# Information Retrieval System By Using Vector Space Model

**Naol Bakala**

**Abstract:** All other Major research issues in information retrieval are reviewed and developments in knowledge-based approaches [KBA] are analyzed. It is argued that although a light amount of effort has been done, the efficiency of this approach has yet to be demonstrated. It is recommended that statistical techniques and knowledge-based approaches should be viewed as complementary, rather than competitive. So in this Research work that Vector Space Model of information retrieval system [VSMIS] is used to guide searching for relevant document from Oromiffa text corpus. The model is selected since Vector space model is the widely used classic model of information retrieval system. The index file structure used is inverted index file structure and text document corpus is prepared by the researcher encompassing different news article and experiment is made by using 9(nine) different user information need queries. Various techniques of text pre-processing including tokenization, normalization, stop word removal and stemming are used for both document indexing and query text. The performance the system can be increased if stemming algorithm is improved, standard test corpus is used, and thesaurus is used to handle synonymy words in the language.

**Index Terms:** knowledge-based approaches [KBA], Vector Space Model of information retrieval system [VSMIS]

## 1. Introduction

Ethiopia has more than 80 languages. Cushitic family follows Afaan Oromo language and it has large number of speakers across Ethiopia . Afaan Oromo has more than 25 million speakers in Ethiopia as per the statistical report of 2007 [9]. Afaan Oromo uses Latin based script called —Qubee and it has 26 basic characters. It is the official language of Oromia regional state of Ethiopia and also academic language for primary school of the region. Oromo language, literature and folklore delivered as a field of study in many universities located in Ethiopia and other countries [10]. Nowadays journal, magazines, newspapers, news, online education, books, entertainment Medias, videos, pictures, are available in electronic format both on the Internet and on offline sources. There is huge amount of information being released with this language, since it is the language of education and research, language of administration and political welfares, language of ritual activities and social interaction [11]. Information Retrieval (IR) is one of the major branches of Information Science discipline [1]. The trend in information storage and retrieval can be traced back to 2000 BC when people of Sumerians chose special place to store clay tablets with cuneiform inscription [2]. After they understand their work is efficient on use of information, they developed special categorization system that identifies every tablets and its content. One of the major evolutions in Information Retrieval is invention of print machine in 1450 A.D [3]. A German goldsmith Johannes Gutenberg invented the first movable printer, thousand years later after Chinese invented paper which provides means for disseminating and storing knowledge. Gutenberg's aim was allowing direct access to mass information that was contained in the Bible and other scholarly works. The invention of print machine ignited the Reformation and Renaissance. With the introduction of print machine, print materials are flourished out enormously than ever before. Accessing library materials was a big problem in those days because of increase in library holdings. In order to simplify accessing library collection classification scheme developed include Dewey Decimal Classification (DDC) created by Melvin Dewey in 18761, Library of Congress Classification Scheme (LC) 2, and Universal Decimal Classification (UDC)3. The development of modern IR is highly related to World War II (WWII) in 1945 and cold war after end of the actual war. Because of the difficulty in storing and retrieving large numbers of scientific papers publications during post war, devising new mechanism become mandatory task. In fact most the papers are published for the interest of US military and it should be accessible to them easily. That is why Venavar Bush, Head of National Science Foundation (NSF) defined the problem as Information explosion mass production of information [4]. The challenge of information explosion is finding document relevant to information need users. Bush designed the machine called MEMEX in 1945 to solve the problem of information explosion. MEMEX is —*a hypothetical, desk size information workstation, contained a massive amount of microfilm, all libraries' worth; including purchased material as well as personal documents that would be scanned or penned in*‖ [5:106]. Documents called using their code by rapid selector. The concept of MEMEX is fully realized in modern hypertext of World Wide Web. This is considered as the starting point for modern IR. These days information explosion is even increasing. Facts show that print materials are being replaced by electronic one and accessing a single document of interest is being difficult, because it is unthinkable to check manually each and every possible document available to us [4].

Hans Peter Luhn research engineer at International Business Machine Corporation (IBM) started mechanized punch card based system for searching chemical compounds in 1947. The term Information Retrieval is claimed as it was coined by Calvin Mooers in 1950 for the first time. It is on the —International Conference on Scientific Information‖

• *Naol Bakala, Head/Department of Computer Science, Ambo University, Ethiopia, E-mail: naolbakala@gmail.com*

Washington DC, 1958, that IR system is identified as a solution to retrieval problems. Sooner in 1960 Melvin Earl (Bill) Maron and J. L. Kuhns published ―On relevance, probabilistic indexing, and information retrieval‖ and in 1962 Cyril W. Cleverdon developed a model for IR system evaluation. In 1963, Joseph Becker and Robert Hayes publish ―Information Storage and Retrieval: tools, elements and theories‖. The concept of hypertext is promoted by Theodor Nelson early 1970's. Tim Berniers Lee presented the first proposal on World Wide Web at CERN (European Council for Nuclear Research) in 1989. Late 1990's implementation of web search engines was getting more emphasis [3][5]. Information retrieval is defined as finding documents of an unstructured nature that satisfies information need of users from within large collection [6]. An information retrieval system is system that stores and manages information on documents and also enables users finding the information they need. It returns documents that contain answer to users question rather than explicit answer to their information need. Most of the time retrieved documents satisfy user's information needs. Documents which satisfy users information needs called relevant documents, whereas documents which are not satisfying user's information need are irrelevant documents. In fact there is no perfect information retrieval system which retrieves all relevant documents and no irrelevant document [7]. Hence the aim of this research 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.

## 2. Methodology

Methodology is a way to systematically solve the research problem [8]. This research was conducted in order to figure out challenges of implementing Afaan Oromo retrieval system. Towards achieving the main objective the following step by step procedures are followed.

### 2.1. Literature Review

To have conceptual understanding and identify the gap that is not covered by previous studies different materials, including journal articles, conference papers, books, and the Internet have been reviewed. In this study the review is mainly concerned works that have direct relation with the topic and the objective of the study. These include previous works done on local information retrieval system giving more attention Afaan Oromo IR.

### 2.2. Corpus/Data Set Preparation

The corpus size is 100 text documents written in Afaan Oromo language and with Latin alphabets which is called "Qubee'. The corpus is built from the official website of Oromiya National Regional State , Voice of America Radio Afaan Oromo language 10 , Gumii Waaqeffattoota Addunyaa (GWA) Portal11 , International Bible Society official website12, Oromiya Radio and Television Organization (ORTO) 13 news and other Internet based sources. Data set used is mainly, news articles, others resources from Holy Bible books and the Internet. None of the selected documents are domain specific, rather it covers different aspect of life like sport, culture, socio-economic, political, religious, education and development areas. Heterogeneity of the data set helps evaluation of the system more generic.

### 2.3. Evaluation Techniques

The research involves developing the designed system and evaluating its performance. To this end, corpus is prepared, queries are constructed and relevance judgment is made for evaluating effectiveness of the work to measure the effectiveness of the IR system using score in R Language used. Score is fraction of retrieved documents that are relevant, and recall is fraction of relevant document that retrieved [3]. In this work the interpolated score value will be used to evaluate retrieval effectiveness of the system. To measure performance of the system for multiple queries an average score is calculated.

## 3. Vector Space Model (VSM)

The Vector Space Model (VSM) is a way of representing documents through the words that they contain as a vector. It is a standard technique in Information Retrieval system. The VSM allows decisions to be made about which documents are similar to each other and to keyword queries depending on similarity measurements. It is an algebraic model representing textual information as a vector. These vectors represents importance of a term in *tf * idf weighting* technique, and even absence or presence of the term in the document. Each document is broken down into a word frequency table; that means in to inverted index data structure. When represented on vector space the tables are called vectors and can be stored as arrays of terms. A vocabulary is built from all the words in all documents in the system and it includes dictionary terms, which mean there is no repetition of terms. Each document is represented as a vector based against the vocabulary terms and query terms [7].

***Document indexing*** automatic document indexing is used to create dictionary of terms simply selecting all terms from document and converting it in to a dimension in the vector space. The index file structure used in this study is inverted index and it is discussed in section 3.3. Inverted file index has two files Vocabulary file and Post file. These files are used to build vectors of document versus terms.

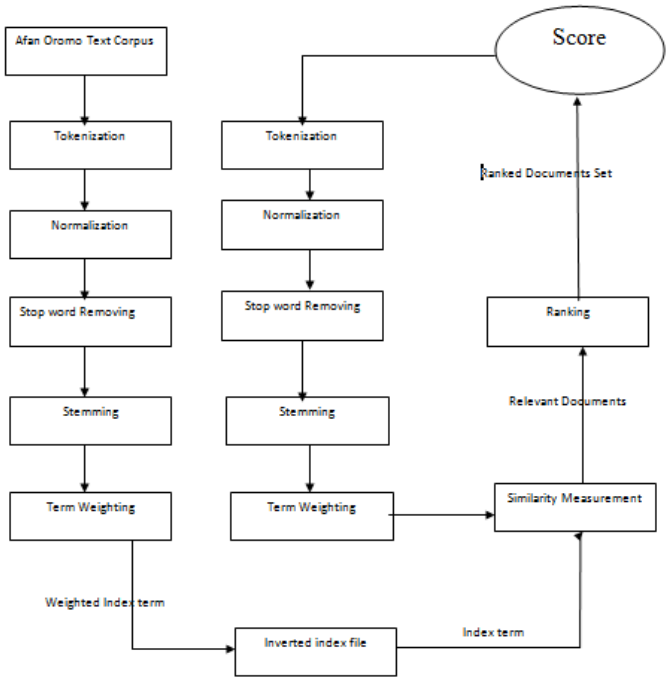
***Term weighting*** The main reason of computing term weight is to assigning weight depending on their level of importance. For every query terms in the query, term document vector is created. Then the weight will be calculated.

$D_i = w_{di1}, w_{di2}, \ldots w_{du}$ $Q = W_{q1}, w_{q2}, \ldots w_{qt}$ w = 0 if a term is absent

***The cosine similarity*** There is many different ways to measure how similar two documents are to each other, or how similar a document is to a query. The cosine measure is a very common similarity measure. Using a similarity measure, a set of documents can be compared to a query and the most similar document returned. If the query term does not occur in the document, similarity measure score should be 0, else the more frequent the query term in the document, the higher the score

(should be). Query and Document similarity is based on length and direction of their vectors.

## 4. Afaan Oromo Text Retrieval Architecture Figure -1




Development of IR system involves various techniques and methods. Information retrieval system designed involves two main components that is indexing and searching. The basic architecture of information retrieval system is depicted in Figure 1 Given Afaan Oromoo text corpus, the IR system organize them using index file to enhance searching. The first step is tokenization of the text words to identify stream of tokens (or terms). This is followed by normalization in order to bring together similar word written with different punctuation marks and variation cases(UPPER ,lower or mixed). The normalized token is checked as it is not stop word. Content bearing terms(non stop words) are stemmed. For all stemmed tokens its respected weight calculated and then 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. Then similarity is measurement techniques (cosine similarity) are used to retrieve and rank relevant documents.

## 5. Experimentation Result :

**Step 1:** Load required packages

```
>library(tm)
> install.packages("tm")
> install.packages("stringr")
Installing package into 'C:/Users/Mine/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/stringr_1.4.0.zip'
Content type 'application/zip' length 215495 bytes (210 KB)
downloaded 210 KB

package 'stringr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Public\Documents\Wondershare\CreatorTemp\RtmpOEZO4s\downloaded_packages
```

```
> install.packages("stringr")
Installing package into 'C:/Users/Mine/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/stringr_1.4.0.zip'
Content type 'application/zip' length 215495 bytes (210 KB)
downloaded 210 KB

package 'stringr' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Public\Documents\Wondershare\CreatorTemp\RtmpoP4ggj\downloaded_packages
> library(stringr)
Warning message:
package 'stringr' was built under R version 3.5.3
> install.packages("qdap")
Error in install.packages : Updating loaded packages

Restarting R session...


> install.packages("qdap")
Installing package into 'C:/Users/Mine/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
also installing the dependency 'tm'

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/tm_0.7-6.zip'
Content type 'application/zip' length 1366609 bytes (1.3 MB)
downloaded 1.3 MB

trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/qdap_2.3.2.zip'
Content type 'application/zip' length 3610552 bytes (3.4 MB)
downloaded 3.4 MB

package 'tm' successfully unpacked and MD5 sums checked
package 'qdap' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Public\Documents\Wondershare\CreatorTemp\RtmpIlRq6m\downloaded_packages
> library(qdap)
Loading required package: qdapDictionaries
Loading required package: qdapRegex
Loading required package: qdapTools
Loading required package: RColorBrewer

Attaching package: 'qdap'

The following object is masked from 'package:base':

    Filter

Warning messages:
1: package 'qdap' was built under R version 3.5.3
2: package 'qdapDictionaries' was built under R version 3.5.2
3: package 'qdapRegex' was built under R version 3.5.3
4: package 'qdapTools' was built under R version 3.5.3
5: package 'RColorBrewer' was built under R version 3.5.2
```

```
> install.packages("slam")
Installing package into 'C:/Users/Mine/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/slam_0.1-45.zip'
Content type 'application/zip' length 208822 bytes (203 KB)
downloaded 203 KB

package 'slam' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Public\Documents\Wondershare\CreatorTemp\RtmpERYHH0\downloaded_packages
> library(slam)
Warning message:
package 'slam' was built under R version 3.5.2
```

**Step 2:** Load all content of data

```
> library("readtext")
Warning message:
package 'readtext' was built under R version 3.5.3
> news_docs = readtext("C:/Users/Mine/Desktop/Publication/P-4/Data-Oromia/*.txt")
> news_list = lapply(news_docs[,2],function(x) genX(x, " [", "]"))
Error in genX(x, " [", "]") : could not find function "genX"
> install.packages("qdap")
Installing package into 'C:/Users/Mine/Documents/R/win-library/3.5'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.5/qdap_2.3.2.zip'
Content type 'application/zip' length 3610552 bytes (3.4 MB)
downloaded 3.4 MB

package 'qdap' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Public\Documents\Wondershare\CreatorTemp\RtmpERYHH0\downloaded_packages
> library("qdap")
Loading required package: qdapDictionaries
Loading required package: qdapRegex
Loading required package: qdapTools
Loading required package: RColorBrewer

Attaching package: 'qdap'

The following object is masked from 'package:base':

    Filter

Warning messages:
1: package 'qdap' was built under R version 3.5.3
2: package 'qdapDictionaries' was built under R version 3.5.2
3: package 'qdapRegex' was built under R version 3.5.3
4: package 'qdapTools' was built under R version 3.5.3
5: package 'RColorBrewer' was built under R version 3.5.2
```

**Step 3 : Load search queries**

```
> search_queries = readtext("C:/Users/Mine/Desktop/Publication/P-4/Data-Oromia/query.txt
",dvsep = "\n")
> search_queries
readtext object consisting of 1 document and 0 docvars.
# Description: data.frame [1 x 2]
  doc_id    text
  <chr>     <chr>
1 query.txt "\"Abiy Ahime\"..."
> queries_list = unlist(strsplit(search_queries[1,2],"\n")
+ )
> queries_list
[1] "Abiy Ahimed" "Afaan"       "Oromiyaati"
> N.query = length(queries_list)
> names(queries_list) = paste0("query", c(1:N.query))
```

**Step 4: Model Creation**

```
> library(tm)
Loading required package: NLP

Attaching package: 'NLP'

The following object is masked from 'package:qdap':

    ngrams


Attaching package: 'tm'

The following objects are masked from 'package:qdap':

    as.DocumentTermMatrix, as.TermDocumentMatrix

Warning messages:
1: package 'tm' was built under R version 3.5.3
2: package 'NLP' was built under R version 3.5.2
> newscorpus = VectorSource(c(news_list,queries_list))
> newscorpus$Names = c(names(news_list),names(queries_list)
+ )
> newscorpus_preproc = Corpus(newscorpus)
> newscorpus_preproc = tm_map(newscorpus_preproc,stripWhitespace)
Warning message:
In tm_map.SimpleCorpus(newscorpus_preproc, stripWhitespace) :
  transformation drops documents
> newscorpus_preproc = tm_map(newscorpus_preproc,removePunctuation)
Warning message:
In tm_map.SimpleCorpus(newscorpus_preproc, removePunctuation) :
  transformation drops documents
> tdm = TermDocumentMatrix(newscorpus_preproc,control = list(weighting = function(x) wei
ghtTfIdf(x, normalize = FALSE)))
> tdm_mat = as.matrix(tdm)
> colnames(tdm_mat) = c(names(news_list),names(queries_list))
> tfidf_mat <- scale(tdm_mat, center = FALSE,scale = sqrt(colSums(tdm_mat^2)))
>
> tfidf_mat <- scale(tdm_mat, center = FALSE,scale = sqrt(colSums(tdm_mat^2)))
>
> query.vectors <- tfidf_mat[, (N.docs + 1):(N.docs+N.query)]
> tfidf_mat <- tfidf_mat[, 1:N.docs]
> doc.scores <- t(query.vectors) %*% tfidf_mat
```

**Step 5: Final Result:**

```
> results.df <- data.frame(querylist = queries_list,doc.scores)
> showTopresults <- function(query){
+     x = results.df[which(results.df$querylist == query),]
+     yy =  data.frame(t(x),rownames(t(x)),row.names = NULL)[-1,]
+     names(yy) = c("score","docs")
+     yy$score = as.numeric(as.character(yy$score))
+     yyy = yy[order(yy$score,decreasing = T),]
+
+     return(yyy[which(yyy$score > 0),][1:3,])
+ }
> showTopresults("Abiy Ahimed")
```

```
Error: unexpected symbol in "showTopresults(""Afaan"
> showTopresults("Afaan")
          score      docs
3     0.4998176 Afaan.txt
NA           NA      <NA>
NA.1         NA      <NA>
> showTopresults("Oromiyaati")
          score           docs
4     0.01765713 Oromiyaati.txt
NA           NA           <NA>
NA.1         NA           <NA>
```

## 6. Conclusion

Text retrieval system is very important for retrieval of textual document. The study attempts to develop Afaan Oromo IR system. The developed prototype has two modules: indexing and searching. The indexing part of the work involves tokenizing, normalization, stop word removal and stemming. Indexing Afaan Oromo text document has common ground with that of the English language because both the language uses Latin alphabets. But Afaan Oromo indexing varies in many different ways. As it has been identified by the study Afaan Oromo has its own grammar (which is called Seerluga) from English. Tokenization of Afaan Oromo is almost similar to the English one except apostrophe is not punctuation mark in Afaan Oromo, rather it is part of words. Normalization of Afaan Oromo documents is also very important. . It has both language dependent and language independent features. The Afaan Oromo stemmer was the only component adopted from previously conducted research. Stemming Oromiffa documents has its own unique procedures, totally different from the English one. Generally speaking, Afaan Oromo document indexing needs its own algorithm and procedure as it is different from other language. The Afaan Oromo text retrieval system developed has also searching components. The main parts are query pre-processing, similarity measurement and ranking. Pre-processing is language dependent process that involves tokenizing, normalization, stop word removal and stemming. The main model is used in this research work is Vector Space Model which is used here for find out measuring and ranking for information retrieved in Afaan Oromo. According to the experiment made the system registered 0.4998176 score in R Language which is promising to design Afaan Oromo information retrieval system that searches within large corpus.

## 7. References


[1] P. Ingwersen, Information Retrieval Interaction, 1st ed. London: Taylor Graham Publishing, 2002.

[2] I. Jukes, ―From Gutenberg to Gates to Google and Beyond: Education For The OnLine World,‖ Singapore, pp. 1 -144, 2005.

[3] V. Bush, ―As We May Think,‖ The Atlantic Monthly, vol. 176, no. 1, pp. 101 -108, 1945.

[4] Bruce R. Schatz, ―Information Retrieval in Digital Libraries: Bringing Search to the Net,‖ Journal of Science, pp. 327-334, vol. 275, 1997

[5] C. D. Manning, P. Raghavan, and H. Schutze, An Introduction to Information Retrieval, Online Edition, Cambridge: Cambridge UP, 2009.

[6] D. Hiemstra, Information Retrieval Models, Wiley Online. New york: John Wiley & Sons, Ltd, 2009.

[7] C. R. Kothari, Research Methodology: Methods and Techniques, 2nd ed. New Delhi: New Age International Ltd., 2004.