D.6: F-measure t-test (computing dataset)



Precision			
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3
0.2	0	0.33	-0.13
0.4	0	0.4	0
0.5	0	0.2	0.3
0.3	0	0.33	-0.03
0.3	0	0.5	-0.2
0.4	0	1	-0.6
0.2	0	1	-0.8
0.1	0	0.2	-0.1
0.3	0	0.5	-0.2
0.4	0	0.125	0.275
0.1	0	1	-0.9
0.2	0	0.14	0.06
0.1	0	0.67	-0.57
0.2	0	0.5	-0.3
0.3	0	0.25	0.05
0.1	0	0	0.1
0.1	0	0.25	-0.15
0.1	0	0	0.1
0	0	0	0
0.1	0	0	0.1
0.3	0	0	0.3
0.2	0	0.5	-0.3
0	0	0	0
0.2	0	0.14	0.06
0.1	0	0.25	-0.15

Figure D.7: Precision values (agricultural dataset excluding rule VI)

(a) t-test to check significant improvement between Kea++ and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from Kea++.
For the sample under test, following are the values of the variables used in t-test:
D_M = -0.1234
µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 25
$\sum D$ = -3.085
$\sum D^2$= 2.7630

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{2.7630 - (-3.085)^2/25\}/24} = 0.3151$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.1234 - 0)/(0.3151/\sqrt{25}) = -1.96$$

At level of significance 0.05 the critical value is -1.711. Since, -1.96 < -1.711, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.8(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.797. Since, -1.96 > -2.797, so we can say that null hypothesis cannot be rejected and our claim comes out to be false. Figure D.8(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

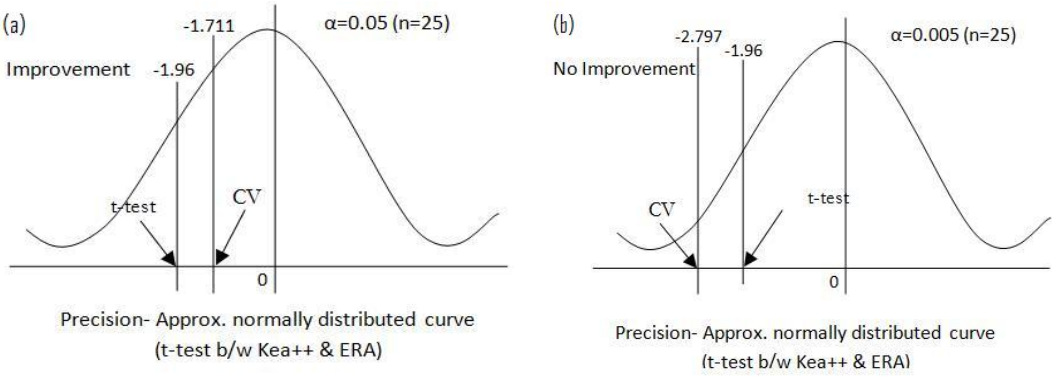


Figure D.8: Precision t-test (agricultural dataset excluding rule VI)

(II) Recall

Figure D.9 shows the recall value for each test document after applying Kea++, refinement and extended refinement algorithm.

(a) t-test to check significant improvement between Kea++ and ERA: Following are the assumption and claim made for the t-test:

Null Hypothesis: There is no improvement achieved as a result of application of ERA.

Claim: ERA has caused improvement in the results obtained previously from Kea++.

For the sample under test, following are the values of the variables used in t-test:

D_M =0.0568

µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)

n= 25

$\sum D$ = 1.42

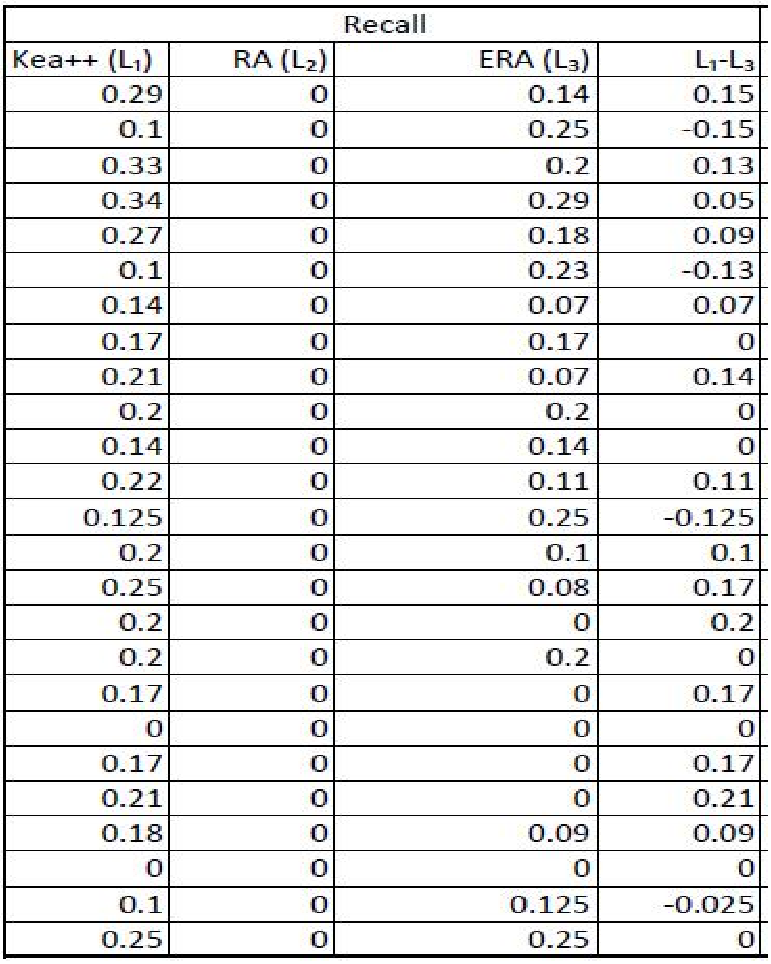
$\sum D^2$= 0.3312

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{0.3312 - (1.42)^2/25\}/24} = 0.1022$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$



Recall			
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3
0.29	0	0.14	0.15
0.1	0	0.25	-0.15
0.33	0	0.2	0.13
0.34	0	0.29	0.05
0.27	0	0.18	0.09
0.1	0	0.23	-0.13
0.14	0	0.07	0.07
0.17	0	0.17	0
0.21	0	0.07	0.14
0.2	0	0.2	0
0.14	0	0.14	0
0.22	0	0.11	0.11
0.125	0	0.25	-0.125
0.2	0	0.1	0.1
0.25	0	0.08	0.17
0.2	0	0	0.2
0.2	0	0.2	0
0.17	0	0	0.17
0	0	0	0
0.17	0	0	0.17
0.21	0	0	0.21
0.18	0	0.09	0.09
0	0	0	0
0.1	0	0.125	-0.025
0.25	0	0.25	0

Figure D.9: Recall values (agricultural dataset excluding rule VI)

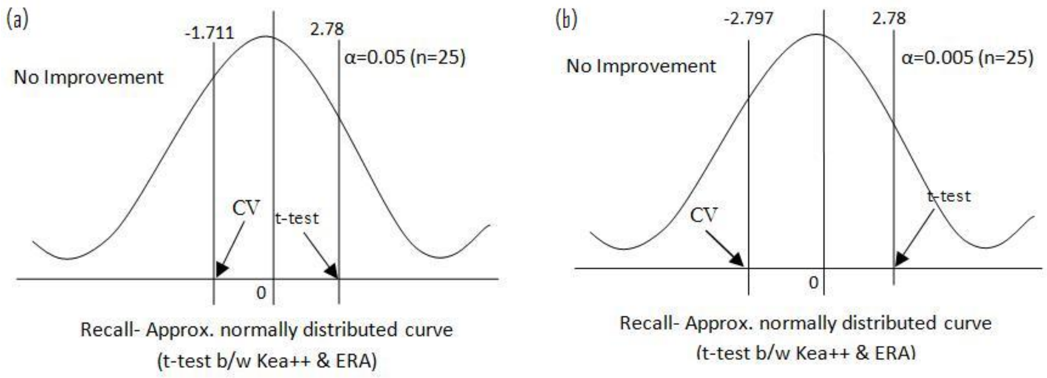


Figure D.10: Recall t-test (agricultural dataset excluding rule VI)

$$t = (0.0568 - 0)/(0.1022/\sqrt{25}) = 2.78$$

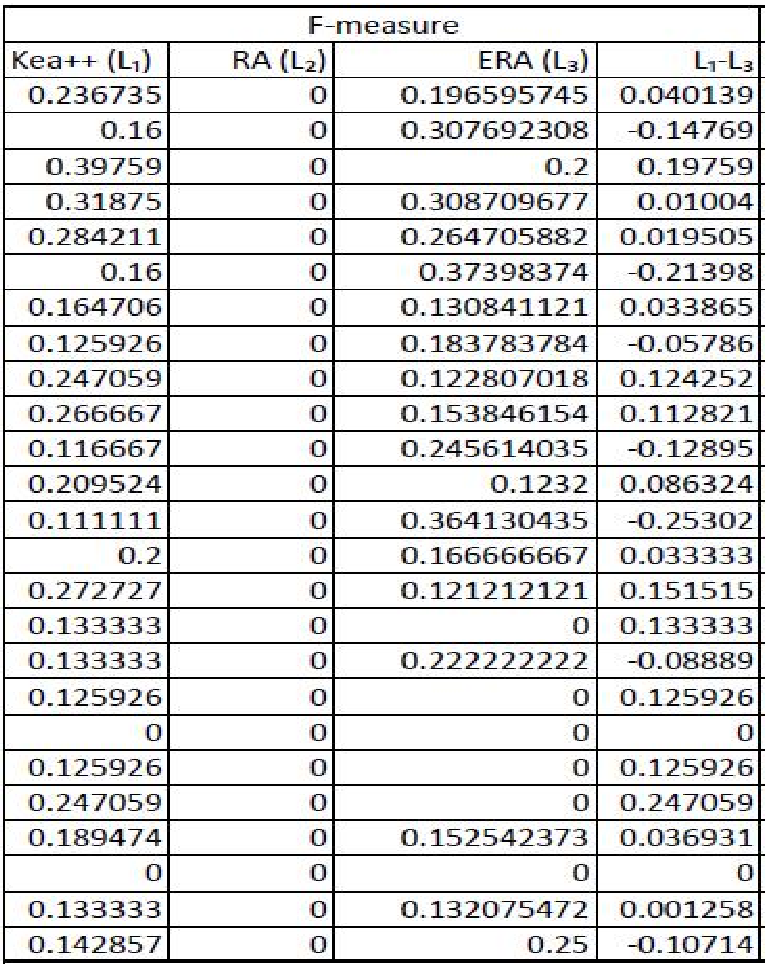
At level of significance 0.05 the critical value is -1.711. Since, 2.78 > -1.711, so we can say that null hypothesis cannot be rejected and our claim comes out to be false. Figure D.10(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.797. Since, 2.78 > -2.797, so we can say that null hypothesis cannot be rejected and our claim comes out to be false. Figure D.10(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

(III) F-measure

Figure D.11 shows the f-measure value for each test document after applying Kea++, refinement and extended refinement algorithm.

(a) t-test to check significant improvement between Kea++ and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from Kea++.
For the sample under test, following are the values of the variables used in t-test:
D_M =0.0193



F-measure			
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3
0.236735	0	0.196595745	0.040139
0.16	0	0.307692308	-0.14769
0.39759	0	0.2	0.19759
0.31875	0	0.308709677	0.01004
0.284211	0	0.264705882	0.019505
0.16	0	0.37398374	-0.21398
0.164706	0	0.130841121	0.033865
0.125926	0	0.183783784	-0.05786
0.247059	0	0.122807018	0.124252
0.266667	0	0.153846154	0.112821
0.116667	0	0.245614035	-0.12895
0.209524	0	0.1232	0.086324
0.111111	0	0.364130435	-0.25302
0.2	0	0.166666667	0.033333
0.272727	0	0.121212121	0.151515
0.133333	0	0	0.133333
0.133333	0	0.222222222	-0.08889
0.125926	0	0	0.125926
0	0	0	0
0.125926	0	0	0.125926
0.247059	0	0	0.247059
0.189474	0	0.152542373	0.036931
0	0	0	0
0.133333	0	0.132075472	0.001258
0.142857	0	0.25	-0.10714

Figure D.11: F-measure values (agricultural dataset excluding rule VI)

µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 25
$\sum$D = 0.4823
$\sum$D^2=0.3848

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{0.3848 - (0.4823)^2/25\}/24} = 0.1251$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (0.0193 - 0)/(/\sqrt{25}) = 0.77$$

At level of significance 0.05 the critical value is -1.711. Since, 0.77 > -1.711, so we can say that null hypothesis cannot be rejected and our claim comes out to be false. Figure D.12(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.797. Since, 0.77 > -2.797, so we can say that null hypothesis cannot be rejected and our claim comes out to be false. Figure D.12(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

D.3 Statistical t-test on agricultural domain dataset (including Rule VI)

(I) Precision

Figure D.13 shows the precision value for each test document after applying Kea++, refinement and extended refinement algorithm.

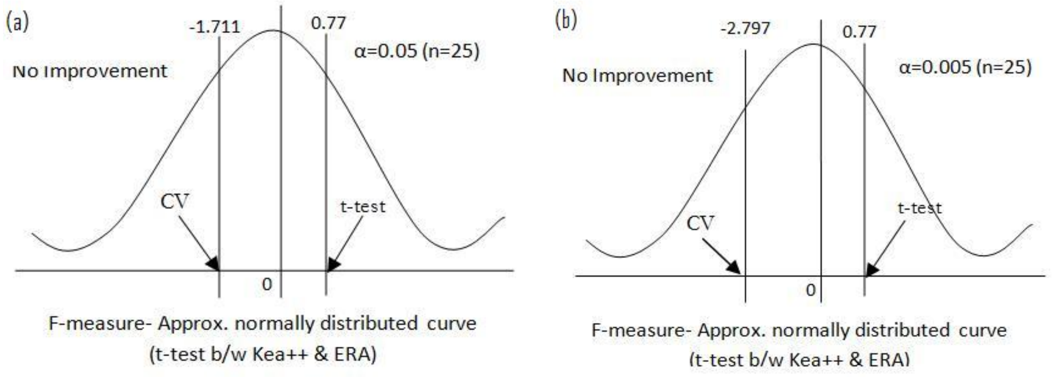
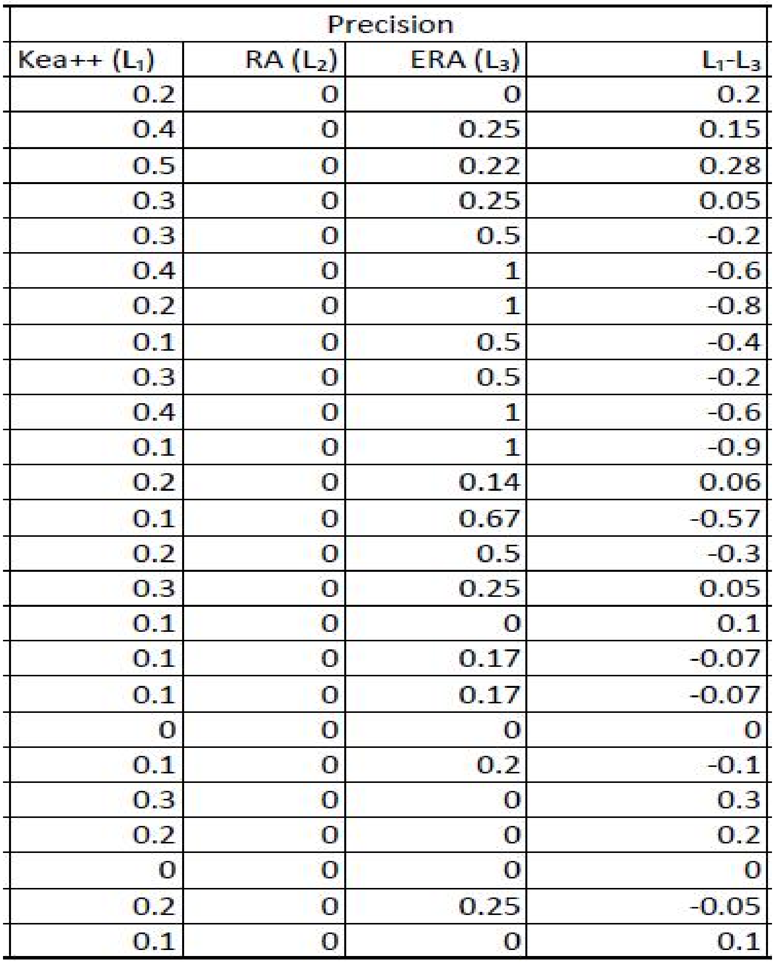


Figure D.12: F-measure t-test (agricultural dataset excluding rule VI)



Precision			
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3
0.2	0	0	0.2
0.4	0	0.25	0.15
0.5	0	0.22	0.28
0.3	0	0.25	0.05
0.3	0	0.5	-0.2
0.4	0	1	-0.6
0.2	0	1	-0.8
0.1	0	0.5	-0.4
0.3	0	0.5	-0.2
0.4	0	1	-0.6
0.1	0	1	-0.9
0.2	0	0.14	0.06
0.1	0	0.67	-0.57
0.2	0	0.5	-0.3
0.3	0	0.25	0.05
0.1	0	0	0.1
0.1	0	0.17	-0.07
0.1	0	0.17	-0.07
0	0	0	0
0.1	0	0.2	-0.1
0.3	0	0	0.3
0.2	0	0	0.2
0	0	0	0
0.2	0	0.25	-0.05
0.1	0	0	0.1

Figure D.13: Precision values (agricultural dataset including rule VI)

(a) t-test to check significant improvement between Kea++ and ERA:
Following are the assumption and claim made for the t-test:

Null Hypothesis: There is no improvement achieved as a result of application of ERA.

Claim: ERA has caused improvement in the results obtained previously from Kea++.

For the sample under test, following are the values of the variables used in t-test:

D_M = -0.1348

µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)

n= 25

$\sum D$ = -3.37

$\sum D^2$= 3.1467

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{3.1467 - (-3.37)^2/25\}/24} = 0.3349$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.1348 - 0)/(0.3349/\sqrt{25}) = -2.01$$

At level of significance 0.05 the critical value is -1.711. Since, -2.01 < -1.711, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.14(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.797. Since, -2.01 > -2.797, so we can say that null hypothesis cannot be rejected and our claim comes out to be false. Figure D.14(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

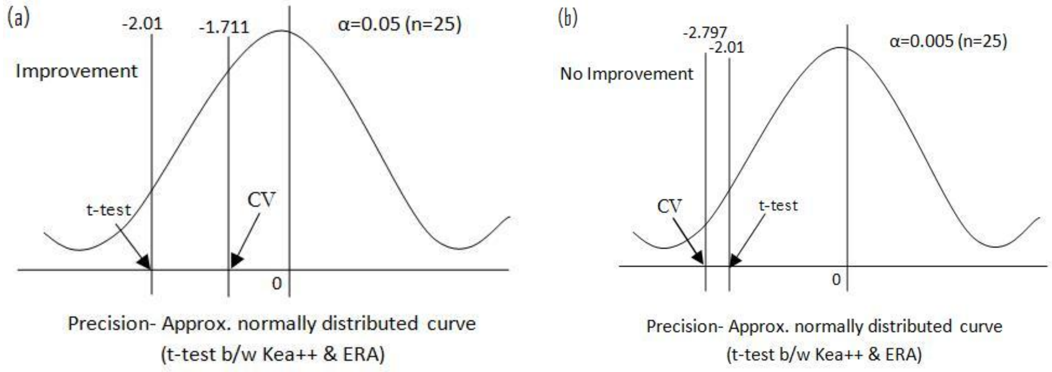


Figure D.14: Precision t-test (agricultural dataset including rule VI)

(II) Recall

Figure D.15 shows the recall value for each test document after applying Kea++, refinement and extended refinement algorithm.

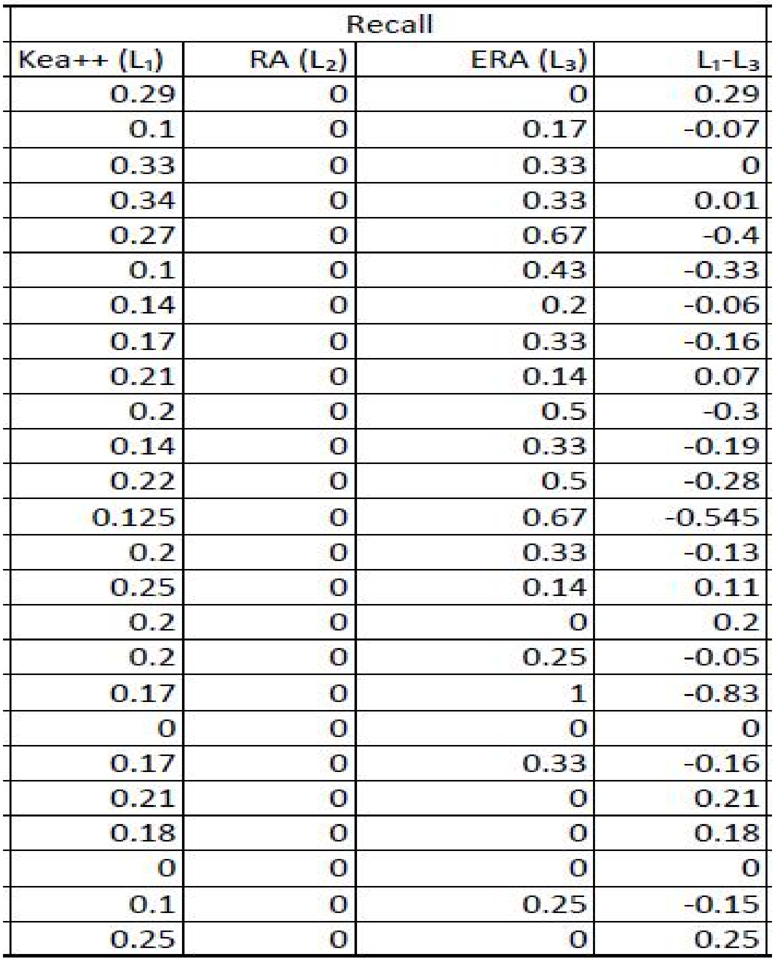
(a) t-test to check significant improvement between Kea++ and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from Kea++.
For the sample under test, following are the values of the variables used in t-test:
D_M = -0.0934
µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 25
$\sum D$ = -2.335
$\sum D^2$= 1.8411

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{1.8411 - (-2.335)^2/25\}/24} = 0.2601$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$



Recall			
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3
0.29	0	0	0.29
0.1	0	0.17	-0.07
0.33	0	0.33	0
0.34	0	0.33	0.01
0.27	0	0.67	-0.4
0.1	0	0.43	-0.33
0.14	0	0.2	-0.06
0.17	0	0.33	-0.16
0.21	0	0.14	0.07
0.2	0	0.5	-0.3
0.14	0	0.33	-0.19
0.22	0	0.5	-0.28
0.125	0	0.67	-0.545
0.2	0	0.33	-0.13
0.25	0	0.14	0.11
0.2	0	0	0.2
0.2	0	0.25	-0.05
0.17	0	1	-0.83
0	0	0	0
0.17	0	0.33	-0.16
0.21	0	0	0.21
0.18	0	0	0.18
0	0	0	0
0.1	0	0.25	-0.15
0.25	0	0	0.25

Figure D.15: Recall values (agricultural dataset including rule VI)

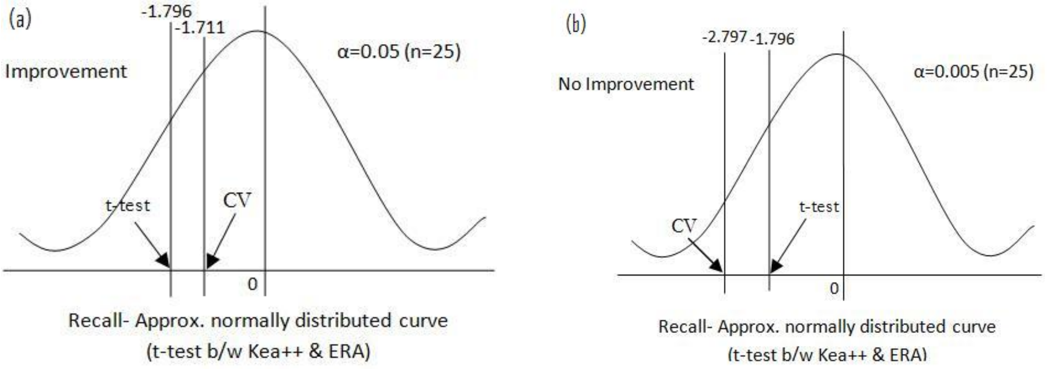


Figure D.16: Recall t-test (agricultural dataset including rule VI)

$$t = (-0.0934 - 0)/(0.2601/\sqrt{25}) = -1.796$$

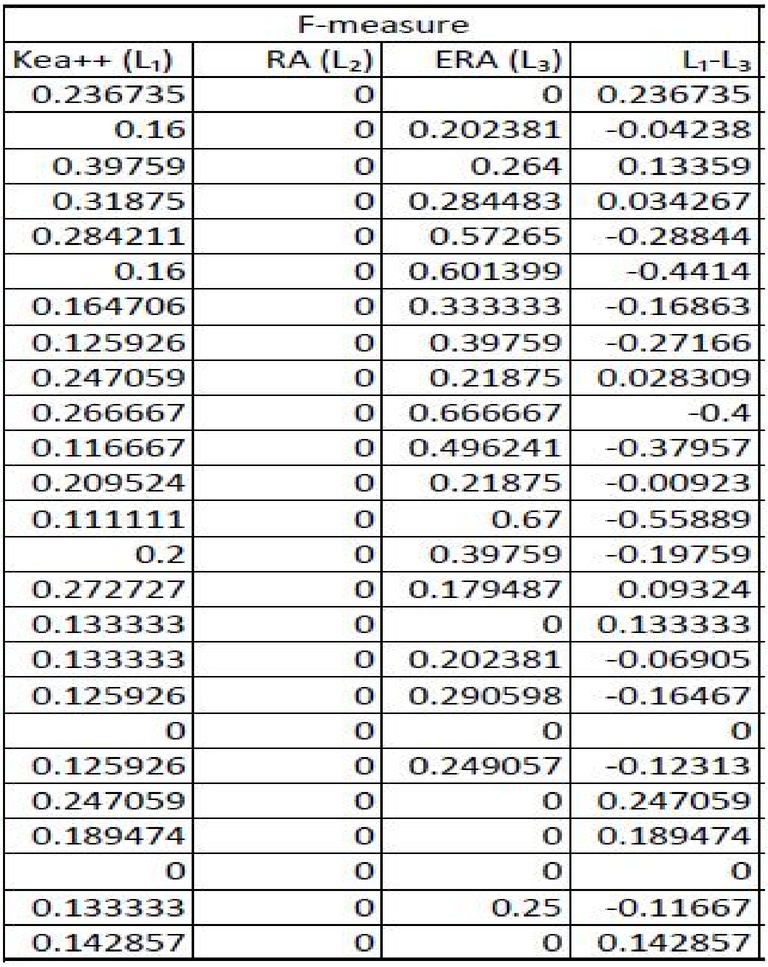
At level of significance 0.05 the critical value is -1.711. Since, -1.796 < -1.711, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.16(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.797. Since, -1.796 > -2.797, so we can say that null hypothesis cannot be rejected and our claim comes out to be false. Figure D.16(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

(III) F-measure

Figure D.17 shows the f-measure value for each test document after applying Kea++, refinement and extended refinement algorithm.

(a) t-test to check significant improvement between Kea++ and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from Kea++.
For the sample under test, following are the values of the variables used in t-test:
D_M =-0.0797



F-measure			
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3
0.236735	0	0	0.236735
0.16	0	0.202381	-0.04238
0.39759	0	0.264	0.13359
0.31875	0	0.284483	0.034267
0.284211	0	0.57265	-0.28844
0.16	0	0.601399	-0.4414
0.164706	0	0.333333	-0.16863
0.125926	0	0.39759	-0.27166
0.247059	0	0.21875	0.028309
0.266667	0	0.666667	-0.4
0.116667	0	0.496241	-0.37957
0.209524	0	0.21875	-0.00923
0.111111	0	0.67	-0.55889
0.2	0	0.39759	-0.19759
0.272727	0	0.179487	0.09324
0.133333	0	0	0.133333
0.133333	0	0.202381	-0.06905
0.125926	0	0.290598	-0.16467
0	0	0	0
0.125926	0	0.249057	-0.12313
0.247059	0	0	0.247059
0.189474	0	0	0.189474
0	0	0	0
0.133333	0	0.25	-0.11667
0.142857	0	0	0.142857

Figure D.17: F-measure values (agricultural dataset including rule VI)

µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 25
$\sum$D = -1.992
$\sum$D^2= 1.318

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{1.318 - (-1.992)^2/25\}/24} = 0.2198$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.0797 - 0)/(0.2198/\sqrt{25}) = -1.813$$

At level of significance 0.05 the critical value is -1.711. Since, -1.813 < -1.711, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.18(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.797. Since, -1.813 > -2.797, so we can say that null hypothesis cannot be rejected and our claim comes out to be false. Figure D.18(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

D.4 Statistical t-test on mathematics domain dataset

(I) Precision

Figure D.19 shows the precision value for each test document after applying Kea++, refinement and extended refinement algorithm.

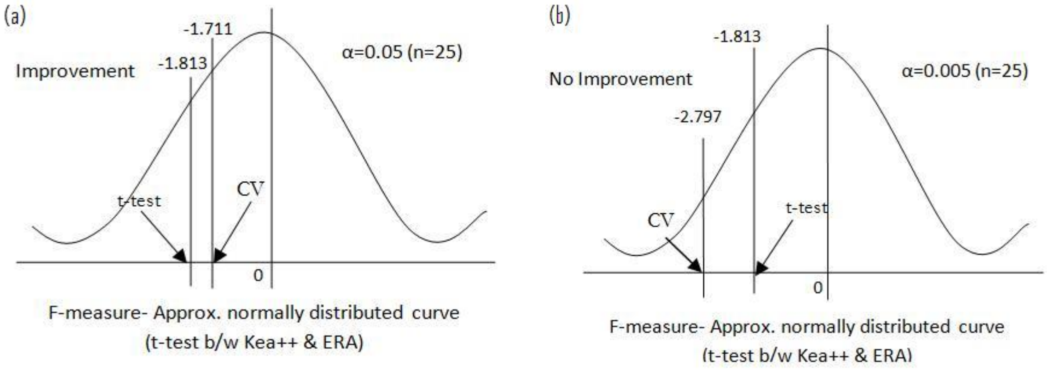
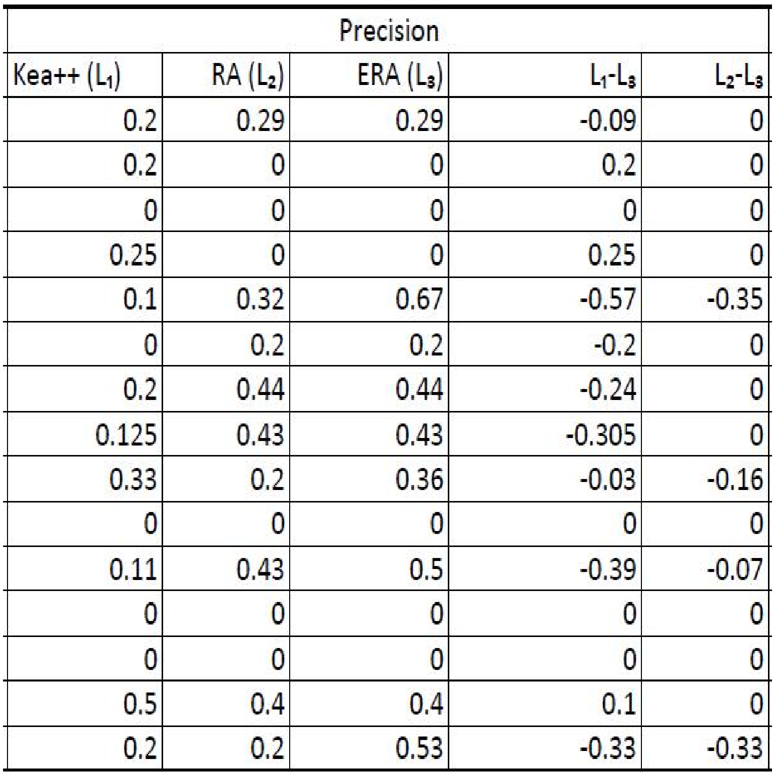


Figure D.18: F-measure t-test (agricultural dataset including rule VI)



Precision				
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3	L_2-L_3
0.2	0.29	0.29	-0.09	0
0.2	0	0	0.2	0
0	0	0	0	0
0.25	0	0	0.25	0
0.1	0.32	0.67	-0.57	-0.35
0	0.2	0.2	-0.2	0
0.2	0.44	0.44	-0.24	0
0.125	0.43	0.43	-0.305	0
0.33	0.2	0.36	-0.03	-0.16
0	0	0	0	0
0.11	0.43	0.5	-0.39	-0.07
0	0	0	0	0
0	0	0	0	0
0.5	0.4	0.4	0.1	0
0.2	0.2	0.53	-0.33	-0.33

Figure D.19: Precision values (mathematics dataset)

(a) t-test to check significant improvement between Kea++ and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from Kea++.
For the sample under test, following are the values of the variables used in t-test:
D_M = -0.107
µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 15
$\sum D$ = -1.605
$\sum D^2$= 0.898

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{0.898 - (-1.605)^2/15\}/14} = 0.2278$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.107 - 0)/(0.2278/\sqrt{15}) = -1.819$$

At level of significance 0.05 the critical value is -1.761. Since, -1.819 <-1.761, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.20(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.977. Since, -1.819 >-2.977, so we can say that null hypothesis cannot be rejected and our claim comes out to be false. Figure D.20(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

b) t-test to check significant improvement between RA and ERA:

Following are the assumption and claim made for the t-test:

Null Hypothesis: There is no improvement achieved as a result of application of ERA.

Claim: ERA has caused improvement in the results obtained previously from RA.

For the sample under test, following are the values of the variables used in t-test:

D_M = -0.0607

µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)

n= 15

$\sum$D = -0.91

$\sum D^2$= 0.2619

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{0.2619 - (-0.91)^2/15\}/14} = 0.1215$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.0607 - 0)/(0.1215/\sqrt{15}) = -1.934$$

At level of significance 0.05 the critical value is -1.761. Since, -1.934 <-1.761, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.20(c) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from RA.

At level of significance 0.005 the critical value is -2.977. Since, -1.934 >-2.977, so we can say that null hypothesis cannot be rejected and our claim comes out to be false. Figure D.20(d) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from RA.

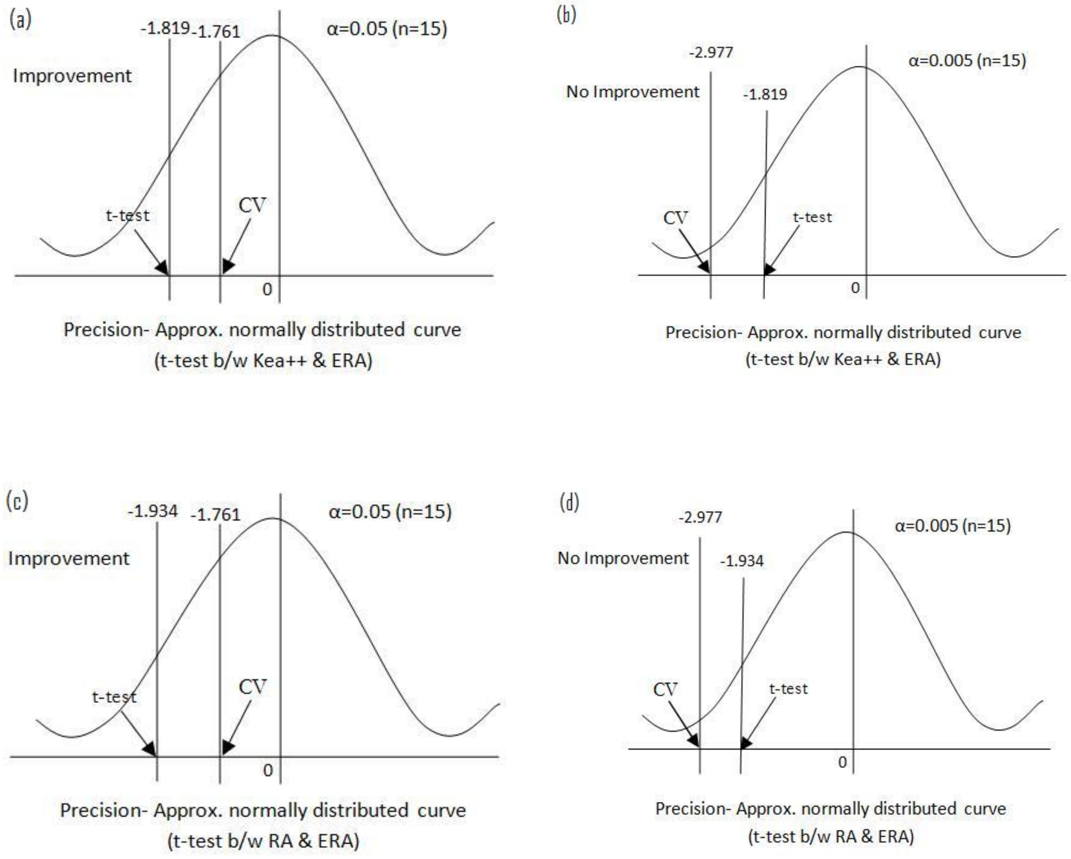
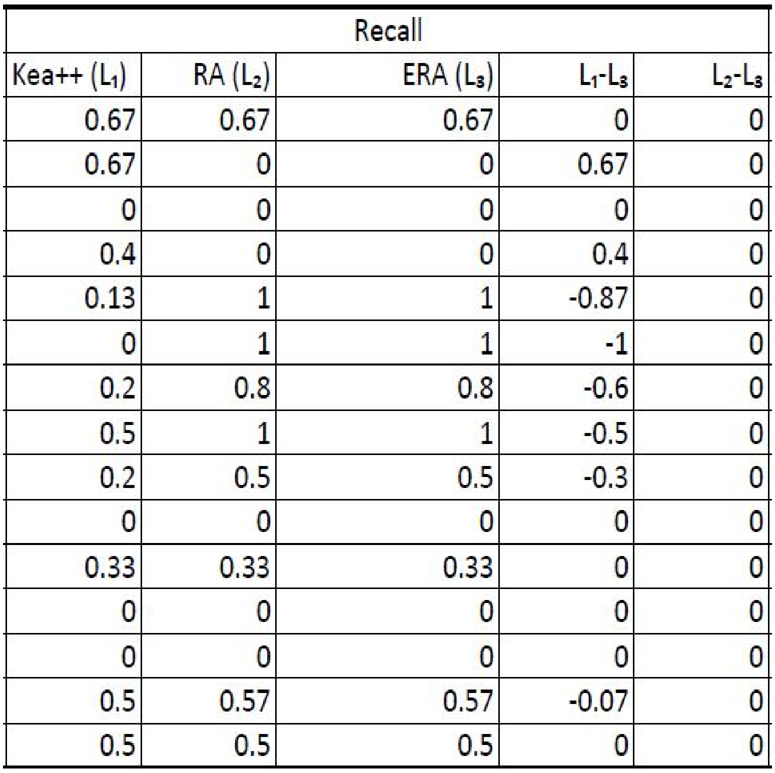


Figure D.20: Precision t-test (mathematics dataset)



Recall				
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3	L_2-L_3
0.67	0.67	0.67	0	0
0.67	0	0	0.67	0
0	0	0	0	0
0.4	0	0	0.4	0
0.13	1	1	-0.87	0
0	1	1	-1	0
0.2	0.8	0.8	-0.6	0
0.5	1	1	-0.5	0
0.2	0.5	0.5	-0.3	0
0	0	0	0	0
0.33	0.33	0.33	0	0
0	0	0	0	0
0	0	0	0	0
0.5	0.57	0.57	-0.07	0
0.5	0.5	0.5	0	0

Figure D.21: Recall values (mathematics dataset)

(II) Recall

Figure D.21 shows the recall value for each test document after applying Kea++, refinement and extended refinement algorithm.

(a) t-test to check significant improvement between Kea++ and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from Kea++.
For the sample under test, following are the values of the variables used in t-test:
D_M = -0.1513
µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 15
$\sum D$ = -2.27
$\sum D^2$=3.0707

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

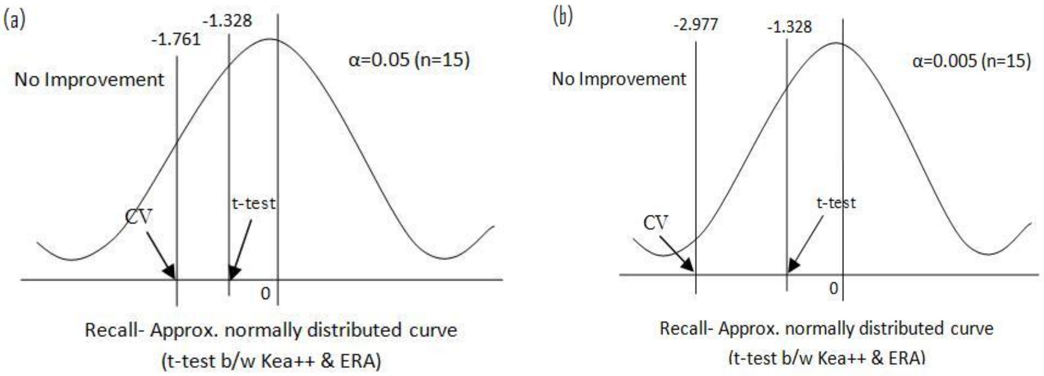


Figure D.22: Recall t-test (mathematics dataset)

$$s = \sqrt{\{3.0707 - (-2.27)^2/15\}/14} = 0.4414$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

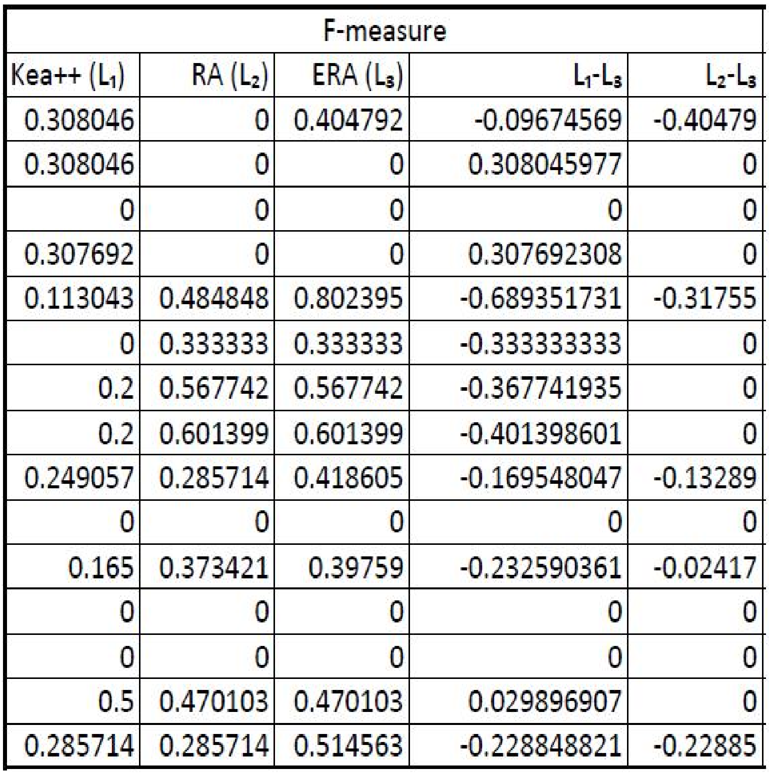
$$t = (-0.1513 - 0)/(0.4414/\sqrt{15}) = -1.328$$

At level of significance 0.05 the critical value is -1.761. Since, -1.328 >-1.761, so we can say that null hypothesis can not be rejected and our claim comes out to be false. Figure D.22(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.977. Since, -1.328 >-2.977, so we can say that null hypothesis can not be rejected and our claim comes out to be false. Figure D.22(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

(III) F-measure

Figure D.23 shows the f-measure value for each test document after applying Kea++, refinement and extended refinement algorithm.



F-measure				
Kea++ (L_1)	RA (L_2)	ERA (L_3)	L_1-L_3	L_2-L_3
0.308046	0	0.404792	-0.09674569	-0.40479
0.308046	0	0	0.308045977	0
0	0	0	0	0
0.307692	0	0	0.307692308	0
0.113043	0.484848	0.802395	-0.689351731	-0.31755
0	0.333333	0.333333	-0.333333333	0
0.2	0.567742	0.567742	-0.367741935	0
0.2	0.601399	0.601399	-0.401398601	0
0.249057	0.285714	0.418605	-0.169548047	-0.13289
0	0	0	0	0
0.165	0.373421	0.39759	-0.232590361	-0.02417
0	0	0	0	0
0	0	0	0	0
0.5	0.470103	0.470103	0.029896907	0
0.285714	0.285714	0.514563	-0.228848821	-0.22885

Figure D.23: F-measure values (mathematics dataset)

(a) t-test to check significant improvement between Kea++ and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from Kea++.
For the sample under test, following are the values of the variables used in t-test:
D_M = -0.1249
µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 15
$\sum D$ = -1.874
$\sum D^2$=1.218

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{1.218 - (-1.874)^2/15\}/14} = 0.265$$

Now for the t-test:

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.1249 - 0)/(0.265/\sqrt{15}) = -1.82$$

At level of significance 0.05 the critical value is -1.761. Since, -1.82 <-1.761, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.24(a) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from Kea++.

At level of significance 0.005 the critical value is -2.977. Since, -1.82 >-2.977, so we can say that null hypothesis can not be rejected and our claim comes out to be false. Figure D.24(b) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from Kea++.

b) t-test to check significant improvement between RA and ERA:
Following are the assumption and claim made for the t-test:
Null Hypothesis: There is no improvement achieved as a result of application of ERA.
Claim: ERA has caused improvement in the results obtained previously from RA.
For the sample under test, following are the values of the variables used in t-test:
D_M = -0.0739
µ = 0 (considering the hypothesis that there will be no improvement or increase in the values after applying the new technique)
n= 15
∑D = -1.1082
$\sum D^2$=0.3353

$$s = \sqrt{(\sum D^2 - \frac{(\sum D)^2}{n})/(n-1)}$$

$$s = \sqrt{\{0.3353 - (-1.1082)/15\}/14} = 0.1345$$

Now for the t-test:

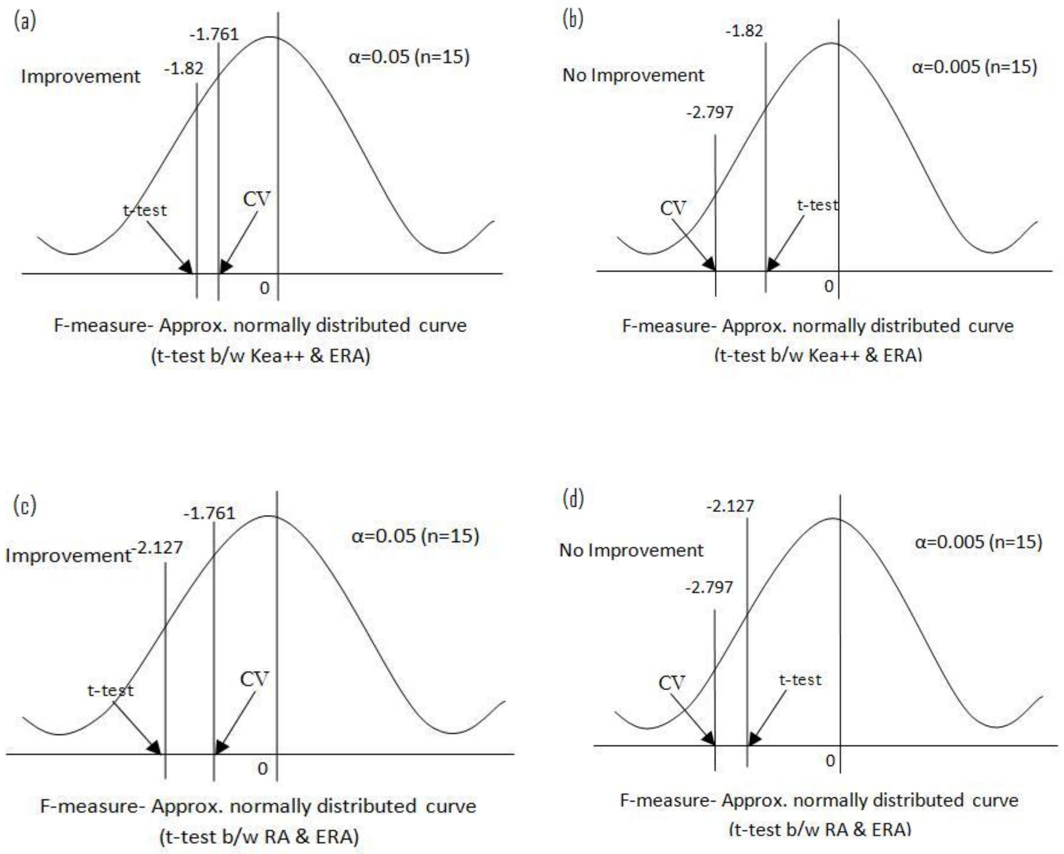


Figure D.24: F-measure t-test (mathematics dataset)

$$t = D_M - \mu/(s/\sqrt{n})$$

$$t = (-0.0739 - 0)/(0.1345/\sqrt{15}) = -2.127$$

At level of significance 0.05 the critical value is -1.761. Since, -2.127 <-1.761, so we can say that null hypothesis can be rejected and our claim comes out to be true. Figure D.24(c) shows the occurrence of the critical value and the t-test value. So we can say that ERA has caused significant improvement in the results obtained previously from RA.

At level of significance 0.005 the critical value is -2.977. Since, -2.127 > -2.977, so we can say that null hypothesis can not be rejected and our claim comes out to be false. Figure D.24(d) shows the occurrence of the critical value and the t-test value. So we can say that ERA has not caused significant improvement in the results obtained previously from RA.