HARAMAYA UNIVERSITY

POSTGRADUATE PROGRAM DIRECTORATE

College: Computing and Informatics

Department: Information Science

Program: Master of Science in Information Science

Course Title: Information Storage and Retrieval Systems

Course Code: INFO 511

Instructor: Dr. Million M. (PhD)

Information Retrieval System for Document Collection Written In Afaan Oromo Language

By

Name	ID
Irandufa Indebu	SGS/0612/11
Getaneh	Sgs/0620/11

Submission Date: 25/02/2019

FEBRUARY 2018

HARAMAYA UNIVERSITY, HARAMAYA

ABSTRACT

Information retrieval enables to search for relevant documents from large corpus as per the information need of users. Afaan Oromo is a Cushitic language spoken today by about 40 million people in Ethiopia. Afaan Oromo text retrieval developed in this study has indexing and searching subsystems. While Indexing organizes index terms, searching enables matching query terms with index terms in order to retrieve relevant documents. For this project study different Afaan Oromo news articles, books and Afaan Oromo websites prepares text document corpus. In addition, various techniques of text preprocessing including tokenization, normalization, stop word removal and stemming were used to identify content-bearing words. Among different model of information retrieval model, vector space model was used to guide searching for relevant document from Afaan Oromo text corpus. The model is selected since Vector space model is the widely used classic model of information retrieval system and it uses non-binary weight. The index file structure used is inverted index file structure.

INTRODUCTION

1.1. Background

Human being has arranged and organized information for later retrieval and usage for just about 4000 years[1]. Since the 1940s, the problem of information storage and retrieval has attracted increasing attention. It is simply stated: we have vast amounts of information to which accurate and speedy access is becoming ever more difficult. One effect of this is that relevant information is ignored since it is never uncovered, which in turn leads to much duplication of work and effort. With the advent of computers, a great deal of thought has been given to using them to provide rapid and intelligent retrieval systems. In libraries, many of which certainly have an information storage and retrieval problem, some of the more mundane tasks, such as cataloguing and general administration, have successfully been taken over by computers. However, the problem of effective retrieval remains largely unsolved[2].

In principle, information storage and retrieval is simple. Let say there is a store of documents and a person (user of the store) formulates a question (request or query) to which the answer is a set of documents satisfying the information need expressed by his question. The user can obtain the set by reading all the documents in the store, retaining the relevant documents and discarding all the others. In a sense, this constitutes 'perfect' retrieval. This solution is obviously impracticable. A user either does not have the time or does not wish to spend the time reading the entire document collection, apart from the fact that it may be physically impossible for him to do so.

To overcome this problem new field called Information Retrieval (IR), which focuses on finding relevant resources from unstructured large corpus that satisfies user information needs, is emerged. IR focuses on retrieving documents based on the content of their unstructured components[2]. An IR request (typically called a “query”) may specify desired characteristics of both the structured and unstructured components of the documents to be retrieved, e.g., “the documents should be about ‘Information retrieval’ and their author must be ‘Smith’.” In this example, the query asks for documents whose body (the unstructured part) is “about” a certain topic and whose author (a structured part) has a specified value [3].

IR system should be both effective and efficient but always increasing decreases the other. So coming up with efficient effective system needs tough task. Additionally the evaluation performance IR system is also challenging task, because performance of IR is evaluated in relative to relevance of retrieved document toward users' query and efficiency. However, it is difficult to identify what is relevant from the irrelevant one, because human information needing behavior is fluctuating.

There are more than 86 languages in Ethiopia. Afaan Oromo is one of the languages with large number of speakers under Cushitic family. As the Ethiopia's statistical report of 2007[4]. Afaan Oromo is a Cushitic language spoken today by about 40 million people in Ethiopia (about 40% of the country's population), in Kenya, Somalia and Djibouti and is the 3rd largest language in Africa after Arabic and Hausa. It is the official language of Oromia regional state of Ethiopia and academic language for primary school of the region. Afaan Oromo language and transfer information as primarily for speakers of language increases from time to time; for Example, Kallacha Oromiyaa, Bariisaa, Yeroo, Oromia Television and Radio (Web news),Voice of America (VOA) (web news), and different academic and recreational medias to mention a few [2]. Development of language is highly related with development of technology. The fact that initiated this study is also enabling development of Afaan Oromo to grow with current information technology support, which focuses on developing Afaan Oromo text retrieval system that enables Afaan Oromo readers to search relevant information need as per their needs.

There are different challenges in implementing IR system. Information retrieval is language dependent process, which needs integrating knowledge of information retrieval techniques and natural language. Most of IR techniques are developed for English language and it is always difficult task applying it for other languages like Afaan Oromo. The other thing is trade-off between efficiency and effectiveness in terms of IR system performance.

1.2.Problem statement

IR is not being optional technology, it is something that is very important to everybody and mandatory to use. In this Information Age, information is highly needed than anything else. Like any other languages, many important Afaan Oromo electronic documents are publishing each days by writers. But finding this important information needs modern technology and system support [5]. Hence, the aim of this study is to develop a prototype for Afaan Oromo text retrieval system that organize document corpus using indexing and search relevant ones as per users query based on vector space model.

1.3.Objectives

1.3.1. General objectives

The general objective of this study is to come up with an Information Retrieval system that can enable to search for relevant Afaan Oromo text corpus.

1.3.2. Specific objectives

- To review related works so as to have a conceptual understanding of the area and Study literatures, techniques and information retrieval models and tools applicable for the development of Afaan Oromo text searching systems.
- To know the unique features of word formations in the Afaan Oromo writing system.
- To apply the suitable text operations and indexing scheme to organizing document corpus.
- To know the importance of vector space model enables to design effective Afaan Oromo text retrieval.
- To design an architecture of Afaan Oromo text information retrieval system that enables searching for Afaan Oromo text document from corpus.
- To evaluate the performance of information retrieval designed using different performance evaluation

1.4.Significance of the project

For the development of one language, technology is very important means. When technology grows in one country, the users also wants the internet to use it with their language. In general, this study is used to develop Afaan Oromo text searching system and gives better conceptual understanding for users who want to know how to design and implement information retrieval system for the given local language. Even if it is narrow project, this work also put stone for the future researcher who needs to plays on area.

1.5.Scope of the project

The focus of this study is on designing an information retrieval system that effectively searches with in Afaan Oromo text corpus. The work mainly implements an indexer and searcher from corpus of Afaan Oromo textual documents. To identify content bearing index terms and query terms a series of different text operations are applied. Index terms are organized using inverted index file and searching for documents satisfying query terms are guided by vector space model.

1.6. Limitation of the project

The project also not includes others Afaan Oromo multimedia document collections such as images, audio, video. There is also no query expansion implemented by the system.

2. Literature review

2.1. Overview of information retrieval

Information retrieval (IR) deals with the representation, storage, organization of, and access to information items''. The representation and organization of the documents should provide the user with easy access to the information in which he/she is interested. Unfortunately, characterization of the information need of users is not a simple task.

Information retrieval is defined as the process of finding/searching documents of an unstructured nature that satisfies information need of users from within large collection [6]. It returns documents that are relevant to users query rather than explicit answer to their information need. In any information retrieval system documents collection are classified into two groups. The first class of document is those, which satisfies users' information needs also called relevant documents,

whereas documents, which are not satisfying users' information need, called irrelevant documents. In fact, there is no perfect information retrieval system, which retrieves all relevant documents, and not retrieves non relevant document [7].

2.2. Subsystem of information retrieval

Information Retrieval has two main subsystems, Indexing and Searching. Indexing is an offline process of representing and organizing large document collection using indexing structure such as Inverted file, sequential files, suffix array and signature file to save storage memory space and speed up searching time. Searching is the process of relating index terms to query terms and return relevant hits to users query. Both indexing and searching are interrelated and dependent on each other for enhancing effectiveness and efficiency.

2.2.1. Indexing

Is language dependent process which varies from language to language [8]. Indexing tries to represent documents in terms of the language of the information retrieval system. The language of information retrieval is word. The function of any IR system is to process a user request for information and retrieve documents that have contents that could potentially satisfy the information need of the user. With the ever-increasing volume of text information stored in electronic media, searching for full texts becomes more and more time consuming and uncontrollable. One technique to overcome representation problem coming with the ever-increasing volume of text in electronic format is indexing. Keywords or terms that are considered as appropriate content descriptors are selected and assigned to documents to provide short form descriptions of the documents. Indexing is the process of analyzing text and deriving such short form descriptions for a document which together sum up the message of the document. Therefore, the purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query.

2.2.1.1. Index structure.

To generate index term from document collection one should first select which indexing structure is appropriate for this purpose. There are different indexing structures available. These includes but not limited to sequential file, inverted index, suffix tree, array signature and etc. these indexing structure is selected based on some criteria such as running time, storage space requirements, updating time required (incremental index or re-indexing) , access capability to each word. From the above-mentioned indexing structure Inverted index, fulfill almost all criteria.

2.2.1.2. Inverted Index

There an inverted index is an index data structure storing a mapping from content, such as words or numbers, to its locations0 in a document or a set of documents. As its name indicates inverted means organizing the documents using the word, they contain. The purpose of an inverted index is to allow fast full text searches to the relevant document, at low cost of increased processing when a document is added to the corpus. Inverted files allows incremental index, which used for updating the inverted index without affecting the current index terms. Generally, inverted index has two files vocabulary files and posting files. Having vocabulary file is used for saving storage space requirement, speeding up running time and searching. Inverted index specifies the location of each terms. Having information about the location is used to know where the terms appear in documents. If the location of each term within the documents across the collection is known, the retrieval process is speed up.

2.2.1.3.Advantage of inverted index over the other.

To have vocabulary files which stores distinct terms. Having vocabulary file is used for avoiding term repetition. Therefore, if the term is not repeated the memory space required to store index term is saved. It is also possible to optimize storage space requirement. If we use inverted index file structure one can store vocabulary file on hard disk, posting file on flash drive and actual documents on cloud storages.

When we construct inverted index, the statistical properties of each term should be defined. These statistical properties of the terms includes the frequency of occurrence of the term in the given document(TF), the frequency of occurrence of the terms within the document (DF) and the

frequency of occurrence of the terms across the collection (CF). Having information about the frequency of the term is used to know the extent to which the given terms important to represent the given documents. As the term is frequently occurring in the whole collection, the term is not important for representing that document. Frequency of the each term in the documents is useful for calculating term weighting (tf*idf). Inverted index also used to speed up running time and searching to the document collection. Similarity measurement of the document and query are working based on term weighting. Generally having weight of the given term simplifies the way we are identifying the relevant document from the corpus for the user query.

2.2.2. Searching

Unlike indexing, which is an offline process, searching is an online process that scans through document corpus to find relevant documents as per the information need of the users. Whatever we apply for indexing to identify index terms, we also apply same text operation on query to identify query terms.

2.2.3. Building Term Vectors in Document Space

One common approach to document representation and indexing for statistical purposes is to represent each textual document as a set of terms [3]. Most commonly, the terms are words extracted automatically from the documents themselves, although they may also be phrases, n-grams, or, manually assigned descriptor terms. if the terms are words extracted from the documents, “stop” words (i.e., “noise” words with little discriminatory power) are eliminated, and the remaining words are stemmed so that only one grammatical form (or the stem common to all the forms) of a given word or phrase remain . We can apply this process to each document in a given collection, generating a set of terms that represents the given document. If we then take the union of all these sets of terms, we obtain the set of terms that represents the entire collection. This set of terms defines a “space” such that each distinct term represents one dimension in that space. Since we are representing each document as a set of terms, we can view this space as a “document space” [3].

We can then assign a numeric weight to each term in a given document, representing an estimate (usually but not necessarily statistical) of the usefulness of the given term as a descriptor of the given document, i.e., an estimate of its usefulness for distinguishing the given document from

other documents in the same collection. It should be stressed that a given term may receive a different weight in each document in which it occurs; a term may be a better descriptor of one document than of another.

An important question is how weights are assigned to terms either in documents or in queries. A variety of weighting schemes have been used. Given a large collection, manual assignment of weights is very expensive. The most successful and widely used scheme for automatic generation of weights is the "term frequency * inverse document frequency" weighting scheme, commonly abbreviated "tf*idf". The "term frequency" (tf) is the frequency of occurrence of the given term within the given document. Hence, tf is a document-specific statistic; it varies from one document to another, attempting to measure the importance of the term within a given document. By contrast, inverse document frequency (idf) is a "global" statistic; idf characterizes a given term within an entire collection of documents. It is a measure of how widely the term is distributed over the given collection, and hence of how likely the term is to occur within any given document by chance. The *i*df is defined as "ln (N/n)" where N is the number of documents in the collection and n is the number of documents that contain the given term

Computing the weight of a given term in a given document as tf*idf says that the best descriptors of a given document will be terms that occur a good deal in the given document and very little in other documents. Similarly, a term that occurs a moderate number of times in a moderate proportion of the documents in the given collection will also be a good descriptor. Hence, the terms that are the best document descriptors in a given collection will be terms that occur with moderate frequency in that collection. The lowest weights will be assigned to terms that occur very infer-quently in any document (low-frequency documents), and terms that occur in most or all of the documents (high frequency documents). Generally, each terms in the document collection and query are represented by their numerical term weight.

2.2.4. Computation of Similarity between Document and Query

Once vectors have been computed for the query and for each document in the given collection. The documents can then be ranked according to how similar they are to the query, i.e., the highest ranking document is the document most similar to the query, etc. While it would be too much to hope that ranking by similarity in document vector space would correspond exactly with human

judgment of degree of relevance to the given query, the hope (borne out to some degree in practice) is that the documents with high similarity will include a high proportion of the relevant documents, and that the documents with very low similarity will include very few relevant documents [3].

Ranking of course, allows the human user to restrict his attention to a set of documents of manageable size, e.g., the top 5 documents, etc. The usual similarity measure employed in document vector space is the "inner product" between the query vector and a given document vector. The inner product between a query vector and a document vector is computed by multiplying the query vector component (i.e., weight), QT*i* for each term i, by the corresponding document vector component weight, DT*i* for the same term *i*, and summing these products over all *i*

2.3.Brief on Afaan Oromo

Afaan Oromo is among the major languages that are widely spoken and used in Ethiopia (Abera, 1988). It is considered one of the five most widely spoken languages from among the roughly one thousand languages of Africa (Grage et al., 1982).

Afaan Oromo uses Qubee (Latin based alphabet) that consists of twenty-nine basic letters of which five are vowels, twenty-four are consonants, out of which five are pair letters and fall together (a combination of two consonant characters such as 'ch', 'dh', 'ny', 'ph'). The Afaan Oromo alphabet characterized by capital and small letters as in the case of the English alphabet. In Afaan Oromo language, as in English language, vowels are sound makers and are sound by themselves. These vowels includes 'a', 'e', 'u', 'i', 'o' like English vowels. Vowels in Afaan Oromo are characterized as short (ex: a) and long (Ex: aa) vowels.

The basic alphabet in Afaan Oromo does not contain 'p', 'v' and 'z'. This is because there are no native words in Afaan Oromo that formed from these characters.

Punctuation marks used in both Afaan Oromo and English languages are the same and used for the same purpose with the exception of apostrophe. Apostrophe mark (') in English shows possession but in Afaan Oromo it is used in writing to represent a glitch (called hudhaa) sound. It plays an important role in the Afaan Oromo reading and writing system. For example, it is used to write the word in which most of the time two vowels

are appeared together like “har’a” to mean (“today”) with the exception of some words like “du’a” to mean “die” which is identified from the sound created.

3. Methodology

To get an understanding of the various techniques to design an information retrieval system for documents written in Afaan Oromo language, relevant published documents, materials on the Internet and journal articles are reviewed.

3.1. Software and language used

Personal computer and the programming language Python 2.7.2 were used to implement Afaan Oromo text retrieval system. Python is incredibly efficient language that the programs do more in fewer lines of code than many other languages can require. Python’s syntax is also used to write “clean” code and the code is easy to read, debug, and extend and build upon compared to other languages.

3.2.1. Invert file construction for Afaan Oromo document

The process of construction of inverted index follows the two critical steps [8]: primarily, collecting document to be indexed, secondly, applying text operation on each documents finally index the documents that each term occurs in by creating an inverted index, consisting of a directory and postings. Thus, 10 Afaan Oromo document collection is collected from different article and different site as the fist step and each of which is stored with their document identification.

Index term selection steps

Tokenization

Tokenization is the process generating the list of words from the document corpus. It is the way of chopping on white spaces and throwing away punctuation characters. For example, if the original document contain the sentence Itoophiyaan biyya sabaaf sablammoonii afaan garaagaraa haasawan 80 ol of keessaa qabdi; the tokens will ‘Itoophiyaan’, ‘biyya’, ‘sabaaf’, ‘sablammoonii’, ‘afaan’, ‘garaagaraa’, ‘haasawan’, ‘80’, ‘ol’, ‘of’, ‘keessaa’, ‘qabdi’. So we can define token as an instance of a sequence of characters. Each such token is now a candidate for an

index entry, after further processing. One of challenges related to tokenization is differentiating single word from compound word.

Normalization

Normalization is the process of handling problem related with variation cases (UPPER CASE, or lower Case or Mixed Cases). Therefore, it is about standardizing the artificial difference among word in the given document. The good way to handle this problem is converting the whole document in to similar case. Often it is best to lowercase everything, since most of the time users use lowercase regardless of correct capitalization. Here are some examples: Republican vs. republican, John vs. john vs. JOHN, etc. In some languages like Amharic, which does not have a distinction between upper, and lower case, this might not be a big deal. However, it is very important for languages using Latin characters [10].

Stop-Words

Stop-words are most frequent terms, which are common to every document, and have no discriminating power one document from the other. So these terms should not be considered in indexing process. This is normally done by comparing the input text with a 'stop list' of words, which are to be removed.

Stemming

Stemming is process used in most search engines and information retrieval systems. It is core natural language processing technique for efficient and effective IR system. Generally stemming transforms inflated words in to their most basic form. There are different stemming algorithms but the most common one is that of Porter, called Porter Stemmer. Even if stemming is very similar to lemmatization in most of indexing process, stemming was used. There are various stemming strategies developed for different purposes [9]. Some stemming algorithms utilize a stem dictionary and others a suffix list.

Many stemming algorithms, whose purpose is to improve IR performance, do not use a stem dictionary, but an explicit list of suffixes, and the criteria for removing suffixes. In this project, for removing suffix from each documents for Afaan Oromo document collection, the second method

(suffix list) have been used. This was done by identifying all suffix for all index terms and then applying rule-based algorithm.

Term weighting

Not all word in Afaan Oromo document collection is equally important to represent each document. Thus to identify which word is important to calculate the extent to which the given term important to represent the given documents. This was done by assigning numerical weight to each term in the document. The weight of the term is calculated based on the statistical information of that term.

Searching

Same text operation was applied on searching component of Afaan Oromo information retrieval system to select query terms.

4. Overview of Afaan Oromo text information retrieval system

Development of IR system involves various techniques and methods. Designing an information retrieval system involves two main components that are indexing and searching. So in Information retrieval system document corpus is needed for evaluation of the system. Document collected from different news articles and other online resources passes through different procedures in order to index and use the retrieval system.

Tokenization: at this level all words with in a document is identified listed. *Stop word removal:* stop word removal is removing words that are repeatedly available. *Stemming:* there are words with prefix and suffix is, so suffix and prefix is removed at this level. *Term weighting:* Different index terms have varying importance when used to describe document contents. This effect is captured through the assignment of numerical weights to each index term of a document. There are different index terms weighting methods (TF, DF, CF) based on which TF*IDF weight can be calculated during searching.

At searching level the stage than can be applied to query are stemming and term weighting. There is different term weighting mechanism those Vector-Space Model, Similarity/Dissimilarity Measures. First queries are stemmed to its root word then using term weighting mechanism the document containing the query will be retrieved according to their relevance

4.1. General Architecture of Proposed Afaan Oromo IR System

Development of IR system involves various techniques and methods. It is known that any Information retrieval system involves two main components that is indexing and searching. The architecture of information retrieval for Afaan Oromo document collection is designed as the following diagram.

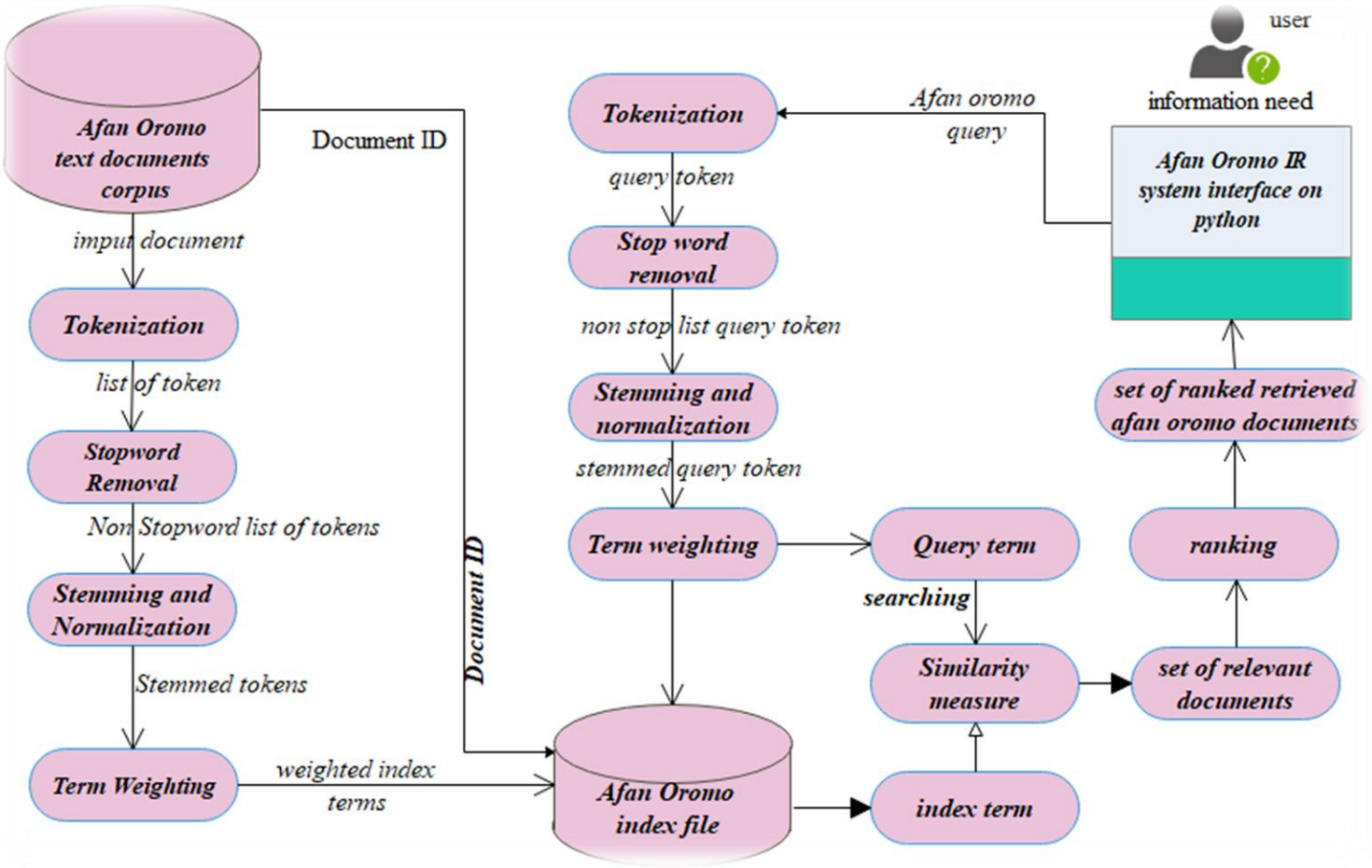

Fig1. Afaan Oromo Text Retrieval System

4.2. How it works?

Given Afaan Oromo text corpus, the IR system organizes those using index file to enhance searching. Thus, first it applies the text operation to select content bearing index terms from afaan Oromo text collection. This text operation includes tokenization of the text words in which attempt to identify stream of tokens (or terms). Then after all words listed from each documents the next step is to apply stop word removal, which attempt to remove the most frequently occurring words from each documents. Since these, stop word occurs in many documents collection, they have no discriminatory power to represent the document. To do so, the system applies stop word removal algorithm by checking the stop word list constructed as the common methodology.

This is followed by normalization in order to bring together similar word written with different punctuation marks and variation cases (UPPER, lower or mixed).

Primarily every term in the document should converted in to similar case format in this study lower case. Lastly, Content bearing terms (non-stop-words) are stemmed. Word with similar meaning should reduce to their root or stem.

For all stemmed tokens, their numerical weights are calculated. These weight of terms are used for identifying which terms is more important than others to represent the given document since each word in each documents is not equally important to represent each documents. There are different term weighting techniques available including binary weights and non-binary weight. For this system, non-binary term weighting techniques that attempt to consider partial matching have been used. Binary weight is not that much important to calculate the weight of the term in the documents. Because it gives, one (1) weight for any terms appears in document and zero (0) for the term not appears in the documents. After the weight each terms in Afaan Oromo documents calculated, inverted index file is constructed.

On the searching similar text pre-processing (tokenization, normalization, stop-word removal, and stemming) technique is followed as it was done in the indexing part.

The system applies partial matching and best matching for identifying the relevant documents for the users information needs. Then, Similarity score is measurement a technique (cosine similarity) used to calculate how the documents nears to the given query terms.

Ranking algorithm is now applied based on cosine similarity between documents and query to order the relevant document before displaying it for the user. After the ranking algorithm is applied, each relevant document are displayed according to the order of similarity score computed for them by using cosine similarity. Now the top ranked document is considered as relevant document for the user.

5. Experimental analysis

5.1. System evaluation

Much of the research and development in information retrieval is aimed at improving the effectiveness and efficiency of retrieval. Evaluation of any information retrieval system is measured by efficiency and effectiveness. Efficiency is usually measured in terms of the computer resources used such as core, backing store, and C.P.U. time. It is difficult to measure efficiency in a machine independent way. In any case, it should be measured in conjunction with effective-ness to obtain some idea of the benefit in terms of unit cost[10]. At the heart of IR, evaluation is the concept of "relevance". Relevance is an inherently subjective concept in the sense that satisfaction of human needs is the ultimate goal, and hence the judgment of human users as to how well retrieved documents satisfy their needs is the ultimate criterion of relevance [11].

Effectiveness is concerned with to what extent the information retrieval system achieved the expected goal. The goal of any information retrieval system is to search/access documents with information that is relevant to the user's information need and helps the user complete a task.

Generally, Cleverdon (1966) listed six criteria that could be used to evaluate an information retrieval system: (1) coverage, (2) time lag, (3) recall, (4) precision, (5) F-measure and (6) user effort. Of these criteria, recall and precision have most frequently been applied in measuring information retrieval [12].

Effectiveness of information retrieval system is commonly measured in terms of precision and recall. *Precision* is the ratio of the number of relevant documents retrieved to the total number of documents retrieved, and *recall* is the ratio of the number of relevant documents retrieved to the total number of relevant documents (both retrieved and not retrieved). Precision is measuring to what extent the system is precise and retrieve only relevant documents.

First the system is tested using one word, then one words, then five words and finally using ten words. From these test the document containing only the query term is retrieved. The result of testing is reviewed bellow as follows:

Test1. Using one query ='oromiyaa' on screen

```
>>>
             ^::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
             ^ ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
             ^                                                                        ^
             ^    BAGA NAGAAN GARA BARBAACHA DOOKUMENTII AFAAN OROMOOTTI DHUFTAN     ^
             ^                          WELCOME TO                                    ^
             ^                                                                        ^
             ^::::::::::::::::::::::    AFAAN OROMOO     ::::::::::::::::::::::::::::: ^
             ^                                                                        ^
             ^                              TEXT                                      ^
             ^                                                                        ^
             ^:::::::::::::::::::::: RETRIEVAL SYSTEM    :::::::::::::::::::::::::::: ^
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

==================================================================================
Please enter your qeury here:
Maaloo waan barbaaduu feetan galchaa:
oromiyaa
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
                                          Dookumentoota  Argaman
                    Search Results
Dokumentoonni armaan gadii akka gaaffi keessanin walfakkaatanitti tartiibaan dhiyaataniiru
The following retrieved documents are ranked According to their nearest to query entered !

                         _________________________________________________________

                                rank 1 :-> Lakkoofsa Galmee|DocumentId: 8
                                rank 2 :-> Lakkoofsa Galmee|DocumentId: 9
                                rank 3 :-> Lakkoofsa Galmee|DocumentId: 7
                                rank 4 :-> Lakkoofsa Galmee|DocumentId: 5
                 ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
             ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^

     ==================================================
 Dubbisuu ni barbaadduu(Are you want to read)?
 Dubbisuuf (To read): 'e'   Xumuruuf (To Stop): 'x' Tuqi(click)
```

Fig2. Screenshot for testing sytem using one query term ('oroiyaa')

The total document containing this term is 4 documents. The result of Recall, Precision and F-Measure is:

Rank	DocID	Relevance	Recall	Precision	F-measure
1	Document8	R	0.25	1	4
2	Document9	R	0.5	1	0.67
3	Document7	R	0.75	1	0.87
4	Document5	R	1	1	1

Test 2 using five query terms

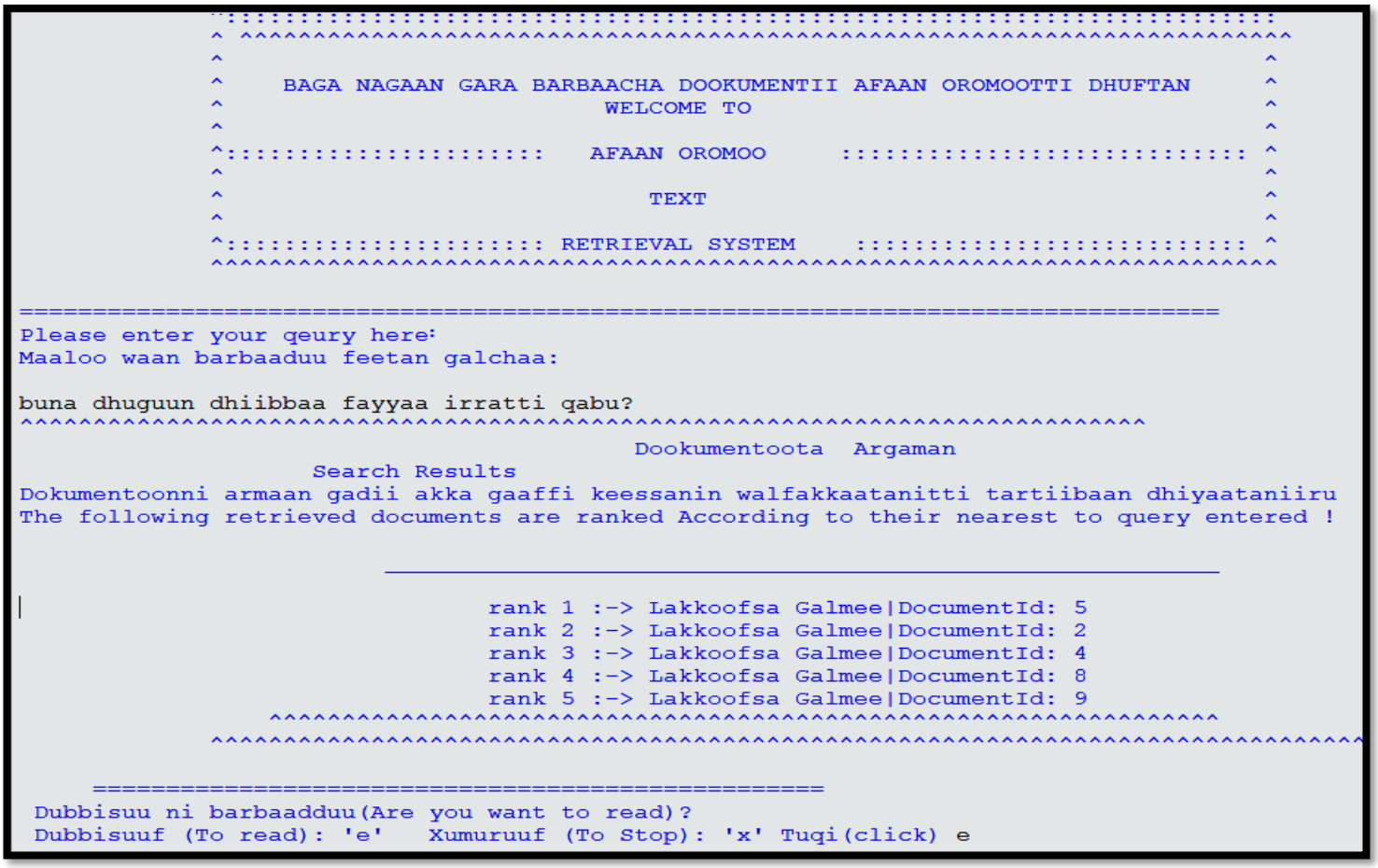

Fig3. Screenshot for testing sytem using five query term (‘bunni oroiyaa keessaa eessatti argama?’

The total document containing this term is five documents. The result of Recall, Precision and F-Measure is

Rank	DocID	Relevance	Recall	Precision	F-measure
1	Document5	R	1/5 =0.2	1/1=1	0.33
2	Document2	R	2/5=0.4	2/2= 1	0.57
3	Document4	R	3/5=0.6	3/3=1	0.75
4	Document8	R	4/5=0.8	4/4=1	0.89
5	Document9	R	5/5=1	5/5/=1	1

Test3. Using 10 query terms

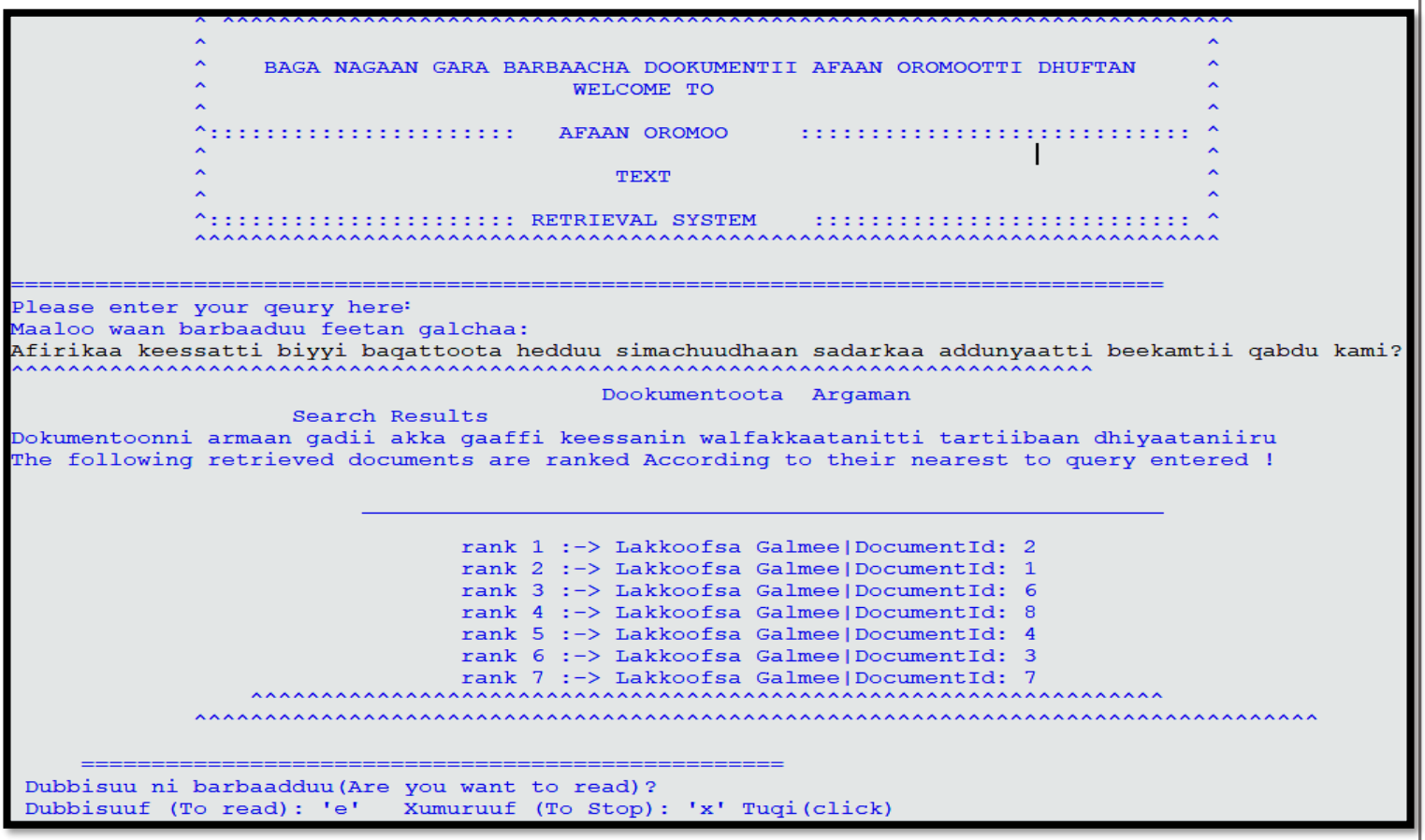

Fig4. Screenshot for testing sytem using 10 query term (‘Afirikaa keessatti biyyi baqattoota hedduu simachuudhaan sadarkaa addunyaatti beekamtii qabdu kami?’)

The total document containing these term is 7 documents. The result of Recall, Precision and F-Measure is:

Rank	DocID	Relevance	Recall	Precision	F-measure
1	Document:2	R	1/7=0.14	1	0.25
2	Document:1	R	2/7=0.29	1	0.45
3	Document:6	R	3/7=0.42	1	0.50
4	Document:8	R	4/7=0.57	1	0.73
5	Document:4	R	5/7=0.71	1	0.83
6	Document:3	R	6/7=0.85	1	0.92
7	Document:7	R	7/7=1	1	1

6. CONCLUSION

The study attempts to develop Afaan Oromo IR system. It performs different activities including generating content bearing index term and query terms by applying text operation. It also cable of applying matching algorithm in both query terms with terms to retrieve relevant document and ranks according to their closeness to query vector. To sum up, this information retrieval system enables users to access and retrieve document collections written in Afaan Oromo language.

However, the developed system has less capability in retrieving relevant document that has word with different structure but same meaning such as ‘hedduu’ (many) and ‘baayyee’ (many).

7. RECOMMENDATION

The future work should give attention on the increasing satisfaction of users in terms of developing query expansion method because it is not included in this system.

The system should also cable to handle polysemy words and synonymy words, which may affect the retrieval performance. Thus, further research should focused on this interesting work.

The future research should also give consideration for Afaan Multimedia documents such as image, audio and video.

8. REFERENCE

[1] B. Baeza-Yates, R. and Ribeiro-Neto, *Modern Information Retrieval*. AddisonWesley New York: ACM Press, 1999.

[2] C. J. van Rijsbergen, *Information Retrieval*. University of Glasgow.

[3] E. Greengrass, "Information Retrieval : A Survey," 2000.

[4] "Summary and Statistical Report of the 2007 Housing Census: Population Size by Age and Sex," Addis Ababa, 2008.

[5] A. Toffler, *The third wave, Progress & Freedom Foundation*. Washington Dc:, 2080.

[6] H. S. C. D. Manning, P. Raghavan, *An Introduction to Information Retrieval*, Online Edi. Cambridge: Cambridge UP, 2009.

[7] D. Hiemstra, *Information Retrieval Models*, Wiley Onli. New york: John Wiley & Sons, Ltd, 2009.

[8] E. Greengrass, *Information Retrieval*. 2000.

[9] E. Airio, "Word Normalization and Decompounding in Mono- and Bilingual IR," vol. 9, no. 3, pp. 1–19, 2004.

[10] C. J. V. R. B. Sc and D. Ph, *INFORMATION RETRIEVAL*. University of Glasgow.

[11] C. Salton, G., Buckley, "Improving retrieval performance by relevance feedback," *J. Am. Soc. Inf. Sci.*, vol. 4, pp. 288–297, 1990.

[12] J. Gwizdka and M. Chignell, "Towards Information Retrieval Measures for Evaluation of Web Search Engines Keywords," 1999.