ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

TERM RE-WEIGHTING BASED QUERY EXPANSION APPROACH FOR AMHARIC INFORMATION RETRIEVAL

BY

ZELALEM ADDIS ALEMU

A THESIS SUBMITTED TO THE SCHOOL OF INFORMATION SCIENCE OF ADDIS ABABA UNIVERSITY IN A PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE

February, 2014

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

COLLEGE OF NATURAL SCIENCE

SCHOOL OF INFORMATION SCIENCE

TERM RE-WEIGHTING BASED QUERY EXPANSION APPROACH FOR AMHARIC INFORMATION RETRIEVAL

BY

ZELALEM ADDIS ALEMU

Signature of the board of examiners for approval

Dereje Tefri (PhD) Advisor _____

Solomon Tegegne (PhD) _____

Adey Edessa _____

_____ _____

DECLARATION

This thesis is my original work and has not been submitted as a partial fulfillment of requirement for the degree in any university

Zelalem Addis Alemu

February, 2014

The thesis has been submitted for examination with our approval as university advisors.

March 3/2014

Dereje Teferi (PhD)

Table of contents

Contents

List of Tables

List of Figures

List of Figures

List of Algorithm

List of Acronyms

CONE..... Conceptual Network Editor

EMC..... EMC Corporation

GUI .....Graphical Users Interface

IDE..... Integrated Development Environment

IDF ..... Inverse Document Frequency

IR ..... Information Retrieval

ITF ..... Inverse Term Frequency

MIDF.....Minimum Inverse Document Frequency

NDC ..... Number of Documents in a Class

SCT ..... Select Common Terms

SMART.....System for the Mechanical Analysis and Retrieval of Text

SVD ..... Singular Value Decomposition

QE.....Query Expansion

TREC ..... Test Retrieval Conference

WWW..... World Wide Web

ACKNOWLEDGMENTS

I would like to express my gratitude to all the people who supported and accompanied me during the progress of this thesis.

It is my pleasure first to express my deepest gratitude to my advisor Dereje Teferi (PhD) for his invaluable advice and guidance. He listened to all my problems I faced during this thesis and showed me the way to overcome them. He has been providing me constructive comments for the betterment of this study.

I would also like to thank Million Meshesha (PhD) for his support, constructive suggestion, constant support and encouragement at the very beginning of my research.

Finally, I extend my heartfelt thanks and respect to all those people who were not mentioned here but their contributions have been inspiring for the completion of this work. In the end, I would like to thank my almighty God for giving me the strength to achieve whatever I have achieved so far.

Abstract

This research has been conducted, aiming at augmenting the precision while lessening the original recall of an Amharic IR system. The main reason for performing query expansion is to provide relevant documents as per user's query that can satisfy their information need. Because users are not fully knowledgeable about the information domain area, they mostly formulate weak queries to retrieve documents. Thus, they end up frustrated with the results found from an IR system. Some of the causes for this type of problem are, polysemous and synonymous terms, which require an integration of query reformulation strategy to the IR system. The present study has explored term re- weighting based query expansion approaches that integrate term re-weighting with Statistical co-occurrence analysis, bi-gram analysis and bi-gram thesaurus methods. In this approach, the users' initial queries and the documents during the user relevance feedback are represented as vectors respectively, and the similarity between them can be obtained by calculating the vector similarity. Then we re-weight the terms through one single document and the entire document set using Rocchio's reweighting scheme respectively, final weight of the term can be obtained by the combination of the above weights; finally the terms with larger weight will be selected as query expansion terms and then fed to the three query expansion techniques for expansion. Statistical co-occurrence analysis and bi-gram analysis use the pseudo-relevance feedback, while the bi-gram thesaurus method utilizes a pre-built thesaurus to expand queries. Both bi-gram analysis and bi-gram based thesaurus use the underling theory of expanding a query, using terms that appear adjacent to a query term frequently. On the other hand, statistical co-occurrence method considers, frequently appearing terms with the query term, regardless of their position. Term re-weighting, three proposed query expansion techniques were integrated to an information retrieval system. Then test result showed that bi-gram method outperformed the other two, and scored 2% improvement in total F-measure. The performance of the system can further be improved by designing ontology based query expansion in order to control expanding terms that are polysemous by themselves.

Keywords: Information Retrieval, Polysemous terms, Term re-weighting, Query Expansion, Bi-gram, Thesaurus, Statistical Co-occurrence

Chapter one

1. Introduction

1.1. Background of the study

A considerable amount of explicit knowledge is scattered throughout various documents within the organizations and the mind of people minds working there. In many cases the possibility to efficiently access (retrieve) and reuse this knowledge is limited [1]. As a result of this, most knowledge is not sufficiently exploited, shared and subsequently forgotten in relatively short time after it has been introduced or invented/discovered within the organization. Therefore, in the information society, it is vitally important for knowledge-intensive organizations to make the best use of information gathered from various information resources inside the organizations and from external sources like the internet. On the other hand, tacit knowledge of authors of the document' provides important context to them, which cannot be effectively intercepted.

Before 1990s most people preferred getting information from others rather than information retrieval system [2]. However, after optimization of the effectiveness, information retrieval systems become the most preferred means of accessing information by people. Therefore, the need to store and retrieve written information effectively and efficiently has become increasingly important. Nowadays, information is everywhere that helps people to make the correct decision at the right time. Information retrieval systems are designed, because of the fast growing electronic or digital information worldwide. The recent decades has witnessed, an explosive growth of online information including web pages, news, articles, email, messages, scientific literature and information about all kinds of product on World Wide Web [3].

Search engines have become the most helpful tools for obtaining useful information ever since the development of the World Wide Web. But, the search engines sometimes fail to cater to the users need. The large volume of information accessible over networks makes it difficult for the user to find the exact information needed.

Information retrieval (IR) is the process of finding relevant documents from unstructured document collection that satisfies information need of users [2]. According to Ricardo and Ribeiro-Neto [4], "Information retrieval deals with the representation, storage, organization and access to information items". While the representation and organization of the information items provides the user with easy access to the information he/she is seeking for, the storage of information allows accumulating knowledge to make documents easily accessible for later use. In addition, the access to information item provides the user with effective and efficient retrieval of information based on their need.

Query expansion is the process of reformulating a seed query to improve the performance in information retrieval operations. In the context of web search engines, query expansion involves evaluating a user's query and expanding the search query to match additional documents. Query reformulation involves the following techniques [5].

- Finding synonyms of a word and searching for the synonyms as well.
- Finding all the various morphological forms of a word in a search query.
- Fixing spelling errors and automatically searching for the corrected form or suggesting it in the results.
- Re-weighting the terms in the original query.
- Query expansion.

The main reason behind the development of term re- weighting based query expansion approach is to find a solution to a synonyms and polysemous words and to retrieve relevant Amharic documents based on the users need.

1.2. Statement of the problem

There are more than 80 languages in Ethiopia and Amharic is the mother tongue language for more than 25 million people [6]. According to Federal Democratic Republic of Ethiopian (FDRE) population census commission [7], it is the first language for more than 20 million and second language for over 5 million people.

Now a day, there are a huge amount of Amharic documents produced in electronic form [8]. The documents contain data about agriculture, water resource, tourism and business development.

Accordingly, there are huge collections of documents available in the form of books, magazines, newspapers, novels, officials and legal documents, etc.

The importance of developing an effective information retrieval system that enables searching and retrieving relevant document written in Amharic language in Geez alphabet arises from alleviating solutions for synonyms and polysemous words.

Several works have been done on Amharic information retrieval system, such as: N-Gram based automatic indexing for Amharic text Bethlehem M. 2009 [9], Amharic text retrieval using latent semantic indexing with singular value decomposition Tewodros H. 2003 [10], design and implementation of Amharic search engine Tesema M. and Solomon A. [11], query expansion for Amharic document retrieval Alemayehu .T 2010 [12] and semantic based query expansion technique for Amharic information retrieval Abiy B. 2011[6]. That attempts to retrieve relevant documents as per the information need of users' expectations. Query formed by search engine users, are generally short and vague. It is very difficult to estimate the exact user need. Sometimes, the queries are well formed and clear, but the document collection does not contain the information in the same form as that of the query. Synonymous terms or related terms are present in the collection. Query expansion adds terms to the original query, which provides more information about the user need.

Query expansion is an effective technique to improve the performance of information retrieval system. Studies have shown that the average query length is only around 2-3 words. Due to this, it is very difficult to understand the user intention behind the query and to provide the ideal set of relevant documents [13].

Alemayehu's work [12] tried to enhance the system's recall at the expense of its precision. The performance analysis shows that there is an enhancement of recall from an average of 0.29% to 0.73%. On the other hand, because of the tradeoff between recall and precision and because of polysemous query term existence, his proposed system decreased the overall precision from 0.91% to 0.57% on average before and after using query expansion. While there is a decline of precision because of the heterogeneous nature of the cluster enhancing recall of the retrieval system while decline the precision and thesaurus terms have no related concepts with the query terms that are generated by the automatic query expansion.

Abiy's [6] has done semantic based query expansion for Amharic information retrieval; he tried to enhancement the system performance of precision but cannot match the recall level of the original query. He used statistical co-occurrence, bi-gram methods and bi-gram thesaurus. In statistical co-occurrence method precision improved by 14% while refined query, bi-gram thesaurus has all improved by 18%. Even though the above researcher tried to implement different methods to enhance the performance of the Amharic document retrieval by using query expansion, there are problem of handling polysemous words in order to retrieve Amharic document. In addition to query expansion, term re-weighting is important to overcome the problem of retrieving relevant documents from Amharic corpus, so the researcher is applying a term re- weighting based query expansion approach that have a paramount importance to enhance the performance of Amharic document retrieval system by integrating query expansion with term re-weighting.

To this end this research attempts to address the following main research questions:

- Dose a term re- weighting based query expansion approach improve effectiveness of the information retrieval system in searching for relevant Amharic documents?
- What kinds of techniques are more appropriate to handle the challenges of polysemous query terms?
- How query terms with synonymous and polysemous will be controlled in order to enhance both precision and recall of Amharic document retrieval?
- What is the testing and evaluating metrics to compare and contrast the proposed techniques?
- How to integrate query expansion, term re-weighting with the current information retrieval system?

1.3. Objective of the study

The following general and specific objectives are outlined in order to reach a level of conceptual understanding towards solving the research problems.

1.3.1. General objectives

The main objectives of this research is to design a term re- weighting based query expansion approach as well as to integrate it to the existing Amharic information retrieval system so that a better set of more relevant documents are retrieved as per the information need of users.

1.3.2. Specific objectives

- To understand concepts of a term re- weighting based query expansion approach and review related works from literatures to define the contribution of this study.
- To explore and identify the linguistic features of the Amharic language applicable to a term re- weighting based query expansion approach.
- To investigate the available techniques for query expansion and term re-weighting.
- To integrate query expansion, and term re-weighting with the current information retrieval system.
- To evaluate the performance of the system using information retrieval effectiveness measures such as recall, precision and f-measure.

1.4. Scope and limitation of the study

This research is a continuation of a research done by Abiy [6]. Hence, in this research the researcher will not design or implement any part of the original Amharic IR system, rather a term re- weighting based query expansion technique is designed for the Amharic IR. Term re-weighting based query expansion considers synonymous terms in general, and polysemous terms in particular. Techniques such as term re-weighting integrated with statistical co-occurrence, bi-gram and bi-gram thesaurus that use statistical information of terms and documents are tested to investigate their capability to find best weighted terms for expansion.

For testing purposes, queries that have polysemous and synonymous terms are formulated in order to challenge the Amharic IR system and measure the extent to which it can discriminate their various meanings.

There are limitations in this research regarding its scope such as lack of best Amharic IR system that have a good stemmer and inconvenience of using a big size document corpus.

1.5. Methodology

According to Abiy et al. [14], methodology is the theory and analysis of deciding how research should proceed, and it involves analysis of the principles and procedures followed in a research. Methodology covers the entire approach of research. Therefore, in this study methodology is used to gain understanding of phenomenon, to produce a better quality documentation, to ensure consistency and requirements met completely. The following methods and procedures are used in order to achieve the stated objectives of this study.

1.5.1. Literature review

In order to have deeper understanding on philosophies, principles/theories and techniques of information retrieval system, particularly on a term re- weighting based query expansion approach, extensive literature review were conducted from books and Internet. Previous related research works, printed materials such as, journal articles, conference papers and electronic materials on the web were assessed. Additional literature review and document analyses are made to investigate and identify the feature of Amharic text, which is important to the research in the course of Amharic text operations.

1.5.2. Dataset collection

By consulting the domain expert in Amharic language and phonetics, characteristics of the language have been studied and data selection is carried out accordingly. The data selected constitutes polysemous and synonym words and word variants or phrase of the Amharic language. Moreover it is checked that the document corpora comprises the polysemous query terms selected for testing. Finally a relevance judgment, stating which documents are relevant as per all testing queries is prepared.

In this research, Amharic documents are collected from Amharic Bible Old Testament taken as a document corpus. The corpus is collected by downloading Iota software from internet. It contains 21,000 steamed terms and 930 documents. [15]. Bethlehem [9] and Tewodros [10] and Amanuel [16] used 100, 200 and 300 collections of local news articles respectively.

1.5.3. Implementation tools

To develop the prototype, Java programming language is used. Java is a programming language developed by Sun Microsystems. It implements a strong security model, which prevents compiled Java programs from illicitly accessing resources on the system where they execute or on the network. Popular World-Wide Web browsers, as well as some World-Wide Web servers and other systems implement Java interpreters. These are used to display interactive user interfaces, and to script behavior on these systems.

Java is selected for its suitability of processing Unicode encoded text, its ease of use and its object oriented characteristic. Moreover, the researcher has a good java programming knowledge in order to prepare a prototype, to meet the research objective and solve the research problem. In addition the original Amharic IR system adopted from Alemayhu [12] and Abiy [6] are developed using Java and thus, it is convincing to use Java for compatibility issues [17].

Java NetBeans 6.9.1 and supporting package Lucene 2.1.4 are used specifically. Java NetBeans is not only free but it has also some excellent features like, well-thought out exception handling, impressive IDE capability, good GUI features and its small size developed applications that are portable and can run in almost any kind of platform [18].

1.5.4. Testing procedure

To test the system, Amharic queries are formulated, terms are selected, terms are re-weighted using Rocchio Beta formula, queries are expanded, and relevance judgment is prepared to construct document query matrix that shows all relevant documents for each test queries. To assess the effectiveness of the system (i.e. the quality of its search results) the most frequent and basic statistical measures, recall, precision and F-measure are used.

Recall is percentage of relevant documents retrieved from the database in response to users query. Precision is percentage of retrieved documents that are relevant to the query. F-measure is a weighted harmonic mean of precision and recall [2].

1.6. Significance of the study

The main significance of the study is designing a term re- weighting based query expansion approach for Amharic language that can help users search and retrieve relevant documents written in Amharic language. It gives direction for researchers to select the best model for future works on Amharic document retrieval based on the performance achieved. Additionally, it is an academic exercise to fulfill the requirement of masters program the researcher is enrolled in.

1.7. Organization of the research

This research is organized in to five chapters. Chapter one unfolds the research background, providing incite to the problem area. In addition, the research objective, research question, scope and limitation, and significance of the study are outlined peculiarly. The research problem is depicted with respect to the drawbacks of the previous study and what should be covered in the current research.

Chapter two discusses the literature review covering the various theories, facts, techniques, methods and mathematical equations from various researches, text books, published journals and the internet. It includes overview of general consensuses, facts, ranking algorithms, term selection ideas and evaluation mechanisms. In relation with query expansion, Historical background of Amharic language, the Amharic writing system, and characteristics of ambiguous Amharic terms and other related topics are covered. In addition, it gives incite to what have been done so far regarding Amharic IR and the Amharic language.

Design of term re- weighting based query expansion techniques is covered in chapter three. The architectural view of query expansion model and a term re- weighting based query expansion model is outlined. Techniques like query refinement and how the proposed techniques work are discussed.

Experimental results measured in recall, precision and f-measure is recorded for each technique on chapter four. It further explains the interpretation of the results to answer the research questions. It also discuses challenges occurred while testing and limitations of the techniques, so that research recommendations can be specified.

Finally, chapter five presents conclusion drawn from the findings of the study and recommendations that should be entertained in future researches.

Chapter two

2. Literature review

2.1. Brief History of IR systems

The practice of archiving written information can be traced back to around 3000 BC, when the Sumerians designated special areas to store clay tablets with cuneiform inscriptions. Even then the Sumerians realized that proper organization and access to the archives was critical for efficient use of information. They developed special classifications to identify every tablet and its content. The need to store and retrieve written information became increasingly important over centuries, especially with inventions like paper and the printing press. Soon after computers were invented, people realized that they could be used for storing and mechanically retrieving large amounts of information. In 1945 Vannevar Bush published a ground breaking article titled "As We May Think" that gave birth to the idea of automatic access to large amounts of stored knowledge [19]. In the 1950s, this idea materialized into more concrete descriptions of how archives of text could be searched automatically. Several works emerged in the mid 1950s that elaborated upon the basic idea of searching text with a computer. There has, however, been a tension throughout the life of information retrieval between simple statistical methods and sophisticated information analysis. This dates to a memo by Warren Weaver in 1949 [Weaver 1955] thinking about the success of computers in cryptography during the war, and suggesting that they could translate languages. He wrote "it is very tempting to say that a book written in Chinese is simply a book written in English which was coded into the 'Chinese code." If we have useful methods for solving almost any cryptographic problem, may it not be that with proper interpretation we already have useful methods for translation?" Just as Vannevar Bush's paper began hypertext, Warren Weaver's paper began research in machine translation [20].

One of the most influential methods was described by H.P. Luhn in 1957, in which (put simply) he proposed using words as indexing units for documents and measuring word overlap as a criterion for retrieval[4]. Several key developments in the field happened in the 1960s. Most notable were the development of the SMART system by Gerard Salton and his students, first at Harvard University and later at Cornell University; [21] and the Cranfield evaluations done by Cyril Cleverdon and his group at the College of Aeronautics in Cranfield [22]. The Cranfield

tests developed an evaluation methodology for retrieval systems that is still in use by IR systems today. The SMART system, on the other hand, allowed researchers to experiment with ideas to improve search quality. A system for experimentation coupled with good evaluation methodology allowed rapid progress in the field, and paved way for many critical developments [23].

The 1970s and 1980s saw many developments built on the advances of the 1960s. Various models for doing document retrieval were developed and advances were made along all dimensions of the retrieval process. These new models/techniques were experimentally proven to be effective on small text collections (several thousand articles) available to researchers at the time. However, due to lack of availability of large text collections, the question whether these models and techniques would scale to larger corpora remained unanswered. This changed in 1992 with the inception of Text Retrieval Conference, or TREC. [24] [31] TREC is a series of evaluation conferences sponsored by various US Government agencies under the auspices of NIST, which aims at encouraging research in IR from large text collections.

With large text collections available under TREC, many old techniques were modified, and many new techniques were developed (and are still being developed) to do effective retrieval over large collections. TREC has also branched IR into related but important fields like retrieval of spoken information, non-English language retrieval, information filtering, user interactions with a retrieval system, and so on. The algorithms developed in IR were the first ones to be employed for searching the World Wide Web from 1996 to 1998. Web search, however, matured into systems that take advantage of the cross linkage available on the web [25].

Information retrieval is a key technology that has been made in the history of humankind. It is the key technology behind search engines and an everyday technology for many web users [26] [30].

2.2. IR Models

D.Hiemstra 2009 [26] stated that, IR model is the mechanism of predicting and explain the need of the user given query to retrieve relevant documents from a collection. IR models serves as blueprint so as to develop applicable IR system. In addition to that, IR models guide the matching process to retrieve a ranked list of relevant document given a query. IR models are

broadly categorized in to two approaches, which are semantic approach and statistical approach. Semantic approaches models such as, latent semantic indexing and neural network try to work on syntactic and semantic analysis. They attempt to implement some degree of understanding the natural language text that users provide. On the other hand, statistical approaches such as, vector space model and probabilistic model attempts to retrieve documents that are highly ranked in terms of statistical measure [27].

The three most widely used information retrieval models that are bases on statistical approaches are Boolean, vector space, and probabilistic [4]. The three IR models again can be classified based on their mathematical basis. In the Boolean model, a set of index terms are used to represent document and query. Therefore, the mathematical basis can be called set theoretic. In the vector space model, a vector in a t-dimensional space is used for representing document and query, therefore, the model is algebraic. In the probabilistic model, probability theory is used for representing document and query. Therefore, the model is probabilistic. Different scholars have proposed several alternative approaches based on their mathematical basis. For set theoretic models, the alternatives are; the fuzzy and the extended Boolean models. For algebraic models, the alternatives are; the generalized vector, the latent semantic indexing, and the neural network. For probabilistic model, the alternatives are; the inference network and the belief network models [4].

2.2.1. Boolean Model

The Boolean model is the first and the simplest model of information retrieval. It works based on set theory and Boolean algebra which allowed users to specify their information need using a combination of classical Boolean operators, ANDs, ORs and NOT [2][29][32]. For instance, query t1 and query t_2 (t1 AND t2) is only satisfied by a given document D1 if and only if D1 contains both terms t1 and t2. Likewise, the query "t1 OR t2" is satisfied by D1 if and only if it contains t1 or t2 or both. On the other hand, the query "t1 AND NOT t2" satisfies D1 if and only if it contains t1 and does not contain t2. The model views each document as just a set of words [28].The weight for index terms in Boolean retrieval model are all binary (i.e. W_{ij} E{0, 1}. The similarity of a document dj to the query q is expressed as: $sim\ dj, = 1 = if\ document\ satisfies\ the\ boolean\ query\ 0 = Otherwise$ Where, if similarity of document dj to the query q is equal to

1 (present), the document dj is relevant. Whereas, if similarity of document d_j to the query q is equal to 0(absent) the document is not relevant [4].

Boolean model have an advantage of giving users a sense of control over the system. It is immediately clear why a document has been retrieved given a query. If the resulting document set is either too small or too big, it directly clear which operators will produce respectively a bigger or smaller set. The other advantage of this model is the clean formalism behind the model and its simplicity [4] [27]. However, the model has a number of clear disadvantages. First, most user find it difficult to translate their information need into a Boolean expression. Second, the model has not follow document ranking principle by nature, which means all retrieved documents are considered equally important. Third, the model either retrieves too few or too many documents, or sometimes not retrieves any, which might lead to null result. In addition it's sometimes creates its own problems, such as, misunderstanding and misrepresentation of the users information needs [4] [32]. Several attempts has been done so far to make the model effective by developing different alternatives of the model such as fuzzy set model and the extended Boolean model. However, IR communities consider that Boolean system is less effective than ranked retrieval system [33]. Accordingly, a number of models have been proposed for this process; the widely used IR models are the Vector Space Model and the Probabilistic Model [27].

2.2.2. Vector Space Model

Vector space model is one of the commonly used statistical approaches to represent each textual document as a set of terms [26]. In this model, documents and terms are represented as vectors in a k-dimension space where each dimension corresponds to a possible document feature. The word "terms" is not inherent in the model, but terms are typically words and phrases. The values assigned to elements in the vector space describe the degree of importance of term in representing the semantics of the document. Sometimes a given term may receive a different weight in each document in which it occurs. If the term is not appearing in a given document the weight of that term will be zero in that document [4]. For a given document (dj) the weight of the terms in it can be expressed as the coordinates of dj in the document space. A document collection containing a total of documents ("d") described by terms ("t") is represented as term by document matrix (T x D). Each row of this matrix is a term vector and each column of this

matrix is a document vector. The element at row i, column j, is the weight of term j in document I [2].

Term list	Doc1	Doc	Doc3	Doc4	.....	Doc n
Term 1	W11	W 12	W13	W14	.....	W1n
Term 2	W 21	W 22	W 23	W 24	.....	W 2n
Term 3	W31	W 32	W33	W34	.....	W3n
Term 4	W41	W 42	W43	W44	.....	W4n
Term 5	W51	W 52	W53	W54	.....	W5n
.....	.....	.....	.....	.....	.....	.....
Term m	Wmn	Wmn	Wmn	Wmn	.....	Wmn

Table 2.1: Example of Term Document Matrix

In this model, query is interpreted as another document in document space. If the term that is not in the collection but appear in the query, this will add additional dimension in the document space [4]. The weight of terms in the documents or in the queries assigned by using term frequency (TF) and inverse document frequency (TF*IDF) scheme which are the most successful and widely used automatic generation of weights [4]. The term frequency is the frequency of occurrence of the given term within the given document. This scheme attempts to measure the degree of importance of the term within the given document. Inverse document frequency (idf) is a frequency of a given term within an entire collection of documents. It attempts to measure how widely the term is distributed over the given collection.

The similarity of document and query in vector space model is determined by correlation between the vectors dj and vector q. The correlation is quantified by the associative coefficients based on the inner product (dot product) of the document and query vector, where documents whose vectors are close to the query vector are more relevant to the query than documents whose vectors are far from the query vector [4]. As discussed in section 2.1.3 there are different similarity measurements. However, the most widely used by vector space model and popular similarity measure is the cosine coefficient, which measures the angle between the document vector and the query vector [4]. According to [4], Vector space models (VSM) have several advantages. First, it improves retrieval performance using its term-weighting scheme. Second,

the partial matching strategy of VSM allows retrieval of document approximate the query condition. Third, it sort and rank the documents according to their degree of similarity to the query using cosine similarity measurement. Finally, it is simple to implement and fast. However, the model has also several disadvantages. It considers terms as unrelated objects in the semantic space. This means, if no common words are shared between the query and documents in text collection, the similarity value will be zero and no document will be retrieved. This is usually happened when users and authors prefer to use different words which have the same meaning [4]. Vector space model is not attempt to define uncertainty in IR system and its ranked answers sets is difficult to improve upon without the integration of query expansion and reformulation modules to the model [34]. Literature suggested that one of the alternative modeling paradigm, which is capable of solving the above mentioned problem of vector space model is probabilistic IR model [34][35].

2.2.3. Probabilistic Model

In information retrieval process, user first start with information needs which is then translated into query representation. On the other side, documents are also translated into document representation. Finally, the system attempts to determine the relevance of the document to the information need of the users. In IR models such as, Boolean and Vector space model, given a query and document representation matching is done without considering the semantic relationship between query and documents. IR systems build upon those models has an uncertain guess of whether a document has content relevant to the information need. However, probabilistic theory provides a principled foundation for such reasoning under uncertainty [2] [34]. Maron and Kuhns was the first to introduce ranking by the probability of relevance, soon after Stephen Robertson brought new idea called, probabilistic ranking principle in 1977 [29] [4] that a document di will be judged relevant by the user with respect to query q. which is expressed as, P(R|q,di), where, R is the set of relevant document. Typically, in probabilistic model, based on the query of user the documents are divided in to two parts. The first contain relevant documents and the second contain non-relevant (irrelevant) documents. However, the probability of any document is relevant or irrelevant with respect to user query is initially unknown. Therefore, the probabilistic model needs to guess at the beginning of searching process. The user then observe the first retrieved documents and gives feedback for the system by selecting

relevant documents as relevant and irrelevant documents as irrelevant. By collecting relevance feedback data from a few documents, the model then can be applied in order to estimate the probability of relevance for the remaining documents in the collection. This process iteratively applied to improve the performance of the system so as to retrieve relevant documents which satisfies users need [20]. In probabilistic model, the order in which documents are presented to the user is to rank documents by their estimated probability of relevance with respect to the user information need. The principle behind this assumption is called, probability ranking principle (PRP) [2][4]. Probability ranking principle asserts that [36]: If a reference retrieval system's response to each request is a ranking of the documents in the collections in order of decreasing probability of usefulness to the user who submitted the request, where the probabilities are estimated as accurately as possible on the basis of whatever data has been made available to the system for this purpose, then the overall effectiveness of the system to its users will be the best that is obtainable on the basis of that data Several retrieval models have been developed, which has a probabilistic basis. The mostly used and recent developed methods of probabilistic model are, Binary Independent Model, Inference Network Model and Belief Network Model [4].

- **Binary Independent model**

The classical Binary independence model (BIM) was introduced in 1976 by Roberston and Sparck Jones [4]. According to Greengrass [27], the model has been called "binary independence" because it has been assumed that to arrive at the model one must make the simplifying assumption that the document properties that serve as clues to relevance are independent of each other in both the set of relevant documents and the set of non-relevant documents. In BIM, binary is equivalent to Boolean; queries and documents are represented as binary incidence vectors of terms. D= $\{d_1, d2,\ldots,d_n\}$ where, di=1 if term i is present in document d and $_{di}$=0 if term i is not present in document d. Moreover, query q represented by the incidence vector. As expressed above, in BIM model, the probability of P (R|d,q) that a document is relevant through the probability in terms of term incidence vectors P(R|x ,q) in both document and query.

- **Bayesian Networks Model**

Bayesian networks model was introduced by Turtle and Croft [2] as information retrieval model on the basis of probabilistic theory. Bayesian networks use a form of probabilistic graphical model. They use directed acyclic graphs (DAGs) to show probabilistic dependencies between variables, in which the nodes represent random variables, the arcs depict causal relationships between these variables. They have child and parent causal relationship, which is represented by linking each parent node to the child node. Only the root nodes are without parents [4].

- **Bayesian Inference Network Model**

According to Baeza-Yates and Ribeiro-Neto [4], there are two traditional schools of thought in probability which are based on the frequentist view and the epistemologist view. The frequentist views probability as a statistical notion related to the laws of chance. The epistemologist views probability as a degree of belief whose specification might be devoid of statistical experimentation.

Bayesian Inference Network model is built up on epistemologist view of the information retrieval problem. It associates random variables with the index terms, the documents and the user queries. The model computes P(I|D), the probability that a user's information need (I) is satisfied given a particular document (D). This probability can be computed separately for each document in a collection. The documents can then be ranked by probability, from highest probability of satisfying the user's need to lowest [4].

- **Bayesian Belief Network Model**

Bayesian belief network is the use of Bayesian calculus to determine the probabilities of each node from the predetermined conditional and prior probabilities [37]. As Baeza-Yates and Ribeiro-Neto [4], stated in Bayesian belief network the users query q is modeled as a network node to its associated random variable. Whenever q completely covers the concept space C the variable is set to 1. Therefore belief network computes the probability of q, (i.e. P (q)) by the degree of coverage of the space C by q. Document dj is modeled as network node to its associated binary random variable. If dj completely covers the concept space C the variable set to 1. To compute the probability of dj (P (dj)), compute the degree of coverage of the space C by dj.

In belief network documents and the user query modeled as subsets of index terms. Each subset is interpreted as concept in the concept space C [4]. The ranking principle in belief network expressed as, the degree of coverage provided to the concept dj by the concept q. P (dj|q) is adopted as the rank of the document dj with respect to the query q [4].

In general, probabilistic models attempt to capture the information retrieval problem within a probabilistic framework. Unfortunately, the probabilistic model has got its own drawbacks. First, the probability theory analysis takes much more time and efforts, and it offer unnecessary theoretical burden on the researcher. Second, probabilistic model need to guess the initial separation of documents into relevant and non-relevant sets. Third, the model does not take into account the frequency with which an index term occurs inside a document. In other word, all weighs are binary [4] [27].

However, probabilistic model have several potential advantages [4] [27]. The first, advantage is the expectation of retrieval effectiveness that is near to optimal relative to the evidence used is high. Second, it has less reliance on traditional trial and error retrieval experiments. Third, each document's probability of relevance estimate can be reported to the user in ranked output. It would presumably be easier for most users to understand and base their stopping behavior (i.e., when they stop looking at lower ranking documents).

2.3. Query operation

Without detailed knowledge of the collection make-up and of the retrieval environment, most users find it difficult to formulate queries which are well designed for retrieval purposes. In fact, as observed with Web search engines, the users might need to spend large amounts of time reformulating their queries to accomplish effective retrieval. This difficulty suggests that the first query formulation should be treated as an initial (naive) attempt to retrieve relevant information. Following that, the documents initially retrieved could be examined for relevance and new improved query formulations could then be constructed in the hope of retrieving additional useful documents. Such query reformulation involves two basic steps: expanding the original query with new terms and reweighting the terms in the expanded query [modern]. In this research, methods involving query expansion and term re-weighting are investigated.

Relevance feedback is the most popular strategy used to select expanding terms for query reformulation purpose [4]. A process involving, the use of relevance feedback and query reformulation is called query operation. The two basic approaches to provide relevance feedback are user's relevance feedback and pseudo-relevance feedback [4].

2.4. User's Relevance Feedback

Relevance feedback is the most popular query reformulation strategy. In a relevance feedback cycle, the user is presented with a list of the retrieved documents and, after examining them, marks those which are relevant. In practice, only the top 10 (or 20) ranked documents need to be examined. The main idea consists of selecting important terms, or expressions, attached to the documents that have been identified as relevant by the user, and of enhancing the importance of these terms in a new query formulation. The expected effect is that the new query will be moved towards the relevant documents and away from the non-relevant ones.

User relevance feedback is the process of involving users in the retrieving process. A system that uses this approach needs judgment from users, So that a query with better meaning expression power can be formulated. User relevance feedback involves a series of steps. First the user issues a query on which the system returns an initial set of documents. Among the retrieved documents, the user marks some of them as relevant. The systems then computes a better representation of the query, based on user's relevance judgment. Finally the system uses the reformulated query to retrieve the revised set of documents Manning et al [26].

2.5. Query Expansion and Term Reweighting for the Vector Model

The application of *relevance feedback* to the vector model considers that the term weight vectors of the documents identified as relevant (to a given query) have similarities among themselves (i.e., relevant documents resemble each other). Further, it is assumed that non-relevant documents have term-weight vectors which are dissimilar from the ones for the relevant documents. The basic idea is to reformulate the query such that it gets closer to the term-weight vector space of the relevant documents [4].

Let us define some additional terminology regarding the processing of a given query q as follows,

D_r: set of relevant documents, as identified by the user, among the *retrieved* documents;

D_n: set of non-relevant documents among the *retrieved* documents;

C_r: set of relevant documents among all documents in the collection;

$|D_r|$, $|D_n|$, $|G_r|$: number of documents in the sets D_r, D_n and G_r, respectively; α, β, r tuning constants.

Consider first the unrealistic situation in which the complete set Gr of relevant documents to a given query q is known in advance. In such a situation, it can be demonstrated that the best query vector for distinguishing the relevant documents from the non-relevant documents is given by,

$$\vec{q}_{opt} = \frac{1}{|C_r|} \sum_{\forall \vec{d}_j \in C_r} \vec{d}_j - \frac{1}{N - |C_r|} \sum_{\forall \vec{d}_j \notin C_r} \vec{d}_j \qquad \text{..... (2.1)}$$

The problem with this formulation is that the *relevant* documents which compose the set O_r are not known a priori. In fact, we are looking for them. The natural way to avoid this problem is to formulate an initial query and to incrementally change the initial query vector [4]. This incremental change is accomplished by restricting the computation to the documents *known* to be relevant (according to the user judgment) at that point. There are three classic and similar ways to calculate the modified query $\vec{q}_m$ as follows,

Standard_Rochio: $$\vec{q}_m = \alpha \vec{q} + \frac{\beta}{|D_r|} \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \frac{\gamma}{|D_n|} \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j \qquad \text{.....(2.2)}$$

α: Tunable weight for initial query.

β: Tunable weight for relevant documents.

γ: Tunable weight for irrelevant documents.

I de-Regular: .....(2.3)

$$\vec{q}_m = \alpha \vec{q} + \beta \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \sum_{\forall \vec{d}_j \in D_n} \vec{d}_j$$

α: Tunable weight for initial query.

β: Tunable weight for relevant documents.

γ: Tunable weight for irrelevant documents.

I de.Dec.Hi:

$$\vec{q}_m = \alpha \vec{q} + \beta \sum_{\forall \vec{d}_j \in D_r} \vec{d}_j - \gamma \max_{non-relevant} (\vec{d}_j) \quad \text{.....(2.4)}$$

α: Tunable weight for initial query.

β: Tunable weight for relevant documents.

γ: Tunable weight for irrelevant documents.

The Rochio formulation is basically a direct adaptation of equation 2.1in which the terms of the original query are added in. The motivation is that in practice the original query q may contain important information. Usually, the information contained in the relevant documents is more important than the information provided by the non-relevant documents. This suggests making the constant, γ smaller than the constant β (3. An alternative approach is to set, γ to 0 which yields a *positive* feedback strategy [4].

The main advantages of the above relevance feedback techniques are simplicity and good results. The simplicity is due to the fact that the modified term weights are computed directly from the set of retrieved documents. The good results are observed experimentally and are due to the fact that the modified query vector does reflect a portion of the intended query semantics. The main disadvantage is that *no* optimality criterion is adopted: [4]

2.6. Term Reweighting for the Probabilistic Model

The probabilistic model dynamically ranks documents similar to a query q according to the probabilistic ranking principle [4].

$$sim\ (d_j, q) \propto \sum w_{i,q} w_{i,j} \left(\log \frac{P(k_i \mid R)}{1 - P(k_i \mid R)} + \log \frac{P(k_i \mid \overline{R})}{1 - P(k_i \mid \overline{R})} \right) \quad \text{...(2.5)}$$

where $P(k_i|R)$ stands for the probability of observing the term k_i in the set R of relevant documents and $P(k_i|R)$ stands for the probability of observing the term k_i in the set R of non-

relevant documents. Initially, equation 2.5 cannot be used because the probabilities $P(k_i|R)$ and $P(k_i|R)$ are unknown. A number of different methods for estimating these probabilities automatically (i.e., without feedback from the user)

For the initial search (when there are no retrieved documents yet), assumptions often made include: (a) $P(k_i|R)$ is constant for all terms k_i (typically 0.5) and (b) the term probability distribution $P(k_i|R)$ can be approximated by the distribution in the whole collection. These two assumptions yield:

$$P(k_i|R) = 0.5$$

$$P(k_i|R) = n_i/N$$

Where, as before, *ni* stands for the number of documents in the collection which contain the term~. Substituting into equation 5.2, we obtain

$$sim(d_j, q) = \sum_{i=1}^{t} w_{i,q} w_{i,j} \left(\log \frac{N - n_i}{n_i} \right) \quad \ldots\ldots\ldots (2.6)$$

For the feedback searches, the accumulated statistics related to the relevance or non-relevance of previously retrieved documents are used to evaluate the probabilities $P(ki|R)$ and $P(ki|R)$. As before, let $\overrightarrow{Dr}$ be the set of relevant retrieved documents (according to the user judgment) and $D_r i$ be the subset of Dr composed of the documents which contain the term k_i . Then, the probabilities $P(ki|R)$ and $P(ki|R)$ can be approximated by

$$P(k_i \mid \overline{R}) = \frac{n_i - |\overrightarrow{D_{r,i}}|}{N - |D_r|} \quad ; \quad P(k_i \mid R) = \frac{|D_{r,i}|}{|D_r|} \quad \ldots\ldots (2.7)$$

Using these approximations, equation 5.2 can rewritten as

$$sim(d_j, q) = \sum_{i=1}^{t} w_{i,q} w_{i,j} \log \left(\frac{|D_{r,i}|}{|D_r| - |D_{r,i}|} \Big/ \frac{n_i - |D_{r,i}|}{N - |D_r| - (n_i - |D_{r,i}|)} \right) \quad \ldots\ldots (2.8)$$

Notice that here; contrary to the procedure in the vector space model, no query expansion occurs. The same query terms are being reweighted using feedback information provided by the user.

The main advantages of this relevance feedback procedure are that the feedback process is directly related to the derivation of new weights for *query* terms and that the term reweighting is optimal under the assumptions of term independence and binary document indexing ($W_i q$ £ {O, 1} and $W_i j$ £ {O, 1}) [4].

The disadvantages include:

(1) Document term weights are *not* taken into account during the feedback loop;

(2) Weights of terms in the previous query formulations are also disregarded; and

(3) No query expansion is used (the same set of index terms in the original query is reweighted over and over again).

2.7. Pseudo-relevance feedback

The previous discussed method of reformulating queries involves users. User's relevance judgment is necessity to come up with words which have the potential to increase query quality. Thought, user's relevance feedback tends to reformulate user's queries as per their judgment, it can also be boring & time taking for them, which in turn degrades their interests of further searching. As a solution to this problem pseudo-relevance feedback method was introduced, which doesn't need user's involvement? As IR system which uses pseudo-relevance feedback method automatically generates expanding terms. There exist two approaches that utilize pseudo-relevance feedback method called Local automatic analysis and global automatic analysis [4]. For relevance feedback, the former utilizes documents that are initially retrieved using users original query and the latter analyzes the whole document corpus [4] In both cases the process is completely automatic, such that users have no clue, whether their first query have been reformulated or not.

2.7.1. Local automatic analysis

Local automatic analysis is a method that uses the document which is considered as relevant by the system in the first retrieval attempt, to formulate the original query. This means, the documents retrieved for a given query q are examined at query time to determine terms for query expansion [4]. There are two kinds of local automatic analysis techniques; query expansion based clustering and query expansion based on both local and global analysis [4].

2.7.1.1. Query Expansion through Local Clustering

Adoption of clustering techniques for query expansion is a basic approach which has been attempted since the early years of information retrieval. The standard approach is to build global structures such as association matrices which quantify term correlations (for instance, number of documents in which two given terms co-occur) and to use correlated terms for query expansion. The main problem with this strategy is that there is not consistent evidence that global structures can be used effectively to improve retrieval performance with general collections. One main reason seems to be that global structures do not adapt well to the local context defined by the current query. One approach to deal with this effect is to devise strategies which aim at optimizing the current search. Such strategies are based on *local clustering*. Some of these techniques are discussed below [4].

✓ **Association clustering**

This method tries to associate terms which co-occur in various documents. An association cluster is based on the co-occurrence of stems (or terms) inside documents. The idea is that stems which co-occur frequently inside documents have a synonymity association [4].

- Definition:
  - fsi,j : the frequency of a stem s_i in a document d_j ,
  - Let $m=(m_{ij})$ be an association matrix with $|S_l|$ row and $|D_l|$ columns, where $m_{ij}=f_{si,j}$.
  - The matrix s=mm is a **local stem-stem association matrix**.

Each element $s_{u,v}$ in s expresses a correlation $c_{u,v}$ between the stems s_u and s_v:

$$c_{u,v} = \sum_{d_j \in D_l} f_{s_u,j} \times f_{s_v,j} \quad \text{.....(2.9)}$$

The correlation factor Cu,v quantifies the absolute frequencies of co-occurrence and is said to be unnormalized. Thus, if we adopt $s_{u,v} = c_{u,v}$

Then the association matrix s is said to be unnormalized. An alternative is to normalize the correlation factor. For instance, if we adopt

$$s_{u,v} = \frac{c_{u,v}}{c_{u,u} + c_{v,v} - c_{u,v}} \qquad \text{.....(2.10)}$$

The association clustering method them takes the q^{th}row holding the co-occurrence information of term q, from the normalized matrix $S_{u,v}$ returns n number of terms with the highest $S_{q,v}$ values. This in turn makes a cluster of n new terms have good co-occurrence with the term q.

- ✓ **Metric clustering**

Association clusters are based on the frequency of co-occurrence of pairs of terms in documents and do not take into account *where* the terms occur in a document. Since two terms which occur in the same sentence seem more correlated than two terms which occur far apart in a document, it might be worthwhile to factor in the distance between two terms in the computation of their correlation factor. Metric clusters are based on this idea [4].

Let the distance *r(ki, kj)* between two keywords k_i and k_j in a same document. If k_i and k_j are in distinct documents we take $r(ki, kj) = \infty$

A local stem-stem metric correlation matrix s is defined as:

Each element $s_{u,v}$ of s expresses a metric correlation $c_{u,v}$ between the set m_s s_u, and s_v

$$c_{u,v} = \sum_{k_i \in V(s_u)} \sum_{k_j \in V(s_v)} \frac{1}{r(k_i, k_j)} \qquad \text{.....(2.11)}$$

In this expression, as already defined, *V(su)* and *V(sv)* indicate the sets of keywords which have *Su* and *Sv* as their respective stems. Variations of the above expression for *Cu,v* have been reported in the literature (such as *1/r(ki,kj)* but the differences in experimental results are not remarkable [4].

The correlation factor *Cu,v* quantifies absolute inverse distances and is said to be unnormalized. Thus, if we adopt

$Su,v = Cu,v$

Then the association matrix s is said to be unnormalized. An alternative is to normalize the correlation factor. For instance, if we adopt

$$s(k_u, k_v) = \frac{c(k_u, k_v)}{|V(k_u)| \times |V(k_v)|} \quad \text{.....(2.12)}$$

At last from the normalized matrix $S_{u,v}$, those having the highest value with qth raw are selected for expansion purpose.

- ✓ **Scalar clustering**

One additional form of deriving a synonymity relationship between two local stems (or terms) *Su* and *Sv* is by comparing the sets *Su(n)* and *Sv(n)*. The idea is that two stems with similar *neighborhoods* have some synonymity relationship. In this case we say that the relationship is indirect or induced by the neighborhood. One way of quantifying such neighborhood relationships is to arrange all correlation values *Su,i* in a vector Su, to arrange all correlation values *Sv,i* in another vector *sv,* and to compare these vectors through a scalar measure. For instance, the cosine of the angle between the two vectors is a popular scalar similarity measure [4].

$$S_{u,v} = \frac{\vec{S_u} \bullet \vec{S_v}}{|\vec{S_u}| \times |\vec{S_v}|} \quad \text{.....(2.13)}$$

The correlation matrix *s*is said to be induced by the neighborhood.

2.7.2. Global automatic analysis

The methods of automatic local analysis extract information from the local set of documents retrieved to expand the query. It is well accepted that such a procedure yields improved retrieval performance with various collections. An alternative approach is to expand the query using information from the whole set of documents in the collection. Strategies based on this idea are called global analysis procedures. Until the beginning of the 1990s, global analysis was

considered to be a technique which failed to yield consistent improvements in retrieval performance with general collections. This perception has changed with the appearance of modern procedures for global analysis [4]. Two modern procedures for global analysis are query expansion based on similarity thesaurus and query expansion based on statistical thesaurus.

2.7.2.1. Query Expansion based on a Similarity Thesaurus

The similarity thesaurus is based on term to term relationships rather than on a matrix of co-occurrence. A *similarity thesaurus* is built considering term to term relationships. However, such relationships are not derived directly from co-occurrence of terms inside documents. Rather, they are obtained by considering that the terms are concepts in a concept space. In this concept space, each term is indexed by the documents in which it appears. Thus, terms assume the original role of documents while documents are interpreted as indexing elements [4]. Therefore an individual terms k_i is seen as a vector and is defined as $\vec{k}_i = (w_{i,1}, w_{i,2}, \ldots, w_{i,N})$. Where wi,j is a weight associated to index-document pair [ki,dj]. These weights are computed as follows [4].

$$w_{i,j} = \frac{(0.5+0.5\times\frac{f_{i,j}}{\max_j(f_{i,j})})\times itf_j}{\sqrt{\sum_{l=1}^{N}(0.5+0.5\times\frac{f_{i,l}}{\max_l(f_{i,l})})^2 itf_j^2}} \quad \ldots\ldots\ldots (2.14)$$

Where *maX j(fi,j)* computes the maximum of all factors fi,j for the i^{th} term (i.e., over all documents in the collection). We notice that the expression above is a variant of tf-idf weights but one which considers inverse term frequencies instead. The relationship between two terms *ku* and *kv* is computed as a correlation factor *Cu,v* given by

$$c_{u,v} = \overrightarrow{k_u} \cdot \overrightarrow{k_v} = \sum_{\forall d_j} w_{u,j} \times w_{v,j} \quad \ldots\ldots\ldots\ldots (2.15)$$

The global similarity thesaurus is built using this correlation factor *Cu,v.* after the thesaurus is built a system expands a query with concepts similar to the whole of the query q, in this steps [4].

First the query q is associated a vector q in the term-concept space given by

$$\vec{q} = \sum_{k_i \in q} w_{i,q} \vec{k_i} \quad \text{..... (2.16)}$$

Where $w_{i,q}$ is a weight associated to the index-query pair$[k_i,q]$

Second Compute a similarity $sim(q,k_v)$ between each term k_v and the user query q

$$sim(q, k_v) = \vec{q} \cdot \vec{k_v} = \sum_{k_u \in q} w_{u,q} \times c_{u,v} \quad \text{..... (2.17)}$$

Where $c_{u,v}$ is the correlation factor

Finally, Add the top r ranked terms according to $sim(q,k_v)$ to the original query q to form the expanded query q'. To each expansion term k_v in the query q' is assigned a weight $w_{v,q'}$ given by

$$w_{v,q'} = \frac{sim(q, k_v)}{\sum_{k_u \in q} w_{u,q}} \quad \text{..... (2.18)}$$

The expanded query q' is then used to retrieve new documents to the user

2.7.2.2. Query expansion based on statistical thesaurus

The global thesaurus is composed of classes which group correlated terms in the context of the whole collection. Such correlated terms can then be used to expand the original user query. To be effective, the terms selected for expansion must have a high term discrimination value which implies that they must be low frequency terms. However, it is difficult to cluster low frequency terms effectively due to the small amount of information about them (they occur in few documents). To circumvent this problem, we cluster documents into classes instead and use the low frequency terms in these documents to define our thesaurus classes. In this situation, the document clustering algorithm must produce small and tight clusters [4].

A document clustering algorithm which produces small and tight clusters is the *complete link algorithm* which works as follows (naive formulation).

(1) Initially, place each document in a distinct cluster.

(2) Compute the similarity between all pairs of clusters.

(3) Determine the pair of clusters [Cu, *Cvl* with the highest inter-cluster similarity.

(20) Merge the clusters Cu and *Cv'*

(5) Verify a stop criterion. If this criterion is not met then go back to step 2.

(6) Return a hierarchy of clusters.

The similarity between two clusters is defined as the minimum of the similarities between all pairs of inter-cluster documents (I.e., two documents not in the same cluster). To compute the similarity between documents in a pair, the cosine formula of the vector model is used. As a result of this minimality criterion, the resultant clusters tend to be small and tight [4].

After preparing the document cluster, three parameters are taken from users to select the most appropriate document clusters from which the thesaurus class is generated. The first parameter is threshold class (TC). The second, number of documents in a class (NDC), and the third is the minimum inverse document frequency (MIDF). TC is determined by setting the minimum amount of similarity required between two clusters, which are about to be merged. Therefore, for two clusters to be merged, the similarity between the two surpasses a given TC. NDC is the minimum number of document that should be found in a cluster. If there are two clusters $C_{a,b,c}$ and $C_{a,b,}$ and if the NDC factor is small, the system prefers the cluster $C_{a,b,}$ and if it is a big number then $C_{a.b.c}$ is selected. The last factor MIDF is used to select words from each cluster that are used to create the threshold class. An average term weight wt_c for each thesaurus class C is computed as follows [4].

$$wtc = \frac{\sum_{i=1}^{|C|} w_{i,C}}{|C|} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.19)$$

Where |C| is the number of terms in the thesaurus class C, and *Wi,c* is a pre-computed weight associated with the term-class pair *[ki , C].* This average term weight can then be used to compute a thesaurus class weight *Wc, as*;

$$Wc = \frac{wtc}{|C|} \times 0.5 \quad \text{..... (2.20)}$$

The above weight formulations have been verified through experimentation and have yielded good results [4]. However, the main problem is estimating parameters of TC, NDC and MIDF. This is because, different values of these parameters yield various results, which most of them might not be effective [4].

2.8. IR System Evaluation

Retrieving relevant document from the collection that satisfies users need is the heart of IR system evaluation in determining its effectiveness. Two strategies are identified in measuring the effectiveness performance of IR systems. The first is the user-centered strategy, which uses relevant judgment so as to evaluate the performance of the system and the second is system centered strategies which work based on reference judgment available prior to testing process [4]. Based on the concept of relevance (i.e. to a given query or information need), there are several techniques of measures of IR performance available, such as, precision and recall, F-measure, E-measure, MAP (Mean average precision), R-measure, etc [28]. In this study, the three widely used techniques namely; precision, recall, and F-measure are used to measure the effectiveness of the IR system designed.

Precision and recall are the two most frequent and basic statistical measures. Recall is the percentage of relevant documents retrieved from the database in response to users query, whereas precision is percentage of retrieved documents that are relevant to the query [1]. To show these metrics, assume the document collection be D. Let R_t is all retrieve documents from the collection D and R_i a number of relevant documents in D. The joint of R_t and R_i is a set of documents retrieved and relevant, RA. Therefore, the recall and precision can be calculated using equation 2.21 and 2.22 respectively.

$$\text{Recall} = RA \mid R_t \mid\mid R_i \mid \text{..... (2.21)}$$

$$\text{Precision} = RA \mid R_t \mid\mid R_t \mid \text{..... (2.22)}$$

The formula show above for precision and recall assume that, all documents retrieved (Rt) is examined by user. Thus, the retrieved document cannot be presented for the user at once. Rather, the retrieved documents are presented according to their degree of relevance as per the user query. Then, the user examines the ranked documents starting from the top. This kind of examination of documents by the user leads the recall and precision measures to vary. Therefore, for appropriate evaluation of recall and precision, plotting a precision versus recall curve is necessary [4].

To draw precision-recall curve, for example let documents retrieved by the system is Dr Where Dr = {d3, d33, d9, d1, d10, d11, d14, d7, d44, d49, d50, d53, d55, d77, d133} in ranked order. Assume Rq contain a set of relevant documents for the query. Where, Rq = {d1, d3, d7, d10, d14, d33, d44, d49, d55, d133}. Based on the given information recall- precession curve is constructed. However, when constricting the curve based on the original recall and precession may result in saw tooth curve, there is a need to smooth the curve using interpolation technique [4].

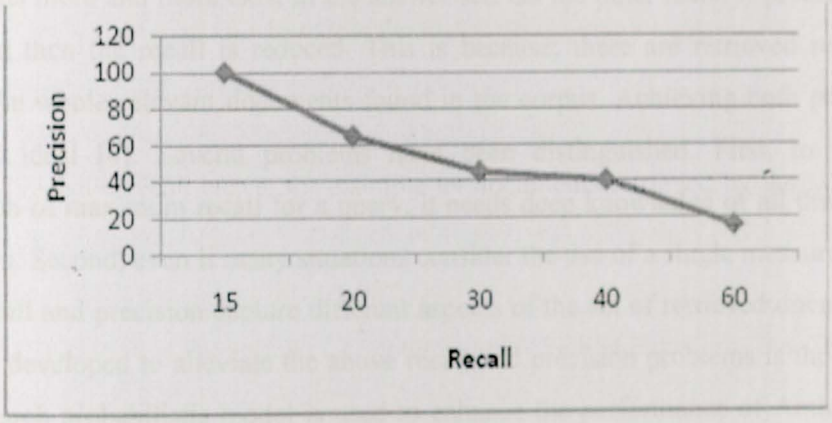


Figure 2.1: Example of precision and recall graph

The example shown above for plotting precision and recall is carried out for a single query. When measuring the performance of IR system multiple queries are used. To evaluate the performance of the system using k queries, average precision at each recall level is used. This can be expressed as follows [4];

$$\bar{P}(r) = \sum_{i=1}^{Nq} Pi(r)\ Nq,\ Nqi=1 \quad \ldots\ldots\ldots\ldots\ldots (2.23)$$

Where, $\bar{P}(r)$ is the average precision at the recall level r, Nq is the number of queries used, and Pi(r) is the precision at recall level r for the i^{th} query. It is an empirical fact that on average as recall increases, precision decreases. The recall levels for each query might be deferent from the standard recall levels, which is difficult to compare performance across queries. Therefore, interpolation procedure is necessary. As a result, a single precision value for each query can be used that takes a precision at some recall level for each single query. The interpolated precision versus recall curve is shown as follows; Let rj, j E {0, 1,2, ... , 10}, be a reference to the j^{th} standard recall level. Then, P(rj) = max rj ≤ r≤rj+1 P(r), "which states that the interpolated precision at the jth standard recall level is the maximum known precision at any recall level between the jth recall level and the $(j + l)^{th}$ recall level"[4].

In general, precision and recall have been used widely so as to evaluate information retrieval system performance. However, they measure two different aspects of the system and thus they are inversely relative. If recall of a system is improved then the precision is reduced. The reason behind this is that, when attempt is made to include many of the relevant documents, irrelevant documents more and more exist in the answer set. On the other hand, if precision of a system is improved then the recall is reduced. This is because; there are retrieved relevant documents among the whole relevant documents found in the corpus. Achieving both precision and recall 100% is ideal [4]. Several problems have been distinguished. First, to make appropriate estimation of maximum recall for a query, it needs deep knowledge of all the documents in the collection. Second, even if many situations consider the use of a single measure, which combines both, recall and precision capture different aspects of the set of retrieved documents. One of the methods developed to alleviate the above recall and precision problems is the F-measure [4]. In this research probabilistic model is used to enhance the performance of Amharic IR system by increasing the precision without affecting the recall ability of the system. Thus, the F-measure is used to measure the performance of the system since it balances the precision and recall values. F-measure is a single measure that trades off precision versus recall. It is the weighted harmonic mean of precision and recall expressed as follows [1];

$$F(j) = 2.P.RP+R \quad \ldots\ldots\ldots\ldots (2.24)$$

Where R= (r1, r2,....r11,...rn) and P=(p1,p2,....p11,...pn) are recall and precision at the j^{th} document respectively. F-measure will be 0 when no relevant document is retrieved and 1 when

all the first ranked documents are relevant. Moreover, the harmonic mean F assumes a high value only when both recall and precision are high.

In summary, there are different methods in designing probabilistic based IR system. However, the binary independent method is used to develop probabilistic based IR system for Amharic language so as to enhance the performance of the IR and to alleviate the problem of uncertainty exists in IR. To evaluate the performance of the method, the model is implemented and tested using Amharic news articles.

2.9. The Amharic Writing system and its Features

Ethiopia has several languages, which are spoken by different nations and nationalities. Among the languages, Amharic or "አማርኛ" is the national language of Ethiopia until 1995. Following the declaration of constitution of Ethiopian federal democratic government, it becomes the working/official language. Because of its wide application, Amharic documents exist in both hard copy and electronic forms. Since this research is conducted considering Amharic documents, it is important to investigate the potential features of the language that have the capability of representing the contents of the documents that demands one to understand the characteristics of the language.

2.9.1. History of Amharic Writing System

The origin of Amharic writing system traced back when Semitic scripts were flourished in the Middle East before three thousand years ago [38], The North Semitic and South Semitic are branched from Proto-West Semitic. The children in North Semitic branch are Hebrew, Arabic, Greek, Roman and Cyrillic. The South Semitic side produces Ethiopic via the Sabean system, which is speculatively dated as emerging in the 11th and 10th centuries BC. The three Semitic languages which are only found in Ethiopia and Eritrea are Geez or Ge'ez, Amharic and Tigrinya which are used in a representation for Ethiopic system [38].

Geez play a significant role in the development and expansion of Amharic language and writing system. Several religious texts, such as Bible, translations of Arabic Christian texts from Egypt and literatures such as the "qine" (ቅኔ) and poems are all written in Geez. The emergence and expansion of Geez inscriptions in the Ethiopic script traced back to the 20th century AD, when

Geez was the language of the empire of Aksum North Ethiopia. Even if the use of geez is limited to Orthodox Church, it is still a source for the coining of Ethiopian literary and technical terms [38].

The Sabean script is attributed as the source of the Geez script, likewise, the Geez writing system attributed as the source of Amharic writing system. In the early age, Amharic has been the language of the politically dominant ethnic group in Ethiopia and it is now used in every part of the country.

The Amharic language alphabet consists of 33 basic characters having 7 different forms for each consonant vowel combination [38]. The basic character occurs in one form and the remaining six different forms are building based on the basic characters having a little modification form. For example, the seven orders of the consonants of ሀ (hä) and ዘ (ze) expressed as follows;

ሀ	ሁ	ሂ	ሃ	ሄ	ህ	ሆ
Hä	hu	hi	hä	he	h	ho
ዘ	ዙ	ዚ	ዛ	ዜ	ዝ	ዞ
Zä	zu	zi	zä	ze	z	zo

A consonantal system represents symbols separately as consonants and vowels, while the syllabary writing system has individual signs for syllables (i.e. consonant +vowel combination). There is a dispute as to whether or not the Amharic writing system is a syllabary [38]. As further noted in [38] scholars like bender (1968) and bender, et al.(1976) referred it as a syllabary, while Sampson (1985) explicitly rules it out of the syllabary category with the argument that the Amharic writing system has related symbols for phonologically similar syllables. Thus it could be argued that Ethiopic to some extent resembles other Semitic scripts such as the Arabic and Hebrew consonantal systems, which basically indicate consonants but, for teaching purposes, etc, have developed optional diacritics to signify vowels[38].

Unlike the Arabic and Hebrew, the Amharic writing system is written from left to right. As compared to English or Latin language writing system, the Amharic writing system has no upper

and lower case latter variations and has no conventional cursive (i.e. written in a connected letters)form [38].

When Ge'ez become the spoken and written language in common use in northern Ethiopia, it took only 24 of the 29 sabaean symbols, modify most of them and add two new symbols to represent sounds of Greek and Latin loanwords not found in Ge'ez [12]. The two extra symbols are added to cope with two new sounds (/p/ as in "ፐ" and /p'/as in "ጰ", originally required for use in ecclesiastical Greek and Latin borrowing and names (e.g. poulos, police) [39]. Amharic inherited symbols from Ge'ez. It took all of the symbols and added eight new ones ("ቨ", "ጨ", "ኘ", "ኸ", "ዠ" "ሸ" "ኀ", and "ገ") that represent sounds not found in Ge'ez [12].

Considering all the 34 consonant symbols, there is considerable regularity of letter shapes, but some orders are more regular than others. The shapes are most consistent in the 5th order (e.g. ሄ፣ ሌ፣ ሜ፣ ሴ፣ ሬ having a ring on their right legs except ጨ), slightly less in the 3rd (e.g. ሂ፣ ሊ፣ ሚ፣ ሲ፣ ቢ having a small hyphen like extension on the bottom of their right legs, except ሪ and ዪ), slightly less again in the 2nd (e.g. ሁ፡ ሉ፡ ሡ፡ ቡ፡ having a small hyphen like extension on their right legs except ሩ ፉ፡ and ጡ), still less in the 20th (e.g. ላ፣ ሳ፣ ሻ፣ ባ፣ ካ፣ ዛ፣ ዘ፣ ዳ፣ ጃ፣ ጳ have their left leg shortened, on the other hand ጣ ጫ has there two left legs shortened. Others like ቃ ታ ተ ጋ ፓ have bow like legs) and even less in the 7th (e.g. ሶ ሾ ቦ ኮ ዞ ገ ዶ have their right legs shortened, on the other hand ፎ and ጦ have their two right legs shortened, others like ሆ, ሎ, ኖ, ሾ, ኖ, ሮ have a ring somewhere on their base letter or have some kind of modification like ሞ and ሃ) and the 6th (e.g. ል፣ ም፣ ስ፣ ድ፣ ቅ፣ ት፣ ጥ፣ ፍ having completely distinctive & pattern less structure] order, with the greatest number of patterns, [38]. Therefore the system is composed of largely unpredictable patterns. For example the set of syllables with /g/ starts off regularly enough except the 20th 'ጋ', 6th 'ግ' and 7th 'ጎ' orders. The /w/ set is even ambiguous having the 2nd 'ወ' read as 'wu' and 6th 'ዉ' read as 'wi' symbols highly alike. The system has regularity in writing which makes it easy for a person to learn the language. For example the symbol ‹ሰ› read as , sə, and ‹ሸ› read as ,she, have relatively the same kind of structure as their accent. Therefore, a person experiences any difficulty to learn a symbol, given that he/she knows the other (for some of the symbols). The Amharic writing system has 34 basic characters and their seven orders given 238 distinct

symbols. In addition, there are forty others that contain a special feature usually representing labialization e.g. ሳ፣ ቿ፣ ሟ፣ ሯ፣ ቈ [12].

- ✓ Two dots (":") called "ሁለት ነጥብ", which is similar to a colon used to separate words, however, nowadays the influence of foreign languages lead to the use of blank space instead.
- ✓ Four dots ("::") called "አራት ነጥብ", which is used as full stop.
- ✓ Two dots with horizontal stroke line over them ("፣") called "ነጠላ ሰረዝ" which is used as comma.
- ✓ Two dots with horizontal stroke above and below them ("፤") called "ድርብ ሰረዝ", which is used as semi-colon. The other punctuations are mostly borrowed symbols such as? , ! , /.\, ", "etc.

Every language has its own way of representing numbers, despite the absence of representing number zero ("0"), Amharic has also its own way of representing numbers. The Amharic numbering system consists of twenty (20) single characters derived from Greek letters by modifying the characters into Amharic character form. The symbols are modified by adding a horizontal stroke above and below. The numbers consists of a single character for one to ten, for multiples of ten (twenty to ninety), hundred and thousand [39]. The Amharic writing system also consists around 17 (seventy) punctuation marks. The most commonly used punctuation marks are [39];

2.9.2. Ambiguities in Amharic Writing System

Amharic writing system adopted all the symbols in Ge'ez and added 8 other symbols and the other 44 symbols. The result is that there is a considerable systemic redundancy of several consonant sounds which lacks in the phonology of Ge'ez [38]. Ambiguities in Amharic Writing system arise mainly due to symbol redundancy [12]. Thus, 4 distinct sets of 7 can represent the sound /h/ + vowel: (ሀ፣ ሐ፣ ኀ ፣ኸ) 2 sets represent /s/ (ሰ፣ ሠ) and 2 /s'/ (ጸ፣ ፀ) [8]. A similar problem is observed in usage of some letters interchangeably, such as ''ቆ'' VS ''ቈ'', ''ኩ'' VS ''ኵ'', and ''ጎ'' vs. ''ጐ'' [12]. In addition to the symbolic redundancy of characters, Amharic writing system suffers slightly from visual similarity of different character, such as ፕ and ኝ, ፖ and ፓ, ፍ and ፎ, ጮ and ጭ [12].

Canonical Amharic	Common Amharic	Possible but Impossible Amharic
ዓለም	አለም	ዐለም፣ ኣለም
ፀሐይ	ጸሃይ፣ፀሃይ፣ጸሐይ	ጸሀይ፣ጼሓይ፣ጸኀይ፣ጸኻይ፣ፀሀይ፣ፀሓይ፣ፀኀይ፣ፀኃይ፣ፀኻይ
ኃይለ ሥላሴ	ኀይለ ሥላሴ ፣ ኃይለ ስላሴ፣ኀይለ ስላሴ፣ኀይለ ስላሤ፣ስላሴ፣ኃይለ ስላሤ፣ሀይለ ሥላሴ፣ሀይለ ሥላሤ፣ሃይለ ሥላሴ፣ሃይለ ስላሴ፣ሃይለ ሥላሤ፣ሥላሤ፣ኃይለ ሥላሴ፣ኃይለ ስላሴ፣ኃይለ ሥላሤ፣ሀይለ ሥላሴ፣ሀይለ ስላሴ፣ሀይለ ሥላሤ፣ሐይለ ሥላሴ፣ሐይለ ስላሴ፣ሐይለ ሥላሤ፣	ሐይለ ሥላሤ፣ሐይለ ሥላሤ፣ሐይለ ሥላሴ፣ሐይለ ስላሴ፣ሐይለ ስላሤ፣ኻይለ ሥላሤ፣ኻይለ ሥላሤ፣ኻይለ ሥላሴ፣ኻይለ ስላሴ

Table 2.2 Spelling variants of the same word adopted from Alemayehu (2002)

Because of those symbols having the same accent in speaking, they can be used interchangeable in the various words of Amharic language, thus, forming different forms of spelling for the same word. The different forms of spelling variants of the same word are shown in table 2.1 above.

2.9.3. Challenges in Designing an Amharic IR System

Because of the different kinds of ambiguities in Amharic writing system, designing An IR system based on Amharic text is challenging. The class of symbols with the same sound falls into two [12]. The first class consists of symbols in the 1st order and the 20th order in a base symbols having the same sound during reading: for example, ሀ and ሃ፣ ሐ፣ ሓ፣ ኀ and ኃ፣ ኣ ፣ ዓ ፣አ and ዐ

The other class is a group of different alphabets that share the same sound, which are ሀ፣ሐ and ኀ፣ሰ and ሠ፣አ and ዓ and ጸ and ፀ. This makes words with the same meaning to have different spelling structure. For instance, the same word 'tsehay' 'ፀሐይ' can be written differently as ጸሃይ፣ፀሃይ፣ጸሐይ etc [12]. Using these symbols interchangeably in words doesn't make reading, or forwarding ideas difficult for human beings. But unlike humans it is

difficult for systems to consider them having the same meaning. Because IR systems only match the symbols in words to check weather a word from a document has the same meaning as in the query (i.e. if the words are a match then they are the same and have same meaning) encountering the different interchangeable symbols in words forces it to consider them as different (i.e. the system considers ጸሀይ and ፀሃይ as different words with different meaning).

The other challenge Amharic IR systems face is compound words. There is no convention as to which words should be combined as one word or separately during writing. For example, the word 'metsedaja bet' which means ''toilet room'' can possibly be written as መፀዳጃቤት (without space) and መፀዳጃ ቤት (with space) and also the word 'bete mengist' which means ''palace'' can be written as (without space) ቤተመንግስት and ቤተ መንግስት (with space) which makes it difficult for the IR system to differentiate between the two compound words [12].

A word with two or more different meanings is known as polysemic words. There are specific combination of words in Amharic language which possess different meaning are called "ፊሊጥ. In Amharic language, polysemic words can be created by the combination of two words, which gives it a different meaning than the obvious meaning of the word combination [40]. For example is a combination 'ሆዱ ሰፊ of two words 'ሆዱ' which means 'abdomen' and 'ሰፊ' which means 'wide'. The obvious meaning is 'one who have big abdomen', but it can also be used as 'polite'. Therefore an Amharic IR system should differentiate between the various meanings of the different word combination before retrieving documents for a query. A single Amharic word can be a polysemous word, because of how it is read (i.e. stressing or not stressing on a letter of a word). Such as the word ገና, has two meaning. When the letter '□ is stressed it means 'too early', On the other hand. When 'ገና' is read loosely it can mean 'Christmas'. If the IR system doesn't understand the meaning of a word in terms of the way it is read, it becomes difficult to capture the relevant document among the rest of non-relevant documents. Other Amharic words can also be understood as adjectives or nouns in different sentences. For example in the sentence 'አበበ ልክ አንበሳ ነዉ' (i.e the sentence in English 'Abebe is just like a lion') the word 'አንበሳ' can mean 'strong' or 'hero'. On another sentence (i.e. the sentence in English this is a lion') the word 'አንበሳ' has its obvious meaning 'animal lion' as a noun.

These words can have associated with the culture of the people who uses the language. Things the society is used, or engages in now, can give birth to a combination of words indicating new meaning for the future. Other source can be the various kinds of language that are used with the existing language that can and have the potential to come up with new meanings for an existing word. And, even anthers can make up words have new meaning. Therefore there is no clear cut where this process of creating new meaning for words, will or suppose to end, so as to 'flourishment' of polysemous words.

Due to this situation it is difficult to design effective IR systems because of the polysemous nature of Amharic language, unless expert records them every time when they face. But such task is monotonous and need a lot of effort. Nonetheless, for existing polysemous terms, there is a need of designing an algorithm that automatically analyzes and attach appropriate meaning to them.

2.10. Review of Related Works

2.10.1. Global IR Research

2.10.1.1. Concept Based Query Expansion

According to Yonggang Qiu and H.P. Frei [41] Query expansion methods have been studied for a long time - with debatable success in many instances. The paper present a probabilistic query expansion model based on a similarity thesaurus which was constructed automatically. A similarity thesaurus reflects domain knowledge about the particular collection from which it is constructed. They address the two important issues with query expansion: the selection and the weighting of additional search terms. In contrast to earlier methods, there queries are expanded by adding those terms that are most similar to the concept of the query, rather than selecting terms that are similar to the query terms. The experiments show that this kind of query expansion results in a notable improvement in the retrieval effectiveness when measured using both recall-precision and usefulness.

They used the three standard test collections by compared the retrieval effectiveness of their automatic query expansion approach with the standard retrieval method using original queries only. For the collections CACM and MED, after extracting all the words from the collections and removing stop words, we used stemmed terms to index both queries and documents.

The results were evaluated by applying the average precision of a set of queries at three representative recall points, namely 0.25, 0.50, and 0.75. In addition, the usefulness was measured to compare the automatic query expansion method with the standard method using original queries.

The main results of the study are summarized as:

1. The automatic query expansion method based on statistical co-occurrence data can result in significant improvement in the retrieval effectiveness when measured using both recall-precision and usefulness. Consistent performance improvement was achieved in both automatically indexed test collections and a test collection indexed by carefully chosen terms.
2. Since the quality of the similarity thesaurus created for a large collection seems to be better than the one for a small collection, the retrieval effectiveness seems to increase with the size of the collection. Likewise, the number of additional terms per query seems to increase with the size of the collection too.
3. The methods that rely on relevance feedback information only select among the terms of a few retrieved documents. In contrast, the method described selects additional search terms out of the entire term set. Therefore, the number of additional search terms is usually larger.

2.10.1.2. Towards more effective techniques for automatic query expansion

According to Claudio Carpineto and Giovanni Romano [42] Techniques for automatic query expansion from top retrieved documents have recently shown promise for improving retrieval effectiveness on large collections but there is still a lack of systematic evaluation and comparative studies. This paper focus on term-scoring methods based on the differences between the distribution of terms in (pseudo-)relevant documents and the distribution of terms in all documents, seen as a complement or an alternative to more conventional techniques. They use distributional methods to select expansion terms within Rocchio's classical reweighting scheme; the overall performance is not likely to improve. However, they also show that when the same distributional methods are used to both select and weight expansion terms the retrieval effectiveness may considerably improve. They then argue, based on their variation in

performance on individual queries, that the set of ranked terms suggested by individual distributional methods can be combined to further improve mean performance, by analogy with ensembling classifiers. The performance of experiments shows that with automatic query expansion it is possible to achieve performance gains as high as 21.34% over non-expanded query

2.10.2. Local IR Research

Alemayehu's work [12] which used thesaurus based query expansion methods to enhance the system's recall at the expense of its precision. The performance analysis shows that there is an enhancement of recall from an average of 0.29% to 0.73%. On the other hand, because of the tradeoff between recall and precision and because of polysemous query term existence, his proposed system decreased the overall precision from 0.91% to 0.57% on average before and after using query expansion. While there is a decline of precision because of the heterogeneous nature of the cluster enhancing recall of the retrieval system while decline the precision and thesaurus terms have no related concepts with the query terms that are generated by the automatic query expansion.

Abiy's [6] has done semantic based query expansion for Amharic information retrieval; he tried to enhance the system performance of precision but cannot match the recall level of the original query. He used statistical co-occurrence, bi-gram methods and bi-gram thesaurus. In statistical co-occurrence method precision improved by 14% while refined query, bi-gram thesaurus has all improved by 18%. Even though the above researcher tried to implement different methods to enhance the performance of the Amharic document retrieval by using query expansion but, still there are problem of handling polysemous words in order to retrieve Amharic document.

The above two research were conducted on the query expansion area only still these researcher ignore the importance of integrating query expansion with term re-weighting to overcome the problem of retrieving relevant documents from Amharic corpus, but these is the main focus of the present research, so the researcher apply a term re- weighting based approach that have a paramount importance to enhance the performance of Amharic document retrieval system by integrating query expansion and term re-weighting.

Chapter Three

3.1. Designing term re- weighting based query expansion model

Query expansion technology extends the original query by seeking for some terms or phrases which are close or related to the initial query to express the user's query intention more precisely [4]. Also this technology can remove the ambiguity of the user's query term and improve the accuracy of search engines significantly. Query terms are used for document retrieval in information retrieval systems. An IR system takes a string query, and retrieves documents based on a certain similarity measurement technique. An IR system has two techniques to measure the performance of the system i.e. precision and recall. As any other systems' performance measurement, it is difficult to score 100% on both precision and recall in the case of information retrieval systems. But good systems are designed to enhance both precision and recall to the possible limit. Thus the aim of this research is, to design a good system, which can enhance the system performance.

This can be achieved by using two query reformulation techniques i.e. term re-weighting (based on the user relevance judgment), i.e. modifying the terms weights) and query expansion method (adding new terms that are attached to the retrieved documents found relevant). Integrating query re-weighting with query expansion model with the system is the main objective of these research, so that it finds words having similar meanings with users' query and retrieves relevant documents that satisfies users' information need.

The main function of a term weighting is the enhancement of retrieval effectiveness. Effective retrieval depends on two factors: one, item likely to be relevant to the user's need must be retrieved; two, items likely to be extraneous must be rejected. Two measures are normally used to assess the ability of the system to retrieve the relevant and reject the non- relevant items of a collection, known as recall and precision respectively.

Due to the promising prospect of query expansion technology, many researches have been done on this topic in information retrieval. Rocchio [43] presented a query expansion method based on Vector Space Model. The key of the method is mainly on query reformulation in order to make the query closer to the terms in the relevant documents. Xu and Croft [44] proposed the local and

global analysis method for query expansion. In this paper the author compared the retrieval effectiveness of different query expansion techniques and shown that local analysis is more effective than global analysis. Cui [45] presented a probabilistic model for query expansion by mining the query logs. The idea of this method is to construct the probabilistic correlations between query terms and document terms. Then the probabilistic correlations can be used to select high-quality expansion terms for the newly coming queries. Lin [46] presented a query expansion method for document retrieval. The proposed method represented documents and queries in Vector Space Model (VSM). The additional query terms were found by evaluating the importance of relevant terms and the different weights were inferred through fuzzy rules. Alemayehu [12] presented Application of Query Expansion for Amharic information retrieval System. The key of this method is mainly use automatic query expansion to improve retrieval effectiveness. Since thesaurus is used to query expansion. Abiy [6] presented Semantic Based Query Expansion Technique for Amharic IR. The key of his method is using different query expansion methods such as: bi-gram, bi-gram thesaurus and statistical co-occurrence to enhance precision without affecting recall.

In this paper, we propose a term re- weighting based query expansion method based on re-weighing terms in order to mine the additional query terms so as to enhance the system performance. Then we re-weight the terms in the document set by calculating the similarity between them.

In this research, a query with four or less words which are considered as short query, are used for experimentation purpose. The system is tested for its tendency of depriving, the ambiguities of short and polysemous word containing queries. Polysemous words are words having two or more different meanings. Unlike humans, computers couldn't understand the essence of a meaning. This fact poses a constraint for IR systems' performance [47].

3.2. Query Reformulation

Retrieved documents have a certain order as ranked according to their relevancy. But, there is no guarantee that most of the top documents are relevant. Therefore, there must be other ordering metric to get the most relevant documents at the top. If we can express a query in terms of aspects which are terms of related ideas, then it is possible to put a new ordering metric in terms

of aspects [4]. Aspects in this context mean a group of query terms. These aspects are made up of two or more users query terms. Finally, the refined query made up of a couple of aspects is redirected to the IR system, hoping to get relevant documents for the pseudo-relevance feedback.

Metric clustering technique is used to measure if two query terms are similar or dissimilar in meaning. A brief discussion of how it works is discussed as follows. The basic idea of metric clustering is co-occurrence, except this method compares similarity based on the number of terms between two queries. The similarity is calculated by adding the inverses of the word count in-between two morphological sets of words. Given a morphological set of A_1, A_2, and A_3 and another set B_1, B_2 and B_3, where their steams are A and B respectively, and given a text $A_1\ c_2\ g_3\ A_2\ k_4\ B_2\ r_4\ g_6\ A_3\ u_7\ B_1\ t_5\ h_4\ y_7\ g_5\ B_3$, the similarity of A and B can be calculated as.

$$Sim(A,B) = \frac{1}{r(A1,B1)} + \frac{1}{r(A1,B2)} + \frac{1}{A1,B3} + \frac{1}{A2,B1} + \frac{1}{A2,B2} + \frac{1}{A2,B3} + \frac{1}{A3,B1} + \frac{1}{A3,B2} + \frac{1}{A3,B3}.$$

$$Sim(A,B) = \frac{1}{9} + \frac{1}{4} + \frac{1}{14} + \frac{1}{6} + \frac{1}{1} + \frac{1}{11} + \frac{1}{1} + \frac{1}{2} + \frac{1}{6}$$

$$Sim(A,B) = 3.356782106782106782106782106782 \quad \ldots.(3.1)$$

The similarity value as can be seen is not between 0 and 1, which means that it is not normalized. A normalization equation is given in [4]. A normalized metric cluster similarity is calculated by dividing the un-normalized similarity by the multiplication of morphology magnitude of the two term steams. The above un-normalized similarity can be calculated as:

$$Norm(sim(A,B)) = \frac{sim(A,B)}{|V(A)|} * |V(B)|$$

$$Norm(sim(A,B)) = \frac{3.356782106782106782106782106782}{3*3}$$

$$Norm(sim(A,B)) = 0.372975789642456309122975789642 44 \quad \ldots\ldots\ldots(3.2)$$

Therefore 0.37 is the similarity of query term A and B. From the value gained in this process the program then can decide whether A and B can be connected by AND or OR based on a given

threshold. In this research the threshold 0.05 is set to be the margin for similarity and dissimilarity through experimentation for this particular document corpus. According to the prototype implemented for this research therefore, A and B are connected with OR in between as long as 0.37 is above the threshold 0.05.

If two query terms does not have the same meaning as per the evaluation metric clustering, then they are connected using AND, and they are connected with OR otherwise. Finally, the refinement process ends by connecting all the aspects using OR operator, to give chances of being selected, to documents having only one of the aspects. An IR system given such a query retrieves documents having all the aspects at the top and documents having only one of the aspects at the bottom.

Given query terms $q1$, q2,...qm and a certain constant x the procedure for analyzing a query in terms of aspects is given in Algorithm 3.1.

```
For i=0 to m do
for j=i to m do
if (i != j) then
Compute similarity sim (qi,qj) using metric clustering
normalize Sim(qi,qj) using Equation 3.2.
if (sim(qi,qj) > x) then
add aspect (qi OR qj) to aspects
else
add aspect (qi AND qj) to aspects
separate each added aspect with OR.
return aspects
```

Algorithm 3.1 Pseudo code for refining original query

The refinement process considers only the top k query terms. This is because, too many terms increase the ambiguity of the query and also has an effect on the computational time. If the query has less than k query terms, it considers all of them. Then, a pair of query terms is taken in to account to calculate their similarity. After normalizing the similarity value, based on the result

the system decides to form an aspect of two query terms connected with the logical operator OR or AND.

The decision is made based on a clustering method called metric clustering. Metric clustering, unlike other clustering methods like association clustering, considers the number of terms between two terms for which similarity is needed [4]. The strategy of considering how much two terms are apart for similarity measurements gives an advantage to be precise.

3.3. Query expansion

This research is a continuation of semantic based query expansion for Amharic IR used by [6]. This semantic based query expansion for Amharic IR is the base for the current research as it was for [6]. This semantic based query expansion for Amharic IR doesn't use term re-weighting techniques. It only focused on three query expansion techniques like bi-gram, bi-gram thesaurus and statistical co-occurrence. The main aim of this research is using the above query expansion technique by integrating with term re-weighting in order to make the system complete and to enhance the performance of the system.

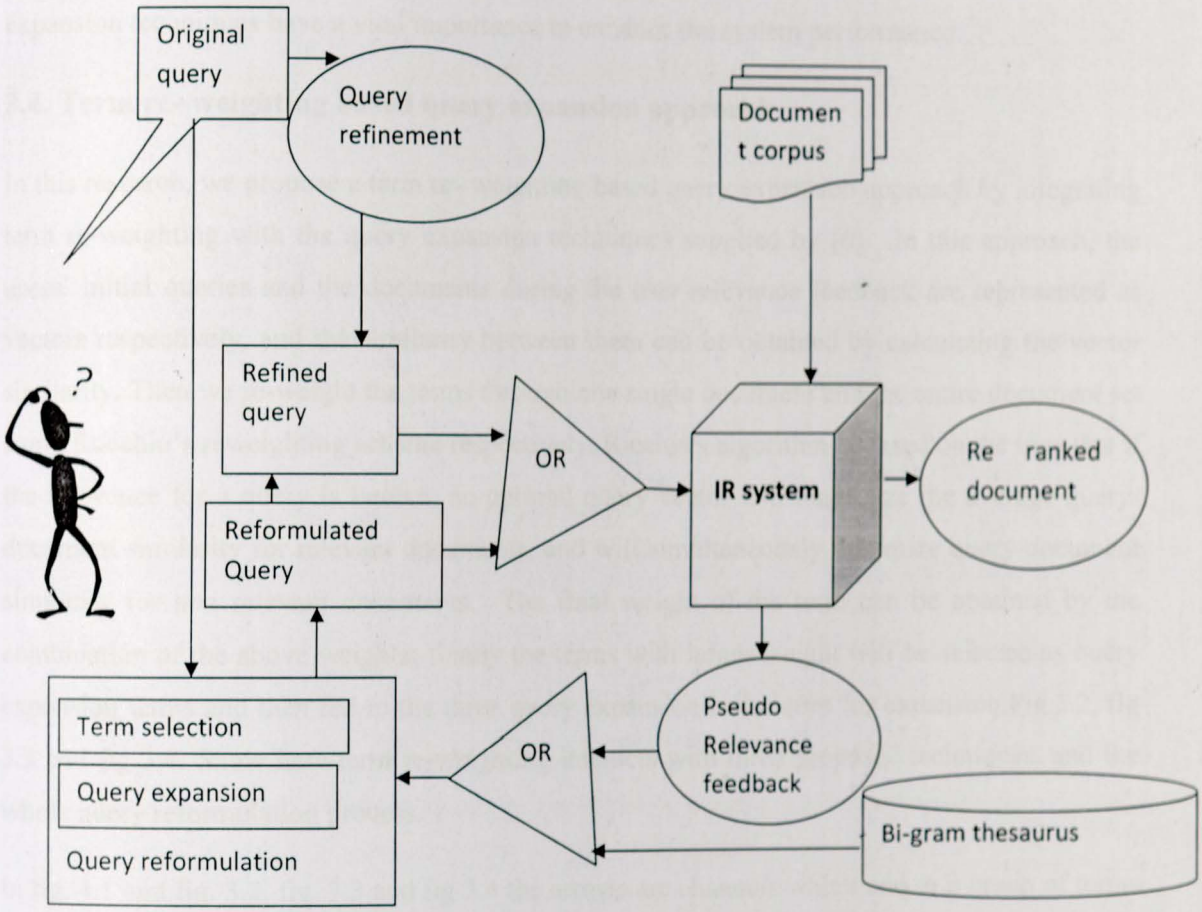


Figure.3.1. the architectural view of query expansion for three techniques adopted by Abiy bruck.

Abiy [6] Presented the query expansion technique i.e. bi-gram, statistical co-occurrence and bi-gram thesaurus and their common processes such as

- Refining the original query
- passing the refined query to the IR system and
- forming the pseudo-relevance feedback

But in his research the importance of term re-weighting is left. Here in the present research term re-weighting is integrated with statistical co-occurrence, bi- gram and bi-gram thesaurus query expansion techniques have a vital importance to enhance the system performance.

3.4. Term re- weighting based query expansion approach

In this research, we propose a term re- weighting based query expansion approach by integrating term re-weighting with the query expansion techniques supplied by [6]. In this approach, the users' initial queries and the documents during the user relevance feedback are represented as vectors respectively, and the similarity between them can be obtained by calculating the vector similarity. Then we re-weight the terms through one single document and the entire document set using Rocchio's reweighting scheme respectively. Ricchio's algorithm is based on the idea that if the relevance for a query is known, an optimal query vector will maximize the average query-document similarity for relevant documents, and will simultaneously minimize query-document similarity for non relevant documents. The final weight of the term can be obtained by the combination of the above weights; finally the terms with larger weight will be selected as query expansion terms and then fed to the three query expansion techniques for expansion.Fig.3.2, fig 3.3 and fig 3.4. Show how term re-weighting interacts with three proposed techniques, and the whole query reformulation process.

In fig. 3.1 and fig. 3.2, fig. 3.3 and fig 3.4 the arrows are channels which one or a group of terms pass through. The two three angles are gates that works differently for different techniques.

The statistical co-occurrence and the bi-gram methods share the following processes.

- Refining the original query
- passing the refined query to the IR system and
- forming the pseudo-relevance feedback
- Term selection
- Term re-weighting and finally
- Query expansion

Using the pseudo-relevance feedback users' query is refined. This refinement process is used again in the case of the bi-gram and bi-gram thesaurus methods. On the other hand statistical co-

occurrence method doesn't need to use the refinement process. That is because, it already selected common expanding terms as a solution for polysemous terms problem. Unlike the other two methods, the bi-gram thesaurus doesn't use the pseudo-relevance feedback. It rather uses a pre-built thesaurus based on the underlining theory used for the bi-gram method [6].

The bi-gram method and bi-gram thesaurus can't select expansion terms for the whole query rather they generate expanding terms for each query term. This is because in these techniques it is most unlikely to find many common terms.

In this research, the three query expansion techniques that are used by [6] is evaluated and how these three techniques accept query terms from the term re- weighting techniques and from the pre build thesaurus are discussed.

3.4.1. Statistical co-occurrence method (technique 1)

Statistical co-occurrence method accepts best weighted query terms from the term re-weighting techniques; the query expansion task is carried out based on the refined document set. For relevance feedback the first k out of the refined documents are selected based on the weight of the terms. That is because; it is assumed that query refinement process ought to populate more relevant documents at the top retrieved documents set. Hence these best weighted k documents are responsible for generating the expansion words. To overcome the polysemous characteristics of query terms, expansion terms are selected by its weight which represent every query terms. This is done by selecting those frequent terms which appear in every expansion terms set of every query term.

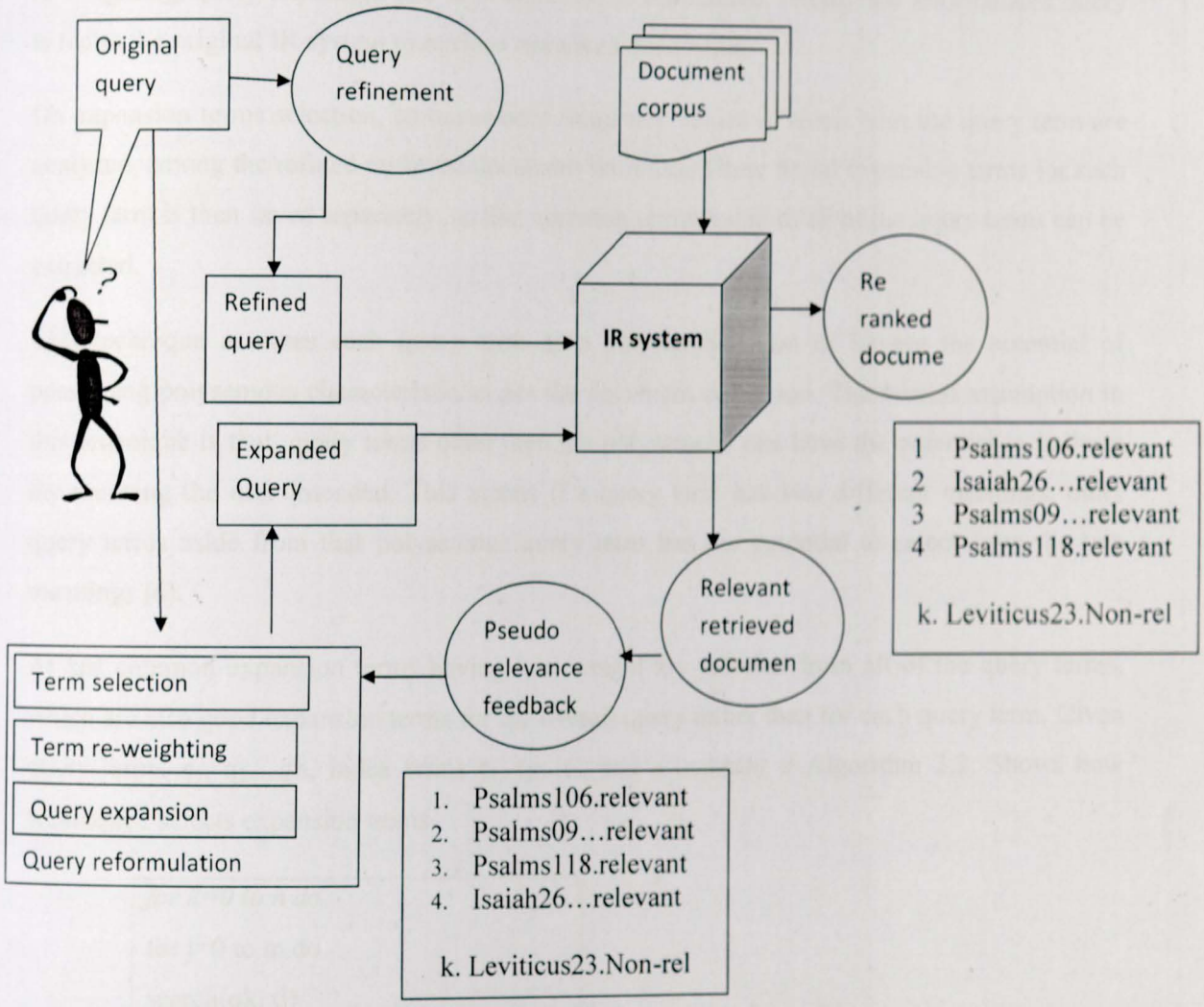


Figure 3.2 The architectural view of term re-weighting with Statistical co-occurrence techniques

The set of top 10 documents retrieved using the refined query is the pseudo relevance feedback for statistical co-occurrence method. These 10 relevant retrieved documents are indexed in a separate inverted index to extract expansion terms easily. This index is over written every time a query is given to the system. That is because; different relevant documents are retrieved for each

3.4.2. Bi-gram (technique 2)

The idea behind the Bi-gram model is guessing a word coming after $N-1$ words. Guessing the next word or word prediction is an essential subtask of speech recognition, hand-writing recognition, augmentative communication for the disabled, and spelling error detection [6].

N-gram uses probability coefficient to predict which word is appropriate to come after a given $N-1$ word sequence. Unlike the statistical co-occurrence method, the idea of selecting common expanding terms for the whole query doesn't hold true for this technique. That is because it is unusual to find two query terms apart with a single term in the middle, which that term in the middle is the only term which can be considered as a common expansion term. Therefore top expansion terms for each query term in equal amount are taken as a total expansion terms for the whole query. Therefore, if k amount of terms are needed for expansion and n amount of query terms are given, k/n amount of terms are taken from each query term expansion terms.

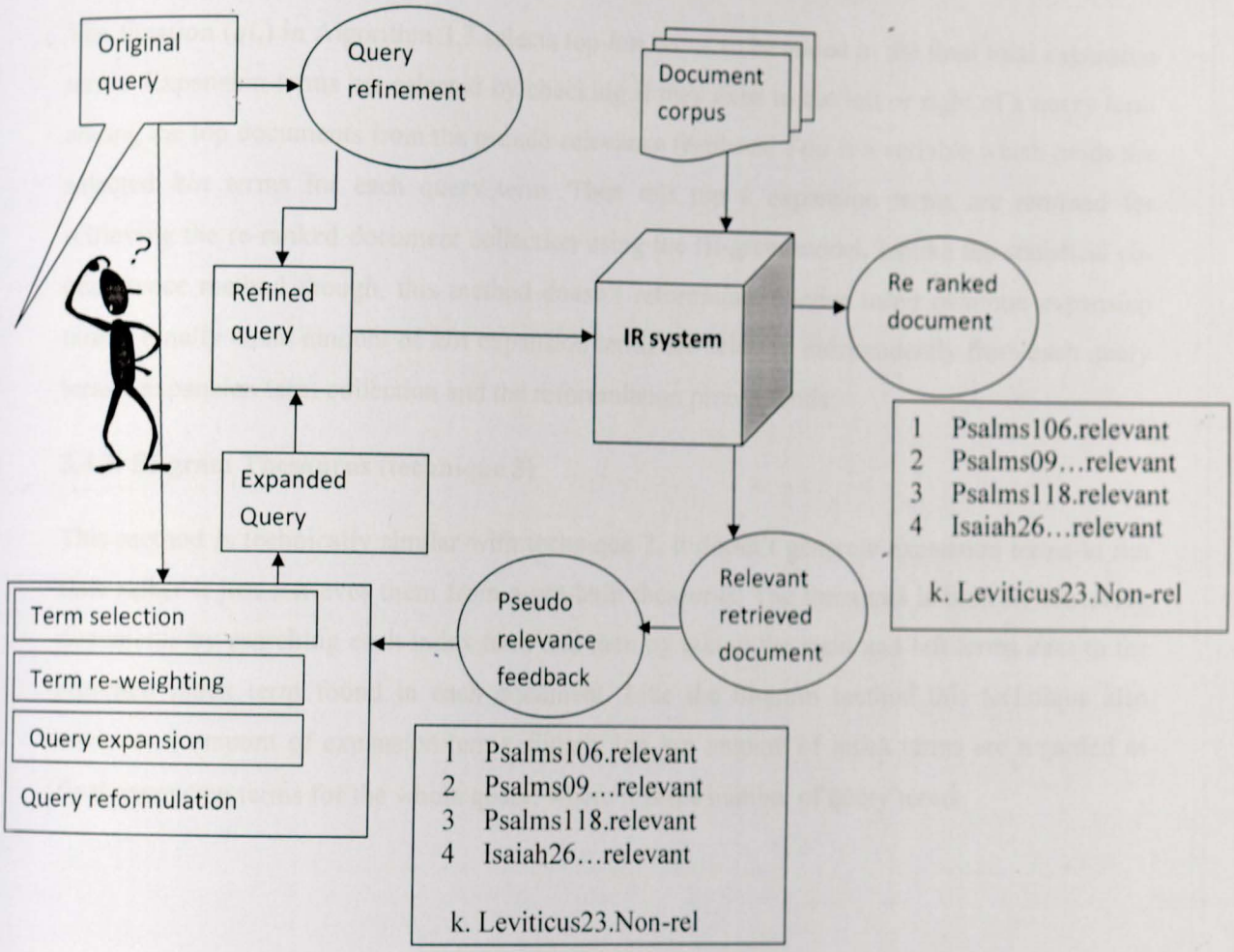


Figure 3.3 The architectural view of term re-weighting with bi-gram techniques

Given $_1$,q_2… query terms and documents $_1$, $_2$… as pseudo-relevance feedback, Algorithm 3.3. Shows how Bi-gram based expansion terms selection works.

```
accept(q1…qn, D1…Dm)
for i=0 to n do
for j=0 to m do
top = Select(qi,Dj) using      2   11
final = Add(Top)
return final
```

Algorithm 3.3 Pseudo code for Bi-gram based expansion terms selection

The function $(qi,)$ in Algorithm 3.3 selects top k/n terms to be added in the final total expansion terms. Expansion terms are selected by checking if they exist to the left or right of a query term among the top documents from the pseudo-relevance feedback. *Top* is a variable which holds the selected k/n terms for each query term. Then this top k expansion terms are returned for retrieving the re-ranked document collection using the Bi-gram model. Unlike the statistical co-occurrence method though, this method doesn't reformulate queries using common expansion terms. Finally equal amount of k/n expansion terms are selected independently from each query term's expansion term collection and the reformulation process ends.

3.4.3. Bi-gram Thesaurus (technique 3)

This method is technically similar with technique 2, it doesn't generate expansion terms in run time rather it just retrieves them from a pre-built thesaurus. The thesaurus is built by retrieving documents by searching each index term and then by taking the right and left terms next to the searched index term found in each document. Like the bi-gram method this technique also retrieves k amount of expansion terms. Finally top k/n amount of index terms are regarded as final expansion terms for the whole query, where n is the number of query terms.

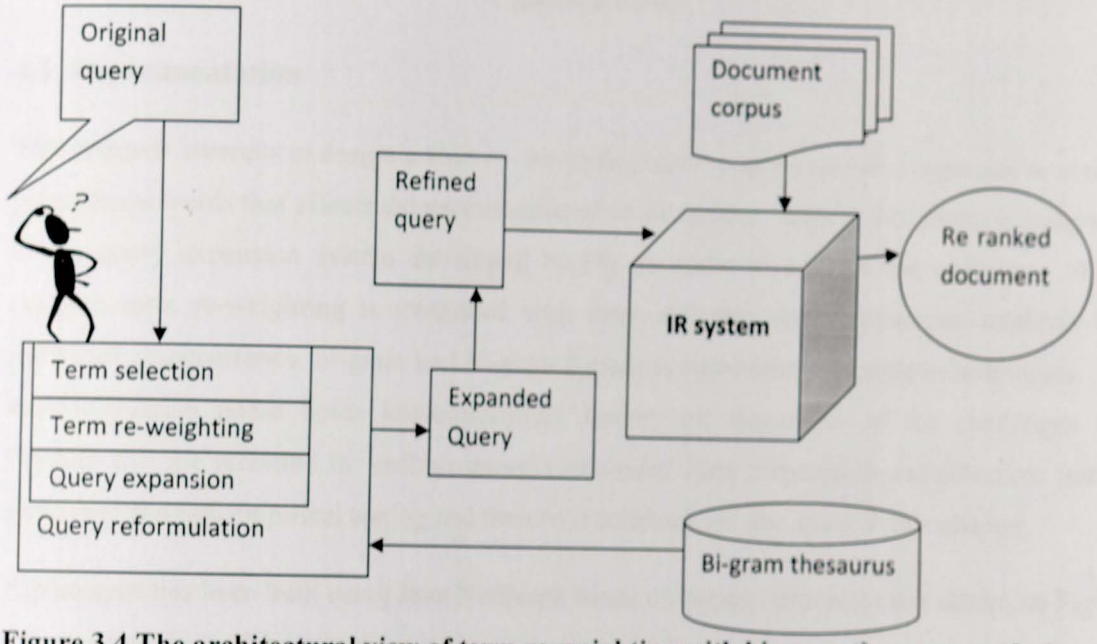


Figure 3.4 The architectural view of term re-weighting with bi-gram thesaurus techniques

```
For i=0 to n do
For j=0 to m do
If q[i]=t[j]
Top = k/n expansion terms from thesaurus
Return Top
```

Algorithm 3.4 A Pseudo code for Bi-gram thesaurus based query expansion

Given query terms 1, 2… and thesaurus terms 1, 2… Algorithm 3.4 shows an algorithm for Bi-gram thesaurus based query expansion.

As in bi-gram method is a variable to hold every / amount of expansion terms for each query term. Expansion terms are automatically added on variable whenever a query term equates a thesaurus term. The final step is returning the expansion terms for refinement.

Chapter Four

4.1. Experimentation

This research attempts to design a term re- weighting based query expansion approach to control polysemous words that affects the performance of an IR system. Term re-weighting is integrated to the query expansion system developed by [6]. In order to achieve the objectives of the research term re-weighting is integrated with three different query expansion methods i.e.; statistical co-occurrence, bi-gram and bi-gram thesaurus based query expansion techniques. The experimentation phase holds implementation, testing and discussion of the challenges and findings that are recorded for each proposed techniques. Data preparation and selection, testing procedure through empirical testing and threshold selection are discussed in this chapter.

A prototype has been built using Java NetBeans based on the architectural view shown on Figure 4.1. A screen shot of the prototype is shown on Figure 4.1.

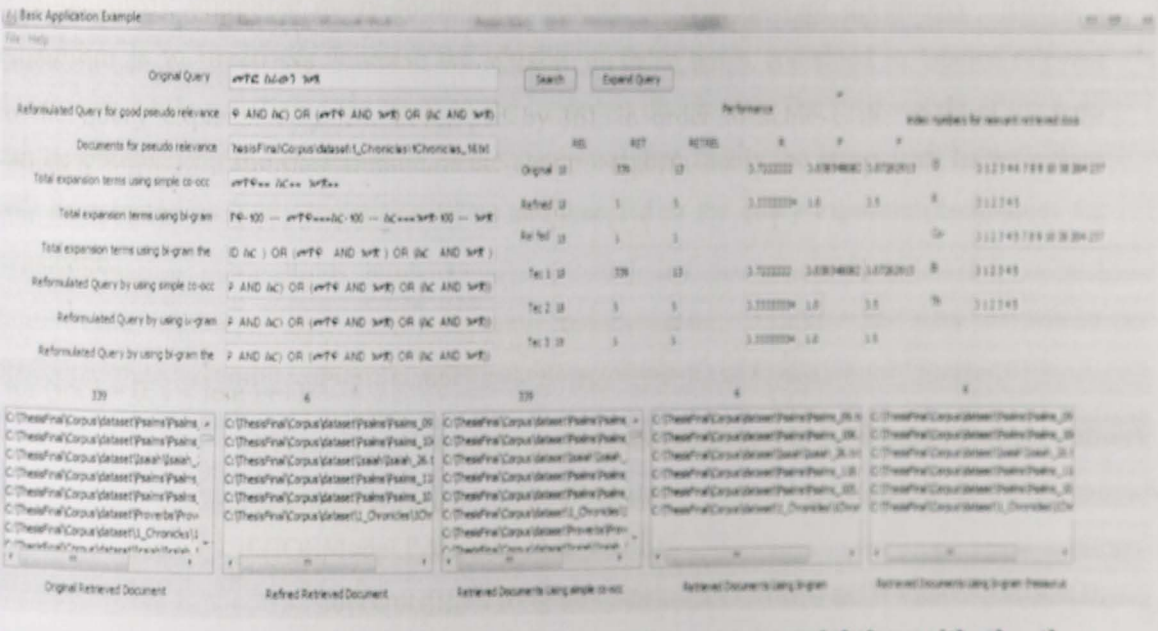


Figure 4.1 A screen shot of the prototype built for term re-weighting with the three proposed techniques

4.2. Dataset preparation

The experiment is carried out on the Amharic Bible Old Testament taken as a document corpus. It contains 21,000 stemmed terms and 930 documents. Since a system centered testing procedure is carried out, test reference collection is prepared for the test queries. Twelve queries with polysemous terms are formulated for testing.

4.3. Term re-weighting methods

In this approach, the users' initial queries and the documents during the user relevance feedback are represented as vectors respectively, and the similarity between them can be obtained by calculating the vector similarity. Then we re-weight the terms through one single document and the entire document set using Rocchio's re-weighting scheme respectively. Ricchio's algorithm is based on the idea that if the relevance for a query is known, an optimal query vector will maximize the average query-document similarity for relevant documents, and will simultaneously minimize query-document similarity for non relevant documents. Ricchio's Algorithm is to iteratively increase the weights of those terms contained in labeled relevant documents while penalizing the terms in the irrelevant documents. The final weight of the term can be obtained by the combination of the above weights; finally the terms with larger weight will be selected as query expansion terms and then fed to the query expansion techniques for expansion.

```
for (int i = 0; i < len; i++){
                    allTerms[i].setWeightExpansion(allTerms[i].getWeightExpansion()/normaliser);
                    expandedTerm[i].normalisedFrequency= terms[i].getWeightExpansion()/normaliser;
                    if (!QEModel.PARAMETER_FREE){
                            allTerms[i].setWeightExpansion(allTerms[i].getWeightExpansion());
                            normalisedFrequency *= QEModel.ROCCHIO_BETA;
        }
    }
```

Algorithm 4.1 shows a java written code for calculating the weight for each query term

This method implements the functionality of assigning expansion weights to the terms in the top-retrieved documents, and returns the most informative terms among them. Query Expansion is used, if the number of expanded terms is set to 0. In this case, no new query terms are added to the query, only the existing ones re-weighted. The number of terms to extract from the top-retrieved documents is set if this parameter is set to 0. If the parameter is greater than 0 it assign weight to terms that are stored in expansion term used for query expansion.

4.4. Query refinement

The query refinement process is implemented to make the pseudo-relevance feedback as good as possible. Users are not satisfied in separating their query terms using *OR*s and *AND*s and thus an automatic way of clustering query terms is necessary [6]. Query terms separated in these logical operators contribute to broaden the information domain which the user intended. This process delivers a query with terms separated in logical operators.

The implementation of this process first should have a way to select query terms in a manner that any of the terms are not combined again (i.e. (q1 *AND* q2) and (q2 *AND* q1) must not be present in the refined query at the same time. This is because computational time and operational cost of the system can be wasted if the same kind of aspect is formulated twice.

Algorithm 4.2 shows the way java written program handles separating two query terms without repeating the combination.

```
for(int i=0; i < queryTerms.length; i++)
for(int j=i; j < queryTerms.length; j++)
if(i != j) //checking out the two query terms are not the same
String[] booleanResult = ResultReturner.foundDocsReturner(queryTerms[i],
queryTerms[j]);
//Boolean result is documents in which query terms i and j exist.
```

Algorithm 4.2 Separating query terms and combining them back in pairs

The other task is to decide whether the combined query terms connect by *AND* or by *OR*. If the two query terms have same meaning then they connect by the logical operator *OR*, because one

of the terms can represent the meanings of both query terms. And if the two query terms have different meaning they connect using *AND*, because neither of them can represent the meaning of both query terms.

Algorithm 4.3 shows how a java written program handles the metric clustering algorithm to calculate similarity between pairs of query terms.

```
for(int x=0; x<booleanResult.length; x++)
    int[] term1Position = TermPositionReturner(booleanResult[x],queryTerms[i]);
    int[] term2Position = TermPositionReturner(booleanResult[x],queryTerms[j]);
    for(int z=0; z<term1Position.length; z++)
    for(int y=0; y<term2Position.length; y++)
    if(term1Position[z] > term2Position[y])
    similarity = similarity + (term1Position[z] - term2Position[y]);
else
    double devidend = ((term2Position[y] - term1Position[z])/5);
    similarity = similarity + (1/devidend);//metric similarity calculation
    similarity = similarity/(term1Position.length * term2Position.length);
    return similarity;
```

Algorithm 4.3 Metric clustering similarity calculation and normalization

Algorithm 4.4 shows how a java written program handles the decision that two query terms should be connected using the logical operator *OR* or *AND*.

```
if (similarity > 0.05)
newQuery = queryTerms[i] + " OR " + queryTerms[j]
else
newConstructedQuery = queryTerms[i] + " AND " + queryTerms[j]
return newConstructedQuery;
```

Algorithm .4.4 Combining query terms with logical operator *OR* or *AND*.

Query	REL	Using refined query				
		RET	RETREL	R	P	F
መጥፎ ስራውን ገሠጸ	18	6	6	0.33	1.0	0.5
እጥፍ ሰጠው	26	9	9	0.35	1.0	0.51
እጥፍ አድርጎ ሰጠው	69	130	14	0.20	0.11	0.14
ሲጨልም አንቀላፋ	10	14	10	1.0	0.71	0.83
መጥፎ ስራውን ገሰጸ	18	6	6	0.33	1.0	0.5
ማቅ በወገቡ ለበሰ	69	8	7	0.10	0.88	0.18
ጌታዬ ባርያህ ነኝ	69	97	33	0.48	0.34	0.40
ቀና እና ቅን ስራዎች	140	223	80	0.57	0.36	0.44
ሰው የተባለ አለቀ	10	32	7	0.7	0.21	0.32
አመታት ተቀመጠ	326	5	5	0.02	1.0	0.03
አመድ እና ማቅ ለበሰ	35	5	5	0.17	1.0	0.29
ዘይት ቀባው	69	10	10	0.14	1.0	0.25
			Avg.	0.37	0.71	0.54

The system performance using the refined testing queries is presented on Table 4.1.

Table 4.1 Systems performance using refined query

The refined query or new constructed query is then fed to the original IR system after terms are re-weighted to retrieve refined documents. Finally these refined documents are compared with those documents retrieved using the original query, to find out how much the refined query alter the pseudo-relevance feedback.

4.5. Statistical co-occurrence method

This method generates synonymous expanding terms for a query term based on index terms co-occurrence information. It analyzes the presence of a term with another term to decide whether they have same or different meaning [6].

On expansion terms selection, co-occurrence frequency values of terms with the query term are analyzed, among the refined retrieved document set index. These found expansion terms for each query term is then saved separately, so that common terms found in all of the query terms can be extracted. Algorithm 4.5 shows a java written code for expanding term selection for each query terms.

```
for(int i=0; i<termsToSelectFromArray.length; i++)
if(!termsToSelectFromArray[i].isEmpty())
int docCount = hitCount(queryTerm + " AND " + termsToSelectFromArray[i]);
if(docCount >= 1)
expandingTerms = expandingTerms + termsToSelectFromArray[i] + "-" + docCount
return expandingTerms.
```

Algorithm 4.5 Statistical co-occurrence based query expansion

In the above java code terms are ordered in descending order based on the frequency of query terms. The final task is to select those terms which are common for the whole query in order to eliminate polysemious terms ambiguity characteristics. Algorithm 4.6 shows how a java written code selects common expansion terms.

This sample code shows how common expanding terms are selected for the whole query among the prepared expanding terms by the code shown on Algorithm 4.6.

```
for(int i=0; i<queryTermsArray.length; i++)
File x = new File("C:/meanings/" + queryTermsArray[i] + ".txt");
StringBuffer read1 = new StringBuffer();
read1 = read("C:/meanings/" + smallSizeQueryTerm +".txt");
String[] sentence = rade.toString().split(" ");
for(int t=0; t<queryTermsArray.length; t++)
if(!commonTerms.equals(""))
sentence = commonTerms.split(" ");
commonTerms = "";
if(!queryTermsArray[t].equals(smallSizeQueryTerm))
StringBuffer read2 = new StringBuffer();
read2 = read("C:/meanings/" + queryTermsArray[t] +".txt");
String[] sentence = read2.toString().split(" ");
for(int j=0; j<sentence.length; j++)
if(sentence2.contains(sentence[j]))
commonTerms = commonTerms + " " + sentence[j];
return commonTerms;
```

Algorithm 4.6 Common expansion terms selection for statistical co-occurrence technique

System centric testing is carried out to evaluate the statistical co-occurrence technique's performance. The statistical co-occurrence's performance is recorded in terms of recall, precision and f-measure as shown on Table 4.2.

Query	REL	Using statistical co-occurrence methods				
		RET	RETREL	R	P	F
መኖሬ ስራውን ገሠጸ	13	18	339	0.04	0.72	0.07
እኖኍ ሰጠው	25	26	303	0.08251	0.96	0.16
እኖኍ አድርጎ ሰጠው	23	69	486	0.04	0.33	0.08
ሲጨልም አንቀላፋ	10	10	14	0.71	1.0	0.83
መኖሬ ስራውን ገሰጸ	13	18	339	0.03	0.72	0.07
ማቅ በወገቡ ለበሰ	34	69	81	0.41	0.49	0.45
ጌታዬ ባርያህ ነኝ	99	69	415	0.23	0.43	0.40
ቀና እና ቅን ስራዎች	129	140	683	0.19	0.92	0.31
ሰው የተባለ አለቀ	7	10	28	0.25	0.7	0.36
አመታት ተቀመጠ	219	326	258	0.84	0.67	0.75
አመድ እና ማቅ ለበሰ	28	35	66	0.42	0.8	0.55
ዘይት ቀባው	46	69	123	0.37	0.66	0.48
			Avg.	**0.30**	**0.70**	**0.50**

Table 4.2 Statistical co-occurrence's performance

4.6. Bi-gram based query expansion

The bi-gram based query expansion, as the statistical co-occurrence method, uses the index terms co-occurrence information. But unlike the statistical co-occurrence method, this method considers those terms which appear just side by side to the query term, as expansion terms. For example in a text "A B C" where A B and C are terms let us say B is the query term, then A and

C are taken as expansion terms according to this technique. The name bi-gram is given to this method because it only considers a query term and its immediate successor or predecessor [6]. This method is used for various other purposes mentioned on chapter three [6].

Bi-gram is implemented in this research with slightly different approach; by letting it to consider a query terms immediate predecessor also. From its implementation, an assumption arises which states that a query terms immediate predecessor and successor can be a good expansion term.

This method is again slightly similar with a statistical co-occurrence method because both uses the pseudo-relevance feedback to select expansion terms for each query term. It is also slightly different from statistical co-occurrence method, because this method doesn't extract common terms for the whole query. The java written code for the bi-gram based query expansion technique is seen in algorithm 4.7.

```
for(int t=0; t<queryTerms.length; t++)
if(!queryTerms[t].equals(firstQueryTerm))
expandingTerms = expandingTerms + "===";
finalExpandingTerms = "";
String tempExpandingTerms = getExpansionTerms(queryTerms[t], doc);
String[] expandingTermsArray = tempExpandingTerms.split(" ");
for(int j=0; j<expandingTermsArray.length; j++)
double probability = getProbability(expandingTermsArray[j],tempExpandingTerms);
if(probability > 0.0)
if(!finalExpandingTerms.contains(expandingTermsArray[j]))
finalExpandingTerms = finalExpandingTerms + expandingTermsArray[j] + "-" + probability +
" " ;
String orderedExpandingTerms = order(finalExpandingTerms);
expandingTerms = expandingTerms + orderedExpandingTerms + " ";
return expandingTerms;
```

Algorithm 4.7 Expanding term selections using the bi-gram method

This code takes the query and the pseudo-relevance feedback and returns expanding terms for each query term based on the weight of terms. Each query term has its own collection of expanding terms ordered in descending probability values. This probability is calculated by considering the probability of finding an expanding term preceding or succeeding a given query term among documents from the relevance feedback [6].

The expansion terms found are finally organized and put in to the refined query to generate the reformulated query. For example if a query is given as "*A B C*" and if "*a*1,*a*2,*a*3","*b*1 ,*b*2 ,*b*3" and "*c*1,*c*2,*c*3" are the expansion terms for *A B* and *C* respectively, and if the refined query is given as (*A OR B*) *OR* (*A AND C*) *OR* (*A AND C*), the reformulated query looks like (*A a*1 *a* 2 *a*3 *OR B b*1 *b*2 *b*3) *OR* (*A a*1 *a*2 *a*3 *AND C c*1 *c* 2 *c*3) *OR* (*A a*1 *a*2 *a*3 *AND C c*1 *c*2 *c*3).

A java written code to generate the final reformulated query, integrated with the refined one is given in Algorithm 4.8.

```
String[] separation1 = expansionTerms.split("===");
 int noOfTermsToSelect = 20/separation1.length + 1;
 for(int i=0; i<separation1.length; i++){
 finalExpansionTerms = "";
 String[] separation2 = separation1[i].split("---");
 if(!separation2[1].contentEquals(" ")){
 String[] termsToSelectFrom = separation2[1].split(" ");
 for(int j=0; j<noOfTermsToSelect; j++){
 finalExpansionTerms = finalExpansionTerms + termsToSelectFrom[j] + " ";
 String[] finalExpansionTermsArray = finalExpansionTerms.split(" ");
 tempReformulatedQuery = tempReformulatedQuery.replace(finalExpansionTermsArray[1],
 finalExpansionTerms);
 return tempReformulatedQuery;
```

Algorithm 4.8 Reformulation of expanded query using the bi-gram method

System centric testing is carried out after integrating with term re-weighting to evaluate the bi-gram method's performance. The bi-gram methods performance is recorded in terms of recall, precision and f-measure as shown on Table 4.3.

		Using bi-gram method				
	REL	RET	RETREL	R	P	F
መጥፎ ስራውን ገሠጸ	18	6	6	0.33	1.0	0.5
እጥፍ ሰጠው	26	9	9	0.35	1.0	0.51
እጥፍ አድርጎ ሰጠው	69	130	14	0.20	0.10	0.14
ሲጨልም አንቀላፋ	10	14	10	1.0	0.71	0.83
መጥፎ ስራውን ገሰጸ	18	6	6	0.33	1.0	0.5
ማቅ በወገቡ ለበሰ	69	8	7	0.10	0.875	0.18
ጌታዬ ባርያህ ነኝ	69	97	33	0.48	0.34	0.39
ቀና እና ቅን ስራዎች	140	223	80	0.57	0.36	0.44
ሰው የተባለ አለቀ	10	32	7	0.7	0.21	0.32
አመታት ተቀመጠ	326	5	5	0.02	1.0	0.03
አመድ እና ማቅ ለበሰ	35	5	5	0.17	1.0	0.29
ዘይት ቀባው	69	10	10	0.14	1.0	0.25
			Avg.	0.37	0.72	0.55

Table 4.3 Systems performance using bi-gram method

3.7. Bi-gram Thesaurus based query expansion

This technique is based on the bi-gram method. The fact that this method doesn't use pseudo-relevance feedback makes it automatic global analysis method than local analysis. That is because it searches each index term and analyzes all the top retrieved documents to generate expansion terms, and save them in a separate thesaurus file for that particular index term [6]. It

also appears to be similar with the bi-gram based expansion method because both use the refined query to deliver the final reformulated query.

In a bi- gram thesaurus there is no need of searching with the refined query to get a pseudo-relevance feedback rather it just retrieve them from a pre-built thesaurus. The java written code for reformulating a query based on a bi-gram thesaurus is shown as follows in algorithm 4.9.

```
for(int i=0; i<queryTerms.length; i++)
String file = getDoc(docDir,queryTerms[i]);
if(!file.equals("null"))
FileInputStream finn = new FileInputStream(new File(file));
InputStreamReader readd = new InputStreamReader(finn, "utf-8");
BufferedReader rdd = new BufferedReader(readd);
String line1;
StringBuffer sb1 = new StringBuffer();
while ((line1 = rdd.readLine()) != null)
sb1.append(line1);
rdd.close();
String sentence1 = sb1.toString();
if(!sentence1.contentEquals("null") && !sentence1.contentEquals("---"))
String[] temp1 = sentence1.split("---");
System.out.print(queryTerms[i]);
String[] temp2 = temp1[1].split(" ");
if(temp2.length >= noOfTerms)
for(int j=0; j<noOfTerms; j++)
expansionTerms = expansionTerms + temp2[j] + " ";
else
for(int j=0; j<temp2.length; j++)
expansionTerms = expansionTerms + temp2[j] + " ";
refinedQuery = refinedQuery.replace(queryTerms[i], queryTerms[i] + " " + expansionTerms);
return refinedQuery;
```

Algorithm 4.9 Bi-gram thesaurus based query expansion

The code shown on Algorithm 4.9 returns the reformulated query and also refines the expanded query in the manner the first refinement process is done on the original query. Finally the reformulated and refined query is redirected for further searching.

Testing is carried out after integrating with term re-weighting to evaluate the bi-gram thesaurus method's performance. The bi-gram thesaurus methods performance is recorded in terms of recall, precision and f-measure as shown on Table 4.4.

Query	REL	Using the bi-gram thesaurus				
		RET	RETREL	R	P	F
መጥፎ ስራውን ገሠጸ	18	6	6	0.33	1.0	0.5
እጥፍ ሰጠው	26	9	9	0.34	1.0	0.51
እጥፍ አድርጎ ሰጠው	69	6	1	0.01	0.16	0.02
ሲጨልም አንቀላፋ	10	14	10	1.0	0.71	0.83
መጥፎ ስራውን ገሰጸ	18	6	6	0.33	1.0	0.5
ማቅ በወገቡ ለበሰ	69	8	7	0.10	0.87	0.18
ጌታዬ ባርያህ ነኝ	69	97	33	0.47	0.34	0.39
ቀና እና ቅን ስራዎች	140	206	63	0.45	0.30	0.36
ሰው የተባለ አለቀ	10	32	7	0.70	0.21	0.32
አመታት ተቀመጠ	326	9	9	0.02	1.0	0.05
አመድ እና ማቅ ለበሰ	35	5	5	0.17	1.0	0.29
ዘይት ቀባው	69	10	10	0.14	1.0	0.25
			Avg.	**0.34**	**0.71**	**0.52**

Table 4.4 Systems performance using the bi-gram thesaurus

4.8. Testing

For testing, system oriented approach has been followed. That means a pre defined relevant documents as per the researcher is prepared before engaging the techniques for retrieval. For testing twelve queries, each having a polysemous term, are used. These queries are categorized as easy, average and tough. The proposed techniques are tested with the whole of the document corpus. First compiled performance presentation of the proposed techniques is presented, followed by comparisons with the original systems performance. Finally the best performer among the proposed techniques is selected.

Average precision, recall and F-measure results of the proposed techniques are compiled along with the original and refined query performances on Table 4.5.

	Average Recall	Average Precision	Average F-measure	
Original query	**0.83**	**0.39**	**0.53**	Before expansion
Refined query	**0.36**	**0.71**	**0.54**	
Statistical co-occurrence	**0.30**	**0.70**	**0.50**	After expansion
The Bi-gram method	**0.37**	**0.72**	**0.55**	
B-gram thesaurus	**0.34**	**0.71**	**0.52**	

Table 4.5 Compiled average recall, precision and f-measure for proposed techniques

Before integrating term re-weighting with the three query expansion techniques the system enhance the average precision by 14% for statistical co-occurrence methods and the other two methods both augmented it by 18% but, the recall level of the original query [6]. After integrating term re-weighting with the three techniques the precision level becomes augmented by 31% for statistical co-occurrence method and 33% and 32% for bi-gram and bi-gram thesaurus methods respectively. In terms of f-measure the bi-gram method outperformed by improving it in total of 2% when compared to original query.

In order to achieve the most optimal result, fixed thresholds for the number of expanding terms has been set. For statistical co-occurrence, bi-gram and bi-gram thesaurus, 8, 13, and 9 numbers of expanding terms has been found to be optimal respectively. Note that, all of these expanding terms have been selected from the most frequently co-occurring terms. It has also been tested using rarely co-occurring terms and results showed that such expanding terms are good for improving precision, but they are poor on their recall level. In addition it is tested using both frequently and rarely co-occurring expanding terms, but results were not satisfactory as that of expanding a query using only frequently co-occurring terms.

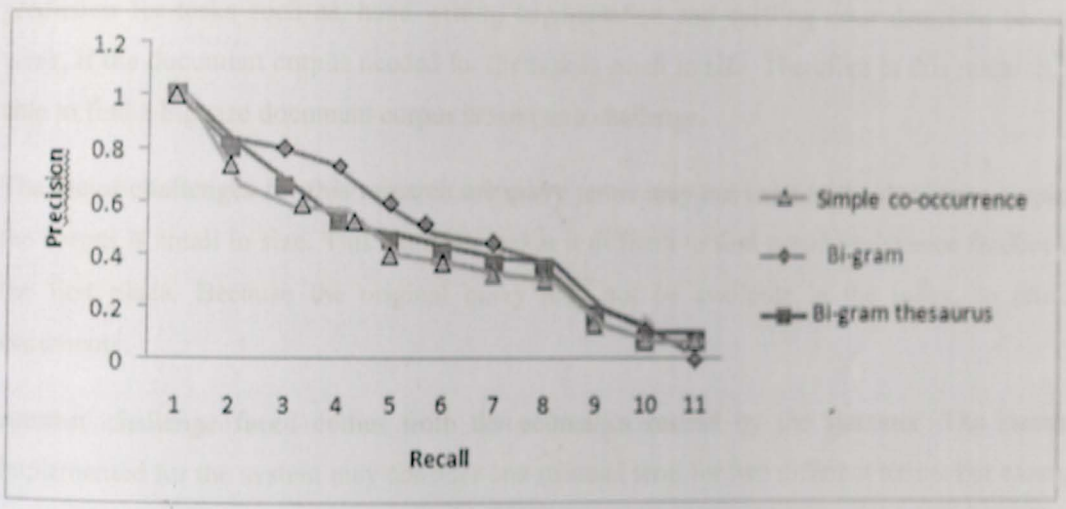


Figure 4.2 Interpolated graph for performances of the three methods

Figure 4.2 shows the graphical representation of the three proposed techniques performance on a recall precision graph. The technique used to build the graph has been adopted from [4]. First the precision of each technique tested on each query for 11 recall levels is recorded. Then by taking the average of the precision at each recall level, such a graph has been presented.

As can be seen from the graph the bi-gram method out performs bi-gram thesaurus methods and statistical co-occurrence methods. All techniques demote the recall of the system but, it has recorded a good precision for the top retrieved documents

4.9. Discussion

In this research, the proposed methods are tested using different sizes corpus. This is necessary to find out in what kind of size a method performs well. Unfortunately when the sizes of the corpus used is small in size, not only the performance weakens but also the basic underlying idea of expanding a query gets into jeopardy. For example testing the system using a small size document corpus, it becomes difficult for the statistical co-occurrence method to generate common expanding terms. This is because; the document corpus is too small to return common query terms. This in turn affects the basic hypothesis which states that, common terms can overcome the problem caused by polysemous terms. As [6] mentioned, the results in word prediction for tasks such as, hand-writing augmentation and spelling error detection becomes weak, if the document corpus needed for the task is small in size. Therefore in this research, not able to find a big size document corpus is seen as a challenge.

The major challenges for this research are query terms may not exist in the document corpus if the corpus is small in size. This situation makes it difficult to find pseudo-relevance feedback in the first place. Because the original query may not be available in the index, to retrieve documents.

Another challenge faced comes from the anomalies caused by the stemmer. The stemmer implemented for the system may consider one steamed term for two different terms. For example a stemmed term "አሰቅ" is given for terms "አሰቃ" and "አሰቀ" which have meanings "boss" and "finished" respectively. Another example is a word "ቀን", which is given as a steam for "ቀን" and "ቀና", which have meanings "day light" and "genial" respectively. This situation in turn distorts the whole word co-occurrence information, guiding the IR system in a falsified direction to retrieve documents.

The stemming algorithm again may give different stemmed terms for one term which makes the statistical information for that term poor. Therefore, such errors subject a system to consider morphologically different but same in meaning, to have different meanings. The consequence is that, more false positives and false negatives populate among the true positives among retrieved document.

Polysemous query terms guide an IR system towards retrieving documents, which are not related to the query in a meaning a user intended. This is basically the problem that initiated this research, and approaches have been introduced as a solution. By treating each query term as polysemous, the proposed method intends to find common expansion terms for the whole query

In order to avoid this problem other experiments have been done. Then it might retrieve documents having all the meaning a user intended, but as the same time it might lowers the recall level and enhances the precision. Results have been recorded for such reformulated query, to have 100% for precision and 0.01% or lower for the recall level. This happens because, finding documents having all the terms is difficult, and even if they are found they are considerably less in number than all of the relevant documents in the corpus.

For expanding terms not to be polysemous, the original users query terms were taken as a guarantee to seek the intended meaning and vice versa. But results clearly showed that, both original users query terms and expanding terms can appear in different contexts, conveying completely different meanings. Therefore, there is a need for IR systems to adopt an approach to check weather expanding terms are polysemous or not.

Chapter Five

5. Conclusion and Recommendation

5.1. Conclusion

In this paper, we have proposed a method for query expansion by re-weighting the terms. With the help of this approach we can extract query expansion terms from the entire document set. The proposed query expansion method uses the vector space model to represent documents and queries. The difference is that optimized documents are added to the identified documents and we assign different weights to them. And the final weights of the terms can be obtained and sorted. The terms with higher weight will be chosen as the query expansion terms.

For query expansion, three techniques are integrated with term re-weighting and explored. The statistical co-occurrence technique, which selects expansion terms from the pseudo relevance feedback by analyzing terms co-occurrence information with query terms. The bi-gram technique which also uses the pseudo relevance feedback for expansion, selects those terms which appear to the right or left of a query term. Finally the bi-gram thesaurus technique unlike the other two uses a pre-built thesaurus to select expansion terms. Though the underlining theory of the bi-gram thesaurus is similar with the bi-gram method, unlike the other methods it uses the whole of the document corpus to select expanding terms and thus it is global context analysis.

Experiments are done to explore the extent to which these techniques improve precision of the system. The results show that the bi-gram method outperformed in terms of F-measure by 2%. It has also been observed that all of the techniques improved the precision of the original system considerably. Statistical co-occurrence method augmented the precision by 31%, bi-gram methods augmented the precision by 33% and bi- gram thesaurus methods augmented by 32%.

Even though the bi-gram method scored high in terms of F-measure, all techniques showed weak performance in terms of recall. It is finally concluded that the bi-gram method is the better one in augmenting the precision and F-measure of the system.

To reduce the problem of polysemeous query terms we apply term re- weighting based query expansion approach by integrating term re-weighting with the three proposed query expansion

methods. However sometimes terms used for expanding query terms are polysemeous themselves with many meanings in different context.

5.2. Recommendations

As a continuation from this research, the following are future research directions to help the development of an IR system for local Ethiopian languages in general and Amharic language in particular.

- The present research attempts to improve the performance of Amharic IR system by integrating term re-weighting with query expansion. But Amharic terms that are used for expansion are polysemous themselves, creating the problem of retrieving irrelevant documents that affect the performance of the IR system. Future research may consider designing ontology-based query expansion technique to control polysemous terms in one way or another.
- Although the findings of this research seem to be fairly satisfactory, lack of effective Amharic stemmer is a major constraint on results of the system. Therefore it is recommended that future researches have to be carried out with the aim of designing a generic stemming algorithm for Amharic language.
- Finding a standard corpus and test queries with relevance judgment for testing the designed system is one of the challenges faced in this research. Therefore, future research need to consider the development of standard Amharic corpus that can be used by researcher to evaluate progress made in designing Amharic IR system.
- In this research, common expanding terms are selected, just by checking their existence in users query term's expanding terms set. This means that, a term can be either common to the whole query or not. Here, it is suggested to use fuzzy logic, to determine the extent to which a term can be common or not.
- In this research, terms are calculated using Rocchio's term re-weighting scheme. It could be better to check probabilistic term re-weighting methods in order to enhance both the precision and recall of the system.

Reference

1. Borghoff U.M & Pareschi R.Eds. "*information Technology for knowledge management*", Springer Verlag, 1998.
2. H.S. Christopher D.Manning, Prabhakar Raghavan, *"An Introduction to Information Retrieval"*, 1st Edition, Cambridge University Press, Cambridge England, 2009.
3. Wing, X, "*improving web search for difficult queries*", 1st edition, Cambridge university press, Cambridge England, 2009.
4. B. R., Ribeiro Neto, *"Modern Information Retrieval"*, 2nd Edition, Addison-Wesley-Longman Publishers, New York, USA 1999.
5. Xiuowei Sheng, Minghu Jiung, " *An information retrieval system based on automatic query expansion and hopfzeld network*", lab of computational linguistics, department of Chinese language, Tsinghua university Beijing, 100084,china.
6. Abiy Bruck, "Semantic Based Query Expansion Technique for Amharic IR", MSc Thesis, School of Information Science, Addis Ababa University, Ethiopia, 2011.
7. F.D. Commission, "Summary and Statistical Report of the 2007 Population and Housing Census", pp. 1-113, 2008.
8. Andargachew Mekonen Gezmu, "*Amharic thesaurus construction for Amharic text retrieval*", MSc Thesis, Addis Ababa University School of Information science, Addis Ababa, Ethiopia, 2009.
9. Bethelehem M., "*N-Gram-Based Automatic Indexing for Amharic Text*", MSC Thesis, Addis Ababa University School of Information science, Ethiopia, 2002.
10. Tewodros H., "*Amharic text retrieval: An Experiment Using Latent Semantic Indexing (LSI) with Singular Value Decomposition (SVD)*", MSc Thesis, Addis Ababa University, School of Information science, Ethiopia, 2003.
11. Tesema M. and Solomon A., "*Design and Implementation of Amharic Search Engine*", fifth international conference on signal image technology and internet based systems, 3(1):318-325, 2009.
12. Alemayehu ,"Application of Query Expansion for Amharic information retrieval System", MSc Thesis, Addis Ababa University, School of Information Science, Ethiopia, 2010.
13. Yogesh Kakde, *"a survey of query expansion until June 2012"*, Indian Institute of Technology, Bombay, June 2012.

14. Abiy Z. , Alemayehu W. , Daniel T. , Melese G. and Yilma S. , "*Introduction to Research Methods*", 1st Edition, Graduate studies and Research Office of Addis Ababa University, 2009.

15. http:www.//iyesus.com/downloads/amharic-bible-software/iota/ Accessed Date; 3/2/2013.

16. Amanuel Hirpa, "*probabistic information retrieval system for Amharic language*". MSc Thesis, Addis Ababa University, School of Information Science, july 2012.

17. http://hitachiid.com/concepts/java.html#sthash.zmTdZZiu.dpuf Accessed Date; 6/2/2013.

18. http://www.techopedia.com/definition/24735/netbeans Accessed Date; 6/2/2013.

19. Vannevar Bush. *As We May Think. Atlantic Monthly*, 176:101–108, July 1945.

20. Warren Weaver; "*Translation*," pages 15-27 in Machine Translation of Languages, eds. W. N.Locke and A. D. Booth, John Wiley, New York(1955).

21. H. P. Luhn. "*A statistical approach to mechanized encoding and searching of literary information*". *IBM Journal of Research and Development*, 1957.

22. Amit Singhal, *Modern Information Retrieval: A Brief Overview*, singhal@google.com

23. Gerard Salton, editor. "*The SMART Retrieval System—Experiments in Automatic Document Retrieval*", Prentice Hall Inc., Englewood Cliffs, NJ, 1971.

24. C. W. Cleverdon. "*The Cranfield tests on index language devices*". *Aslib Proceedings*, 19:173–192, 1967.

25. D. K. Harman. "*Overview of the first Text REtrieval Conference*" (TREC-1). In *Proceedings of the First, Text REtrieval Conference (TREC-1)*, pages 1–20. NIST Special Publication 500-207, March 1993.

26. D. Hiemstra, "*Information Retrieval Models*", John Wiley and son's publisher, Enschede-Noord, Netherlands , 1st Edition, 2009, 2009.

27. Ed Greengrass, "*Information Retrieval: A Survey by Ed Greengrass*",Availableat;www.csee.umbc.edu/cadip/readings/IR.report.120600.book.pdf, Accessed Date; 3/2/2013, 2013.

28. H.S. Christopher D.Manning, Prabhakar Raghavan, "*An Introduction to Information Retrieval*", 1st Edition, Cambridge University Press, Cambridge England, 2009.

29. A. Singhal, "*Modern Information Retrieval*: A Brief Overview", Bulletin of the IEEE Computer Society Technical Committee on Data Engineering, 24(4): pp. 35-42, 2001.

30. www.dsoergel.com, "*The Scope of IR Utility, Relevance, and IR System Performance*", Available at; http:// www.dsoergel.com%2FNewPublications%2FHCIEncyclopediaIRSh rtEF orDS .pd, Accessed Date; 7/2/2013, 1997.

31. Chris Buckley, Gerard Salton, and James Allan. "*Automatic retrieval with locality information using SMART*". In *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 59–72. NIST Special, Publication 500-207, March 1993.

32. H.R. Turtle and W.B. Croft, "A Comparison of Text Retrieval Models", The Computer Journal, 35(2): 279-290, 1992.

33. Information Retrieval Model", Available at; http://comminfo.rutgers.edu/~aspoerri/InfoCrystal/ Ch_2.htm, Accessed Date; 12/2/2013.

34. N. Fuhr, "*Probabilistic Models in Information Retrieval*", the Computer Journal, 35(3):243-255, 1992.

35. S.W. k. Sparck Jones, "*A probabilistic Model of Information Retrieval: Development and Comparative Experiments*", Information Processing and Management, 36(4): 779-808, 2000.

36. C.J. Rijsbergen, "*A Theoretical Basis for the Use of Co-occurrence Data in Information Retrieval*", Journal of Documentation, 23(2):106--119, 1977.

37. M.L. Krieg, "*A Tutorial on Bayesian Belief Networks*", Available at; http://www.dsto.defence. gov. au, Accessed Date; 15/2/2013.

38. T. Bloor, "*The Ethiopic Writing System: a Profile*", Journal of the Simplified Spelling Society, 19(1): 30-36, 1995.

39. M. Bender, "*The Ethiopic Writing System*", Oxford University Press, London, 1976.

40. Dagnachew. A. & worku,A. "*yeamaaregna Felitoch*", kuraz Asatami Derget, (1986).

41. Yonggang Qiu and H.P. Frei," *Concept Based Query Expansion*", Department of Computer Science, Swiss Federal Institute of Technology, Zurich, CH-8092 Zurich, Switzerland.

42. Claudio Carpineto and Giovanni Romano, "*Towards more effective techniques for automatic query expansion*", Fondazione Ugo Bordoni, Via B. Castiglione 59, I-00142, Rome, Italy.

43. Rocchio Jr., J.J.: *"The smart retrieval system: Experiments in automatic document processing"*. In: Relevance feedback in information retrieval, pp. 313-323, 1971.

44. Xu, J., & Croft, W. B. *"Query expansion using local and global document analysis"*. Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval, Zurich, Switzerland, pp.4-11, 1996.

45. Cui, H., Wen, J. R., Nie, J. Y., & Ma, W. Y. *"Probabilistic query expansion using query logs"*. Proceedings of the 11th international conference on World Wide Web, Honolulu, Hawaii, pp. 325-332, 2002.

46. Lin,H. C.,Wang, L. H.,& Chen, S.M. "*A new query expansion method for document retrieval by mining additional query terms*". Proceedings of the 2005 international conference on business and information, Hong Kong, China, 2005.

47. Germann, D.C., Villavicencio,A. and Siqueira M. *"An investigation on polysemy and lexical organization of verbs*", Proceedings of the NAACL HLT 2010 first workshop on computational Neurolinguistics, Los Angeles California, pp. 52-60, 2010.

APPENDICES

Appendix I: Relevance judgment used

	ቀና እና ቅን ስራዎች	አመድ እና ማቅ ለበሰ	እጥፍ አድርጎ ሰጠው	ጌታዬ ባርያህ ነኝ	አመታት ተቀመጠ	መጥፎ ስራውን ገለጸ	ዘይት ቀባው
Doc--1Chronicles_01.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--1Chronicles_04.txt	non relevant	non relevant	relevant	non relevant	relevant	non relevant	non relevant
Doc--1Chronicles_05.txt	non relevant	non relevant	relevant	non relevant	relevant	non relevant	non relevant
Doc--1Chronicles_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_08.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_09.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Chronicles_10.txt	non relevant	Relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Chronicles_11.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--1Chronicles_12.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_13.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Chronicles_14.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	relevant
Doc--1Chronicles_15.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_16.txt	non relevant	non relevant	non relevant	non relevant	non relevant	relevant	non relevant
Doc--1Chronicles_17.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Chronicles_18.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_19.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_20.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_21.txt	non relevant	Relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_22.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_23.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_24.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_25.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_26.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_27.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_28.txt	non relevant	non relevant	relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Chronicles_29.txt	Relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--1Kings_01.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	relevant
Doc--1Kings_02.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--1Kings_03.txt	Relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--1Kings_04.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Kings_05.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	relevant

Doc--1Kings_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Kings_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Kings_08.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Kings_09.txt	Relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Kings_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Kings_11.txt	Relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Kings_12.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--1Kings_13.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Kings_14.txt	Relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Kings_15.txt	Relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Kings_16.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Kings_17.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Kings_18.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--1Kings_19.txt	Relevant	non relevant	non relevant	relevant	non relevant	non relevant	relevant
Doc--1Kings_20.txt	non relevant	Relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Kings_21.txt	non relevant	Relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--1Kings_22.txt	Relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--1Samuel_01.txt	non relevant	non relevant	relevant	relevant	relevant	non relevant	non relevant
Doc--1Samuel_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Samuel_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Samuel_04.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Samuel_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Samuel_06.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Samuel_07.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Samuel_08.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Samuel_09.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--1Samuel_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--1Samuel_11.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Samuel_12.txt	Relevant	non relevant	non relevant	non relevant	relevant	non relevant	Relevant
Doc--1Samuel_13.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Samuel_14.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Samuel_15.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--1Samuel_16.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	Relevant
Doc--1Samuel_17.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Samuel_18.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Samuel_19.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Samuel_20.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--1Samuel_21.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Samuel_22.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--1Samuel_23.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant

Doc--1Samuel_24.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	Relevant
Doc--1Samuel_25.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--1Samuel_26.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	Relevant
Doc--1Samuel_27.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Samuel_28.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--1Samuel_29.txt	Relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--1Samuel_30.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--1Samuel_31.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_01.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_02.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_04.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_06.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	Relevant
Doc--2Chronicles_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_08.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_09.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_10.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_11.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_12.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_13.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_14.txt	Relevant	non relevant	non relevant	non relevant	relevant	non relevant	Relevant
Doc--2Chronicles_15.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_16.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_17.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_18.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--2Chronicles_19.txt	Relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_20.txt	Relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_21.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_22.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	Relevant
Doc--2Chronicles_23.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--2Chronicles_24.txt	Relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_25.txt	Relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_26.txt	Relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_27.txt	Relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_28.txt	Relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--2Chronicles_29.txt	Relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_30.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_31.txt	Relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_32.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant

Doc--2Chronicles_33.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_34.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Chronicles_35.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Chronicles_36.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Kings_01.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Kings_02.txt	non relevant	non relevant	relevant	relevant	relevant	non relevant	non relevant
Doc--2Kings_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Kings_04.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--2Kings_05.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--2Kings_06.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--2Kings_07.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Kings_08.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--2Kings_09.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	Relevant
Doc--2Kings_10.txt	relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--2Kings_11.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	Relevant
Doc--2Kings_12.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Kings_13.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Kings_14.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Kings_15.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Kings_16.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Kings_17.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Kings_18.txt	relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--2Kings_19.txt	non relevant	Relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--2Kings_20.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Kings_21.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Kings_22.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Kings_23.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Kings_24.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Kings_25.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Samuel_01.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	Relevant
Doc--2Samuel_02.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	Relevant
Doc--2Samuel_03.txt	non relevant	Relevant	non relevant	relevant	non relevant	non relevant	Relevant
Doc--2Samuel_04.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--2Samuel_05.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	Relevant
Doc--2Samuel_06.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Samuel_07.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--2Samuel_08.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--2Samuel_09.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--2Samuel_10.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--2Samuel_11.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant

Doc--Ecclesiastes_12.txt	non relevant	non relevant	non relevant	non relevant	non relevant	relevant	non relevant
Doc--Esther_01.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Esther_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Esther_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Esther_04.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Esther_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Esther_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Esther_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Esther_08.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Esther_09.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Esther_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus1.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus10.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Exodus11.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Exodus12.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Exodus13.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus14.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus15.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus16.txt	non relevant	non relevant	relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus17.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus18.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus19.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus2.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Exodus20.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus21.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Exodus22.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Exodus23.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Exodus24.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Exodus25.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus26.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus27.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus28.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Exodus29.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Exodus3.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus30.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Exodus31.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Exodus32.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Exodus33.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus34.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus35.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant

Doc--Exodus36.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus37.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Exodus38.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus39.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus4.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus40.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Exodus5.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus6.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus7.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus8.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Exodus9.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_01.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_03.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_04.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_07.txt	non relevant	relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_08.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_09.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_11.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_12.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_13.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_14.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_15.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_16.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--Ezekiel_17.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_18.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_19.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_20.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_21.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_22.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_23.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_24.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_25.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_26.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_27.txt	non relevant	relevant	non relevant	non relevant	relevant	non relevant	relevant
Doc--Ezekiel_28.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_29.txt	non relevant	non relevant	non relevant	non relevant			

Doc--Ezekiel_30.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_31.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_32.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_33.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_34.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_35.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_36.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_37.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_38.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_39.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_40.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_41.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_42.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_43.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_44.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_45.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_46.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezekiel_47.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezekiel_48.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezra_01.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezra_02.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezra_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezra_04.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezra_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezra_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezra_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezra_08.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Ezra_09.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ezra_10.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Genesis1.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis11.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis12.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis13.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis14.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis15.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis16.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis17.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis18.txt	relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Genesis19.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant

Doc--Genesis2.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis20.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis21.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis22.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis23.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Genesis24.txt	relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Genesis25.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis26.txt	non relevant	non relevant	relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis27.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Genesis28.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis29.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis3.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis30.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis31.txt	non relevant	non relevant	non relevant	non relevant	non relevant	relevant	relevant
Doc--Genesis32.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Genesis33.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Genesis34.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis35.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis36.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis37.txt	non relevant	relevant	non relevant	non relevant	relevant	relevant	non relevant
Doc--Genesis38.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis39.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Genesis4.txt	non relevant	non relevant	relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis40.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Genesis41.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis42.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Genesis43.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Genesis44.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Genesis45.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Genesis46.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis47.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Genesis48.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis49.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis5.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis50.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Genesis6.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis7.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis8.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Genesis9.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Habakkuk_01.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant

Doc--Habakkuk_02.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Habakkuk_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Haggai_01.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Haggai_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Hosea_01.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Hosea_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Hosea_03.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Hosea_04.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Hosea_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Hosea_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Hosea_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Hosea_08.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Hosea_09.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Hosea_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Hosea_11.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Hosea_12.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Hosea_13.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Hosea_14.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_01.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Isaiah_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_03.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Isaiah_04.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_05.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Isaiah_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_08.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_09.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_11.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_12.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_13.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Isaiah_14.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_15.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_16.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Isaiah_17.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_18.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Isaiah_19.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Isaiah_20.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_21.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_22.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant

Doc--Isaiah_23.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_24.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Isaiah_25.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_26.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_27.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_28.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_29.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_30.txt	relevant	non relevant	relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_31.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_32.txt	relevant	relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Isaiah_33.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Isaiah_34.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Isaiah_35.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_36.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Isaiah_37.txt	non relevant	relevant	non relevant	relevant	relevant	relevant	non relevant
Doc--Isaiah_38.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_39.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_40.txt	non relevant	non relevant	relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_41.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_42.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Isaiah_43.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_44.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_45.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_46.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_47.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_48.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_49.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_50.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_51.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_52.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Isaiah_53.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_54.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_55.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_56.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_57.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_58.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_59.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_60.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_61.txt	non relevant	non relevant	relevant	non relevant	non relevant	non relevant	relevant
Doc--Isaiah_62.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant

Doc--Isaiah_63.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_64.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_65.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Isaiah_66.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_01.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_04.txt	relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_06.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_07.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_08.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_09.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_10.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_11.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_12.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_13.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_14.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_15.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_16.txt	non relevant	non relevant	relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_17.txt	relevant	non relevant	relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_18.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_19.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_20.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_21.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_22.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_23.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_24.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_25.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_26.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_27.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_28.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_29.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_30.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_31.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_32.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_33.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_34.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_35.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_36.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant

Doc--Jeremiah_37.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Jeremiah_38.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Jeremiah_39.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_40.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_41.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_42.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_43.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_44.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_45.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_46.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_47.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_48.txt	non relevant	relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_49.txt	non relevant	relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Jeremiah_50.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_51.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jeremiah_52.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_01.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_02.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Job_03.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Job_04.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	relevant	non relevant
Doc--Job_06.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_08.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_09.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_11.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Job_12.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Job_13.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_14.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_15.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_16.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_17.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_18.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_19.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_20.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Job_21.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_22.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Job_23.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_24.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant

Doc--Job_25.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_26.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_27.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_28.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_29.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Job_30.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_31.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_32.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_33.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_34.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_35.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_36.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_37.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Job_38.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_39.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_40.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_41.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Job_42.txt	relevant	non relevant	relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joel_01.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joel_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joel_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jonah_01.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jonah_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jonah_03.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Jonah_04.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_01.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_02.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_03.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Joshua_04.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joshua_05.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Joshua_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joshua_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joshua_08.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_09.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joshua_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joshua_11.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joshua_12.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_13.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_14.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joshua_15.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant

Doc--Joshua_16.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_17.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_18.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joshua_19.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_20.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_21.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_22.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Joshua_23.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Joshua_24.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_01.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Judges_03.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Judges_04.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Judges_05.txt	non relevant	non relevant	relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_06.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Judges_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Judges_08.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_09.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_10.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_11.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_12.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Judges_13.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Judges_14.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Judges_15.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_16.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_17.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_18.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_19.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Judges_20.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Judges_21.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Lamentations_01.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Lamentations_02.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Lamentations_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Lamentations_04.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--Lamentations_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Leviticus1.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Leviticus10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--Leviticus11.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant

Doc--Leviticus12.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Leviticus13.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Leviticus14.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Leviticus15.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Leviticus16.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Leviticus17.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Leviticus18.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Leviticus19.txt	non relevant	non relevant	non relevant	non relevant	non relevant	relevant	non relevant
Doc--Leviticus2.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Leviticus20.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Leviticus21.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--Leviticus22.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Leviticus23.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Leviticus24.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Leviticus25.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Leviticus26.txt	non relevant	non relevant	relevant	non relevant	relevant	non relevant	non relevant
Doc--Leviticus27.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Leviticus3.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Leviticus4.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--Leviticus5.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Leviticus6.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--Leviticus7.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--Leviticus8.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	relevant
Doc--Leviticus9.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Malachi_01.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Malachi_02.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Malachi_03.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Malachi_04.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Micah_01.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Micah_02.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Micah_03.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Micah_04.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Micah_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Micah_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--Micah_07.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Nahum_01.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Nahum_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Nahum_03.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Nehemiah_01.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Nehemiah_02.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant

Doc--Nehemiah_03.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Nehemiah_04.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Nehemiah_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Nehemiah_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Nehemiah_07.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Nehemiah_08.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Nehemiah_09.txt	relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Nehemiah_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Nehemiah_11.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Nehemiah_12.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Nehemiah_13.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Numbers1.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers11.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Numbers12.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Numbers13.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers14.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Numbers15.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Numbers16.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers17.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers18.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--Numbers19.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers2.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers20.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Numbers21.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Numbers22.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Numbers23.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Numbers24.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers25.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Numbers26.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers27.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers28.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers29.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers3.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers30.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers31.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Numbers32.txt	non relevant	non relevant	non relevant	relevant	relevant	non relevant	non relevant
Doc--Numbers33.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Numbers34.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers35.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	relevant

Doc--Numbers36.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Numbers4.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	relevant
Doc--Numbers5.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers6.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	[redacted]
Doc--Numbers7.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	[redacted]
Doc--Numbers8.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Numbers9.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Obadiah_01.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_01.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_02.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Proverbs_03.txt	relevant	non relevant	non relevant	non relevant	non relevant	relevant	non relevant
Doc--Proverbs_04.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_06.txt	non relevant	non relevant	relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_08.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_09.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_11.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_12.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_13.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_14.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Proverbs_15.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_16.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_17.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_18.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_19.txt	non relevant	non relevant	non relevant	non relevant	non relevant	relevant	non relevant
Doc--Proverbs_20.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_21.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Proverbs_22.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_23.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_24.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_25.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_26.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_27.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Proverbs_28.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Proverbs_29.txt	relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Proverbs_30.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Proverbs_31.txt	non relevant	non relevant	relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_01.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant

Doc--Psalms_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_04.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_07.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_08.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_09.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Psalms_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_100.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_101.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_102.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_103.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_104.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_105.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	Relevant
Doc--Psalms_106.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_107.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Psalms_108.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_109.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_11.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_110.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_111.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_112.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_113.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_114.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_115.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_116.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_117.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_118.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_119.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_12.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Psalms_120.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_121.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_122.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_123.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Psalms_124.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_125.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_126.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_127.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_128.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant

Doc--Psalms_129.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_13.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_130.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_131.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_132.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	Relevant
Doc--Psalms_133.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Psalms_134.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_135.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_136.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_137.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Psalms_138.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_139.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_14.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_140.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_141.txt	non relevant	non relevant	non relevant	non relevant	non relevant	relevant	Relevant
Doc--Psalms_142.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_143.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_144.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_145.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_146.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_147.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Psalms_148.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_149.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_15.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_150.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_16.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Psalms_17.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_18.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Psalms_19.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_20.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Psalms_21.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_22.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_23.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Psalms_24.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_25.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_26.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_27.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Psalms_28.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Psalms_29.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Psalms_30.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant

Doc--Psalms_31.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_32.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_33.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_34.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_35.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_36.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_37.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Psalms_38.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_39.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_40.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_41.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_42.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_43.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_44.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_45.txt	relevant	non relevant	non relevant	relevant	non relevant	non relevant	Relevant
Doc--Psalms_46.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_47.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Psalms_48.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_49.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_50.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_51.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_52.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_53.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_54.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_55.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Psalms_56.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_57.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_58.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_59.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_60.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_61.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Psalms_62.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_63.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_64.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_65.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_66.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_67.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_68.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Psalms_69.txt	non relevant	relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Psalms_70.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant

Doc--Psalms_71.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_72.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_73.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_74.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_75.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_76.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_77.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_78.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_79.txt	non relevant	non relevant	relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_80.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_81.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_82.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_83.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_84.txt	relevant	non relevant	non relevant	non relevant	relevant	non relevant	Relevant
Doc--Psalms_85.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_86.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_87.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_88.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_89.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	Relevant
Doc--Psalms_90.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_91.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_92.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_93.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_94.txt	relevant	non relevant	non relevant	non relevant	non relevant	relevant	non relevant
Doc--Psalms_95.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_96.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_97.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_98.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Psalms_99.txt	non relevant	non relevant	non relevant	non relevant	Relevant	non relevant	non relevant
Doc--Ruth_01.txt	non relevant	non relevant	non relevant	non relevant	Relevant	non relevant	non relevant
Doc--Ruth_02.txt	non relevant	non relevant	non relevant	relevant	Relevant	non relevant	non relevant
Doc--Ruth_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Ruth_04.txt	non relevant	non relevant	non relevant	non relevant	Relevant	non relevant	non relevant
Doc--Songof Solomon_01.txt	relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Songof Solomon_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Songof Solomon_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Sonof Solomon_04.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Songof Solomon_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant

Doc--Song of Solomon_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Song of Solomon_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Song of Solomon_08.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Zechariah_01.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	non relevant
Doc--Zechariah_02.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Zechariah_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Zechariah_04.txt	non relevant	non relevant	non relevant	relevant	non relevant	non relevant	Relevant
Doc--Zechariah_05.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Zechariah_06.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Zechariah_07.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Zechariah_08.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--Zechariah_09.txt	non relevant	non relevant	relevant	non relevant	relevant	non relevant	non relevant
Doc--Zechariah_10.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Zechariah_11.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Zechariah_12.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Zechariah_13.txt	non relevant	relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--Zechariah_14.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--zephaniah_01.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant
Doc--zephaniah_02.txt	non relevant	non relevant	non relevant	non relevant	relevant	non relevant	non relevant
Doc--zephaniah_03.txt	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant	non relevant

Appendix II: Java written code:

Written code for term re-weighting

```
package amharicir;
/**
 * @author zish
 */
import org.apache.log4j.Logger;
import java.util.Arrays;
import java.util.Map;
import java.lang.String;
public class ExpansionTerms1 {
  /** The logger used */
        protected Logger logger;
        /** The terms in the top-retrieval documents. */
        protected HashMap<Character, ArrayList> originalTerms = new HashMap<Character,
ArrayList>();
    protected HashMap<Character, String> originalTermids = new HashMap<Character, String>();
```

```
ArrayList<String> word = new ArrayList<String>();
        /** The lexicon used for retrieval. */
        protected Lexicon lexicon;
        /** The number of documents in the collection. */
        protected int numberOfDocuments;
        /** The number of tokens in the collection. */
 String WeightingModel;
 String terms;
protected int EXPANSION_MIN_DOCUMENTS;
protected int NUMBER_OF_EXPANSION_DOCUMENTS;
protected double assignWeights;
protected double expandedTerms;
protected double Rounding;
        protected double numberOfTokens;
protected double normalisedFrequency;
protected String queryTermsArray;
protected double score;
        /** The average document length in the collection. */
        protected double averageDocumentLength;
        /** The number of tokens in the X top ranked documents. */
        protected double totalDocumentLength;
protected boolean NORMALISE_WEIGHTS = true;
        public void setNORMALISE_WEIGHTS(boolean normalise_weights) {
                NORMALISE_WEIGHTS = normalise_weights;
        }
                public double normaliser = 1d;
                protected int termID;
                /** The weight for query expansion. */
                protected double weightExpansion;
                /** The number of occurrences of the given term in the X top ranked documents. */
                protected double withinDocumentFrequency;
                /** The document frequency of the term in the X top ranked documents. */
                protected int documentFrequency;
                public ExpansionTerms1(int termID) {
                        this.termID = termID;
                }
                public ExpansionTerms1(int termID, double withinDocumentFrequency) {
                        this.termID = termID;
                        this.withinDocumentFrequency = withinDocumentFrequency;
                        this.documentFrequency = 1;
                        this.weightExpansion = 0;
                }
                public int getTermID(){
                        return this.termID;
                }
                public void insertRecord(double withinDocumentFrequency){
                        this.withinDocumentFrequency += withinDocumentFrequency;
                        this.documentFrequency++;
                }
                public void setWeightExpansion(double weightExpansion){
                        this.weightExpansion = weightExpansion;
                }
                public int getDocumentFrequency(){
                        return this.documentFrequency;
```

```
                }
                public double getWeightExpansion(){
                        return this.weightExpansion;
                }
                public double getWithinDocumentFrequency(){
                        return this.withinDocumentFrequency;
                }
        public int compareTo(ExpansionTerms1 o) {
                System.out.println(o.getWeightExpansion() + " vs " + this.getWeightExpansion());
                        if (o.getWeightExpansion() > this.getWeightExpansion()) {
                                return 1;
                        }
                        else if (o.getWeightExpansion() < this.getWeightExpansion()) {
                                return -1;
                        }
                        else {
                                return 0;
                        }
                }
        public ExpansionTerms1(
                        int numberOfDocuments,
                        long numberOfTokens,
                        double averageDocumentLength,
                        double totalLength,int lexicon) {
                this.numberOfDocuments = numberOfDocuments;
                this.numberOfTokens = numberOfTokens;
                this.averageDocumentLength = averageDocumentLength;
                this.originalTerms = new HashMap<Character,ArrayList>();
                this.totalDocumentLength = totalLength;
                this.lexicon = lexicon;
        }
   //returns index terms from a given index
   public static String returnIndexTerms(String index) throws IOException{
      StopStemmer stopStemmer = new StopStemmer();
      Directory directory = FSDirectory.getDirectory(index);
      IndexReader indexReader = getReader(directory);
      TermEnum termEnum = indexReader.terms();
      String term = "";
      while(termEnum.next()){
        if (!stopStemmer.isStop(termEnum.term().text())){
        if(!termEnum.term().text().toString().contains("0")
!termEnum.term().text().toString().contains("C:"))
            term = term + termEnum.term().text() + " ";
        }
      }
      return term;
   }
public static String returnCommonTerms(String queryTerms)throws Exception{
      String[] queryTermsArray = QueryTokenizer.tokenizeQuery(queryTerms);
      String sentence2 = "";
      long smallSize = 0;
      String smallSizeQueryTerm = "";
      String commonTerms = "";
      //selects a query term with the smallest number of expanding terms
```

```
for(int i=0; i<queryTermsArray.length; i++){
  File x = new File("C:/meanings/" + queryTermsArray[i] + ".txt");
  if(smallSize == 0){
    smallSize = x.length();
    smallSizeQueryTerm = queryTermsArray[i];
  }
  else if(smallSize > x.length()){
    smallSize = x.length();
    smallSizeQueryTerm = queryTermsArray[i];
  }
}
FileInputStream fin = new FileInputStream(new File("C:/meanings/" + smallSizeQueryTerm +".txt"));
InputStreamReader read = new InputStreamReader(fin, "utf-8");
BufferedReader rd = new BufferedReader(read);
String line;
StringBuffer sb = new StringBuffer();
while ((line = rd.readLine()) != null) {
  sb.append(line);
}
rd.close();
String[] sentence = sb.toString().split(" ");
for(int t=0; t<queryTermsArray.length; t++){
  if(!commonTerms.equals("") && t!=(queryTermsArray.length-1)){
    sentence = commonTerms.split(" ");
    commonTerms = "";
  }
  if(!queryTermsArray[t].equals(smallSizeQueryTerm)){
      FileInputStream fin2 = new FileInputStream(new File("C:/meanings/" + queryTermsArray[t]
+".txt"));
      InputStreamReader read2 = new InputStreamReader(fin2, "utf-8");
      BufferedReader rd2 = new BufferedReader(read2);
      String line2;
      StringBuffer sb2 = new StringBuffer();
      while ((line2 = rd2.readLine()) != null) {
      sb2.append(line2);
      }
      rd2.close();
      sentence2 = sb2.toString();
    for(int j=0; j<sentence.length; j++){
      if(sentence2.contains(sentence[j])){
        commonTerms = commonTerms + " " + sentence[j];
        }

      }
    }
}
String[] commonTermsArray = commonTerms.split(" ");
if(commonTermsArray.length >= 8){
  commonTerms = "";
  for (int x=0; x<8; x++){
    commonTerms = commonTerms + commonTermsArray[x] + " ";
  }
}
else{
```

```
        for(int x=0; x<commonTermsArray.length; x++){
        commonTerms = commonTerms + commonTermsArray[x] + " ";
        }
      }
      return commonTerms;
   }
 public static IndexReader getReader(Directory index) throws IOException{
      return IndexReader.open(index);
   }
   //returns expansion terms for a given query term from all of the terms in
   //the first 10 retrieved relevant document set
   public static String expand(String queryTerm, String termsToSelectFrom) throws Exception{
      String[] termsToSelectFromArray = termsToSelectFrom.split(" ");
      String expandingTerms = "";
      String x = "";
      for(int i=0; i<termsToSelectFromArray.length; i++){
        if(!termsToSelectFromArray[i].isEmpty()  &&  !termsToSelectFromArray[i].contains("C:")  &&
!termsToSelectFromArray[i].contains("0") ){
         int docCount = hitCount("(" + queryTerm + " AND " + termsToSelectFromArray[i] + ")");
         if(docCount >= 1){
          expandingTerms = expandingTerms + termsToSelectFromArray[i] + "-" + docCount + " ";
         }
         else{
          ;
         }
       }
      }
      if(expandingTerms.endsWith(" ")){
        expandingTerms = expandingTerms.substring(0, expandingTerms.lastIndexOf(" "));
      }
      if(expandingTerms.contains(" ")){
        String[] expandingTermsArray = expandingTerms.split(" ");
      for(int t=0; t<expandingTermsArray.length-1; t++){
        for(int j=0; j<expandingTermsArray.length-1; j++){
          String[] rank1 = expandingTermsArray[j].split("-");
          String[] rank2 = expandingTermsArray[j+1].split("-");
          if(Integer.parseInt(rank1[1]) < Integer.parseInt(rank2[1])){
            String temp = expandingTermsArray[j];
            expandingTermsArray[j] = expandingTermsArray[j+1];
            expandingTermsArray[j+1] = temp;
          }
        }
      }
      String[] value = expandingTermsArray[0].split("-");
      for(int j=0; j<expandingTermsArray.length; j++){
        String[] check = expandingTermsArray[j].split("-");
        if(Integer.parseInt(check[1]) != Integer.parseInt(value[1])){
          x = x + expandingTermsArray[j] + " ";
        }
      }
      return x;
      }
    else{
        return expandingTerms;
```

```
    }
  }
public static String order(String expandingTerms){
    String x = "";
    if(expandingTerms.contains(" ")){
      String[] expandingTermsArray = expandingTerms.split(" ");
    for(int t=0; t<expandingTermsArray.length-1; t++){
       for(int j=0; j<expandingTermsArray.length-1; j++){
         String[] rank1 = expandingTermsArray[j].split("-");
         String[] rank2 = expandingTermsArray[j+1].split("-");
         if(Double.parseDouble(rank1[1]) < Double.parseDouble(rank2[1])){
           String temp = expandingTermsArray[j];
           expandingTermsArray[j] = expandingTermsArray[j+1];
           expandingTermsArray[j+1] = temp;
         }
       }
    }
    for(int j=0; j<expandingTermsArray.length; j++){
        x = x + expandingTermsArray[j] + " ";
    }
    return x;
    }
  else{
      return expandingTerms;
    }
 } public static int hitCount(String searchString) throws IOException, ParseException,Exception{
      DefaultListModel mod;
      Query query = null;
      Boolean error = false;
      Hits hits = null;
        mod = new DefaultListModel();
                  Directory directory = FSDirectory.getDirectory("C:/expansion_index");
                  IndexSearcher indexSearcher = new IndexSearcher(directory);
                  QueryParser queryParser = new QueryParser("contents", new AmharicAnalyzer());
        try {
          query = queryParser.parse(searchString);
        } catch (ParseException e) {
          System.err.println("Error parsing query ...");
          e.printStackTrace();
          error = true;
        }
                  if (error == false && indexSearcher != null) {
          try {
            hits = indexSearcher.search(query);
          } catch (Exception e) {
            System.err.println("Error when searching ...");
            e.printStackTrace();
            error = true;
          }
        }
        int hitCount = hits.length();
        if(hitCount>0){
          return hitCount;
        }
```

```
        else{
          return 0;
        }
    }
    //main function to write function and run
        public void assignWeights(WeightingModel QEModel){
                // Set required statistics to the query expansion model
                QEModel.setNumberOfTokens(this.numberOfTokens);
                QEModel.setAverageDocumentLength(this.averageDocumentLength);
                QEModel.setNumberOfDocuments(this.numberOfDocuments);
                // weight the terms
                int posMaxWeight = 0;
                Object[] arr = queryTermsArray.getValues();
                ExpansionTerms1[] allTerms = new ExpansionTerms1[arr.length];
                final int len = allTerms.length;
                for(int i=0;i<len;i++){
                        allTerms[i] = (ExpansionTerms1)arr[i];
            }
                for (int i=0; i<len; i++){
                        try{
                if(allTerms[i].getDocumentFrequency()<= EXPANSION_MIN_DOCUMENTS){
                                        allTerms[i].setWeightExpansion(0);
                                        continue;
                                }
                                double TF = 0;
                                double Nt = 0;
                                lexicon.findTerm(allTerms[i].getTermID());
                                TF = lexicon.getTF();
                                Nt = lexicon.getNt();
                                allTerms[i].setWeightExpansion(QEModel.weight(
                                                allTerms[i].withinDocumentFrequency,
                                                totalDocumentLength, Nt, TF));
                        if(allTerms[i].getWeightExpansion() >allTerms[posMaxWeight].getWeightExpansion())
                                        posMaxWeight = i;

                        } catch(NullPointerException npe) {
                                //TODO print something more explanatory here
        logger.fatal("A nullpointer exception occured while assigning weights on expansionterms at
iteration: "+"i = " + i,npe);
                        }
                }
                // sort the terms by weight
                normaliser = allTerms[posMaxWeight].getWeightExpansion();
                if                      (QEModel.PARAMETER_FREE                         &&
QEModel.SUPPORT_PARAMETER_FREE_QE){
                        normaliser = QEModel.parameterFreeNormaliser(
                                        allTerms[posMaxWeight].getWithinDocumentFrequency(),
                                        numberOfTokens, totalDocumentLength);
                        if(logger.isInfoEnabled()){
                                logger.info("parameter free query expansion.");
                        }
                }
                lexicon.findTerm(allTerms[posMaxWeight].termID);
                if(logger.isDebugEnabled()){
```

```
logger.debug("term with the maximum weight: " + lexicon.getTerm() + ", normaliser: " +
Rounding.toString(normaliser, 4));
                }
                for (int i = 0; i < len; i++){

        allTerms[i].setWeightExpansion(allTerms[i].getWeightExpansion()/normaliser);
                        //expandedTerm[i].normalisedFrequency
terms[i].getWeightExpansion()/normaliser;
                        if (!QEModel.PARAMETER_FREE)
                                allTerms[i].setWeightExpansion(allTerms[i].getWeightExpansion());
                                normalisedFrequency *= QEModel.ROCCHIO_BETA;
                }
        }
        public double getExpansionWeight(int termId, Expander model){
                double score = 0;
                Object o = terms.get(termId);
                if (o != null)
                {
                        double TF = 0;
                        double Nt = 0;
                        lexicon.findTerm(termId);
                        TF = lexicon.getTF();
                        Nt = lexicon.getNt();
                        score = model.score(((ExpansionTerms1)o).getWithinDocumentFrequency(),
                                        TF,
                                        this.totalDocumentLength,
                                        this.numberOfTokens,
                                        this.averageDocumentLength
                                        );
                }
                return score;
        }
        public double getExpansionWeight(double termId){
                Object o = terms.get(termId);
                if (o == null)
                        return -1;
                return ((ExpansionTerms1)o).getWeightExpansion();
        }
        public double getExpansionProbability(double termId) {
                Object o = terms.get(termId);
                if (o == null)
                        return -1;
                return ((ExpansionTerms1)o).getDocumentFrequency() / totalDocumentLength;
        }
        public void insertTerm(double termID, double withinDocumentFrequency) {
                final ExpansionTerms1 et = terms.get(termID);
                if (et == null)
                        terms.put(termID,newExpansionTerms1(termID, withinDocumentFrequency));
                else
                        et.insertRecord(withinDocumentFrequency);
        }
        public void setNumberOfExpansionDocuments(int value) {
                NUMBER_OF_EXPANSION_DOCUMENTS = value;
        }
```

```
public static void main(String[] args) throws IOException,Exception{
  }
}
```

Written code for Statistical co-occurrence based query expansion

```
package amharicir;
/**
 * @author zish
 */
import org.apache.lucene.index.TermEnum;
import org.apache.lucene.index.IndexReader;
import java.io.*;
import org.apache.lucene.queryParser.ParseException;
import org.apache.lucene.queryParser.QueryParser;
import org.apache.lucene.search.Hits;
import org.apache.lucene.search.IndexSearcher;
import org.apache.lucene.search.Query;
import org.apache.lucene.store.Directory;
import org.apache.lucene.store.FSDirectory;
import javax.swing.*;
public class Expander {
  //returns index terms from a given index
  public static String returnIndexTerms(String index) throws IOException{
    StopStemmer stopStemmer = new StopStemmer();
    Directory directory = FSDirectory.getDirectory(index);
    IndexReader indexReader = getReader(directory);
    TermEnum termEnum = indexReader.terms();
    String term = "";
    while(termEnum.next()){
      if (!stopStemmer.isStop(termEnum.term().text())){
      if(!termEnum.term().text().toString().contains("0")
!termEnum.term().text().toString().contains("C:"))
          term = term + termEnum.term().text() + " ";
      }
    }
    return term;
  }
  //returns common terms for the whole query
  public static String returnCommonTerms(String queryTerms)throws Exception{
    String[] queryTermsArray = QueryTokenizer.tokenizeQuery(queryTerms);
    String sentence2 = "";
    long smallSize = 0;
    String smallSizeQueryTerm = "";
    String commonTerms = "";
    //selects a query term with the smallest number of expanding terms
    for(int i=0; i<queryTermsArray.length; i++){
      File x = new File("C:/meanings/" + queryTermsArray[i] + ".txt");
      if(smallSize == 0){
        smallSize = x.length();
        smallSizeQueryTerm = queryTermsArray[i];
      }
      else if(smallSize > x.length()){
        smallSize = x.length();
        smallSizeQueryTerm = queryTermsArray[i];
```

```
        }
      }
      FileInputStream fin = new FileInputStream(new File("C:/meanings/" + smallSizeQueryTerm +".txt"));
      InputStreamReader read = new InputStreamReader(fin, "utf-8");
      BufferedReader rd = new BufferedReader(read);
      String line;
      StringBuffer sb = new StringBuffer();
      while ((line = rd.readLine()) != null) {
        sb.append(line);
      }
      rd.close();
      String[] sentence = sb.toString().split(" ");
      for(int t=0; t<queryTermsArray.length; t++){
        if(!commonTerms.equals("") && t!=(queryTermsArray.length-1)){
          sentence = commonTerms.split(" ");
          commonTerms = "";
        }
        if(!queryTermsArray[t].equals(smallSizeQueryTerm)){
            FileInputStream fin2 = new FileInputStream(new File("C:/meanings/" + queryTermsArray[t]
+".txt"));
            InputStreamReader read2 = new InputStreamReader(fin2, "utf-8");
            BufferedReader rd2 = new BufferedReader(read2);
            String line2;
            StringBuffer sb2 = new StringBuffer();
            while ((line2 = rd2.readLine()) != null) {
            sb2.append(line2);
            }
            rd2.close();
            sentence2 = sb2.toString();
        for(int j=0; j<sentence.length; j++){
          if(sentence2.contains(sentence[j])){
            commonTerms = commonTerms + " " + sentence[j];
            }

          }
        }
      }
      String[] commonTermsArray = commonTerms.split(" ");
      if(commonTermsArray.length >= 8){
        commonTerms = "";
        for (int x=0; x<8; x++){
          commonTerms = commonTerms + commonTermsArray[x] + " ";
        }
      }
      else{
        for(int x=0; x<commonTermsArray.length; x++){
        commonTerms = commonTerms + commonTermsArray[x] + " ";
        }
      }
      return commonTerms;
    }
    //reads index
    public static IndexReader getReader(Directory index) throws IOException{
      return IndexReader.open(index);
```

```
    }
    //returns expansion terms for a given query term from all of the terms in
    //the first 10 retrieved relevant document set
    public static String expand(String queryTerm, String termsToSelectFrom) throws Exception{
      String[] termsToSelectFromArray = termsToSelectFrom.split(" ");
      String expandingTerms = "";
      String x = "";
      for(int i=0; i<termsToSelectFromArray.length; i++){
        if(!termsToSelectFromArray[i].isEmpty()  &&  !termsToSelectFromArray[i].contains("C:")  &&
!termsToSelectFromArray[i].contains("0") ){
          int docCount = hitCount("(" + queryTerm + " AND " + termsToSelectFromArray[i] + ")");
          if(docCount >= 1){
           expandingTerms = expandingTerms + termsToSelectFromArray[i] + "-" + docCount + " ";
          }
          else{
           ;
          }
        }
      }
      if(expandingTerms.endsWith(" ")){
        expandingTerms = expandingTerms.substring(0, expandingTerms.lastIndexOf(" "));
      }
      if(expandingTerms.contains(" ")){
        String[] expandingTermsArray = expandingTerms.split(" ");
      for(int t=0; t<expandingTermsArray.length-1; t++){
        for(int j=0; j<expandingTermsArray.length-1; j++){
          String[] rank1 = expandingTermsArray[j].split("-");
          String[] rank2 = expandingTermsArray[j+1].split("-");
          if(Integer.parseInt(rank1[1]) < Integer.parseInt(rank2[1])){
            String temp = expandingTermsArray[j];
            expandingTermsArray[j] = expandingTermsArray[j+1];
            expandingTermsArray[j+1] = temp;
          }
        }
      }
      String[] value = expandingTermsArray[0].split("-");
      for(int j=0; j<expandingTermsArray.length; j++){
        String[] check = expandingTermsArray[j].split("-");
        if(Integer.parseInt(check[1]) != Integer.parseInt(value[1])){
          x = x + expandingTermsArray[j] + " ";
        }
      }
      return x;
      }
     else{
        return expandingTerms;
      }
    }
    //orders expansion terms according to their similarity with a query term
    public static String order(String expandingTerms){
      String x = "";
      if(expandingTerms.contains(" ")){
        String[] expandingTermsArray = expandingTerms.split(" ");
      for(int t=0; t<expandingTermsArray.length-1; t++){
```

```
        for(int j=0; j<expandingTermsArray.length-1; j++){
          String[] rank1 = expandingTermsArray[j].split("-");
          String[] rank2 = expandingTermsArray[j+1].split("-");
          if(Double.parseDouble(rank1[1]) < Double.parseDouble(rank2[1])){
            String temp = expandingTermsArray[j];
            expandingTermsArray[j] = expandingTermsArray[j+1];
            expandingTermsArray[j+1] = temp;
          }
        }
      }
      for(int j=0; j<expandingTermsArray.length; j++){
          x = x + expandingTermsArray[j] + " ";
      }
      return x;
      }
    else{
        return expandingTerms;
      }
  }
  //counts the number of relevant documents found for a given search string
  public static int hitCount(String searchString) throws IOException, ParseException,Exception{
        DefaultListModel mod;
        Query query = null;
        Boolean error = false;
        Hits hits = null;
          mod = new DefaultListModel();
                    Directory directory = FSDirectory.getDirectory("C:/expansion_index");
                    IndexSearcher indexSearcher = new IndexSearcher(directory);
                    QueryParser queryParser = new QueryParser("contents", new AmharicAnalyzer());
          try {
            query = queryParser.parse(searchString);
          } catch (ParseException e) {
            System.err.println("Error parsing query ...");
            e.printStackTrace();
            error = true;
          }
                    if (error == false && indexSearcher != null) {
            try {
              hits = indexSearcher.search(query);
            } catch (Exception e) {
              System.err.println("Error when searching ...");
              e.printStackTrace();
              error = true;
            }
          }
          int hitCount = hits.length();
          if(hitCount>0){
            return hitCount;
          }
          else{
            return 0;
          }
  }
  //main function to write function and run
```

```
    public static void main(String[] args) throws IOException,Exception{
    }
}
```

Written code for Bi-gram based query expansion

```
package amharicir;
/**
 * @author zish
 */
public class Bi_GramExpander {
    public static String bi_GramExpand(String query, String docs) throws Exception{
        String[] queryTerms = QueryTokenizer.tokenizeQuery(query);
        String firstQueryTerm = queryTerms[0];
        String expandingTerms = "";
        String finalExpandingTerms = "";
        AmharicStemmer stemmer = new AmharicStemmer();
        String[] doc = docs.split(" ");
        for(int t=0; t<queryTerms.length; t++){
        if(!queryTerms[t].equals(firstQueryTerm)){
            expandingTerms = expandingTerms + "===";
        }
        finalExpandingTerms = "";
        String tempExpandingTerms = Bi_gramThesaurusMaker_1.getExpansionTerms(queryTerms[t], doc);
        String[] expandingTermsArray = tempExpandingTerms.split(" ");
            finalExpandingTerms = finalExpandingTerms + queryTerms[t] + "-" + 100 + " ";
            for(int j=0; j<expandingTermsArray.length; j++){
                double probability =
Bi_gramThesaurusMaker_1.getProbability(expandingTermsArray[j],tempExpandingTerms);
                if(probability > 0.0){
                    if(!finalExpandingTerms.contains(expandingTermsArray[j]))
                        finalExpandingTerms = finalExpandingTerms + expandingTermsArray[j] + "-" + probability
+ " " ;
                }
            }
            System.out.println("order is called");
            String fExpandingTerms = Expander.order(finalExpandingTerms);
            if(fExpandingTerms.endsWith(" ")){
                fExpandingTerms = fExpandingTerms.substring(0, fExpandingTerms.lastIndexOf(" "));
            }
            expandingTerms = expandingTerms + fExpandingTerms + " ";
            String[] separateExpandingTerms = fExpandingTerms.split(" ");
            String separateTerms="";
            for(int a=0; a<separateExpandingTerms.length; a++){
                String[] term = separateExpandingTerms[a].split("-");
                separateTerms = separateTerms + term[0] + " ";
            }
            if(separateTerms.endsWith(" ")){
                separateTerms = separateTerms.substring(0,separateTerms.lastIndexOf(" "));
            }
            if(expandingTerms.endsWith(" ")){
                expandingTerms = expandingTerms.substring(0,expandingTerms.lastIndexOf(" "));
            }
            expandingTerms = expandingTerms + " --- " +separateTerms;
```

```
        }
        return expandingTerms;
    }
    //returns 13 expansion terms by selecting equal amount from each query term expansion terms
    public static String getFinalExpansionTerms(String expansionTerms, String reformulatedQuery){
        String tempReformulatedQuery = reformulatedQuery;
        String finalExpansionTerms = "";
        int alternateNoOfTermsToSelect = 0;
        String[] separation1 = expansionTerms.split("====");
        int noOfTermsToSelect = 13/separation1.length;
        for(int i=0; i<separation1.length; i++){
            finalExpansionTerms = "";
            String[] separation2 = separation1[i].split("---");
            if(!separation2[1].contentEquals(" ")){
                String[] termsToSelectFrom = separation2[1].split(" ");
            if(termsToSelectFrom.length >=noOfTermsToSelect){
                alternateNoOfTermsToSelect = noOfTermsToSelect;
            }
            else{
                alternateNoOfTermsToSelect = termsToSelectFrom.length;
            }
            for(int j=0; j<alternateNoOfTermsToSelect; j++){
                finalExpansionTerms = finalExpansionTerms + termsToSelectFrom[j] + " ";
            }
              String[] finalExpansionTermsArray = finalExpansionTerms.split(" ");
              if(finalExpansionTermsArray.length >= 2){
                tempReformulatedQuery = tempReformulatedQuery.replace(finalExpansionTermsArray[1],
finalExpansionTerms);
              }
            }
        }
        return tempReformulatedQuery;
    }
    //main function to write function and run
    public static void main(String[] args) throws Exception{
    }
}
```

Written code for Bi-gram thesaurus based query expansion

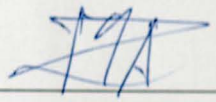
```
package amharicir;
/**
 *
 * @author zish
 */
import java.io.File;
import java.io.*;
import org.apache.lucene.analysis.*;
public class MakeBi_gramThesaurus {
    public static void makeBi_gramThesaurus() throws Exception{
        FileOutputStream fout = new FileOutputStream(new File("c:/Bi_gramThesaurus.txt"));
        OutputStreamWriter rt = new OutputStreamWriter(fout, "utf-8");
        BufferedWriter rtt = new BufferedWriter(rt);
        String terms = Expander.returnIndexTerms("C:/sample_index");
        String[] indexTerms = terms.split(" ");
```

```
for(int i = 0; i< indexTerms.length; i++){
  if(!indexTerms[i].contains(" ") && !indexTerms[i].equals("") && !indexTerms[i].contains("C:") &&
!indexTerms[i].contains("0")){
  rtt.newLine();
  rtt.write(indexTerms[i] + ":--");
  int countk = 0;
  int countk1 = 0;
  double probability = 0.0;
  int breakFactor = 0;
  for(int j=0; j< indexTerms.length; j++){
    if(breakFactor == 1){
      break;
    }
    if(!indexTerms[i].contains(" ") && !indexTerms[i].equals("") && !indexTerms[i].contains("C:")
&& !indexTerms[j].contains("C:") && !indexTerms[i].contains("0") && !indexTerms[j].contains("0") &&
!indexTerms[j].equals("")){
      //rtt.write("|"+indexTerms[i] + " and " + indexTerms[j]+"|");
      String sampleDocs = "C:/sample/1Kings_01.txt C:/sample/Ecclesiastes_06.txt";
      String[] docArray = sampleDocs.split(" ");
      for(int x=0; x<docArray.length; x++){
        FileInputStream fin = new FileInputStream(new File(docArray[x]));
        InputStreamReader read = new InputStreamReader(fin, "utf-8");
        BufferedReader rd = new BufferedReader(read);
        String line;
        StringBuffer sb = new StringBuffer();
        while ((line = rd.readLine()) != null) {
          sb.append(line);
        }
        rd.close();
        String sentence = sb.toString();
        Token token = null;
        AmharicTokenizer tk = new AmharicTokenizer(new StringReader(sentence));
        String tokens = "";
        while((token = tk.next()) != null){
          tokens = tokens + token.term() + " ";
        }
        //String[] tokensArray = QueryTokenizer.tokenizeQuery(tokens);
        String[] tokensArray = tokens.split(" ");
          System.out.println(indexTerms[i] + " and " + indexTerms[j]);
          countk = 0;
        for(int k=0; k<tokensArray.length; k++){
          System.out.println(tokensArray[k]);
          if(tokensArray[k].equals(indexTerms[i])){
            if(k < tokensArray.length-1){
              if(tokensArray[k+1].equals(indexTerms[j]) || tokensArray[k-1].equals(indexTerms[j])){
                System.out.println("yes" + tokensArray[k+1] + " or " + tokensArray[k-1] + " equals "
+ indexTerms[j]);
                countk++;
                System.out.println("countk = " + countk);
                countk1++;
                System.out.println("countk+1 = " + countk1);
                probability = countk1/countk;
                if (probability > 0.5){
                    rtt.write(indexTerms[j] + "-" + probability);
```

```
                        breakFactor = 1;
                        break;
                      }
                    }
                     else{
                      countk++;
                    }
                   }
                  else{
                   if(tokensArray[k-1].equals(indexTerms[j])){
                     countk++;
                     countk1++;
                     probability = countk1/countk;
                     if (probability > 0.5){
                         rtt.write(indexTerms[j] + "-" + probability);
                         breakFactor = 1;
                         break;
                       }
                    }
                     else{
                      countk++;
                    }
                   }
                 }
              }
           }
            * if(countk != 0){
             probability = countk1/countk;
           }
           System.out.print("for" + indexTerms[i] + " and " + indexTerms[j] + " probability = " + probability
+ "with countk = " + countk + " and countk1 = " + countk+1);
           if (probability != 0.0){
             rtt.write(indexTerms[j] + "-" + probability);
           }
            */
           }
        }
     }
     }
    rtt.close();
  }
  public static void main(String[] args) throws Exception{
    final File docDir = new File("C:/sample");
    String txtDocs = "b";
    makeBi_gramThesaurus();
  }
}
```

DECLARATION

This thesis is my original work. It has not been presented for a degree in any other university and all sources of material used for the thesis have been duly acknowledged.



ZELALEM ADDIS ALEMU

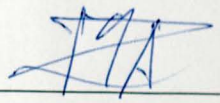
The thesis has been submitted for examination with my approval as university advisors

DEREJE TEFERI (PhD)

February 2014

DECLARATION

This thesis is my original work. It has not been presented for a degree in any other university and all sources of material used for the thesis have been duly acknowledged.



ZELALEM ADDIS ALEMU

The thesis has been submitted for examination with my approval as university advisors

DEREJE TEFERI (PhD)

February 2014