**Amharic Information Retrieval System**

**[የአማርኛ ኢንፎርሜሽን ሪትሪቫል ሲስተም]**

**User manual**

**Designed and Programmed by:**

1. **Biniam Asnake GSR/0809/03**
2. **Girma Aweke GSR/0816/03**
3. **Muzeyen Kedir GSR/0823/03**
4. **Negasi G/Egziabher GSR/0303/02**

The Amharic Information Retrieval System has two main modules. These are indexing and searching.

When running the system for the first time, please select the location of stop list, prefix list and suffix list by going to the Indexing Form and clicking on “ማስተካከያ” -> “ዋና ፋይሎችን ማስተካከያ”.

When a user first Double clicks on the “የአማርኛ ኢንፎርሜሽን ሪትሪቫል ሲስተም” icon, search form appears on the screen. The reason the “Search Form” is shown before “Indexing Form” is, users mostly use the “Search Form “once an index file is constructed. But, users can switch from one form to another using the main menu.

* To display the indexing form, click on ፋይል -> ወደ ማውጫ አዘጋጅ from the menu bar.
* Select folder that contains all the files (the corpus) by clicking “…”
* All ‘.txt’ files in the selected folder are selected. The number of files to be indexed – used for calculating IDF and the folder name - used to display the files at last – is stored in “C:\Voc\_Post\Settings.txt”.
* Click on “ማውጫ አዘጋጅ”
* Automatically, the system performs the following operations.
  + - If there is at least one file to be indexed,
      * The location information of Stop list, suffix list and prefix list is read from a file (“C:\Voc\_Post\PrimaryFile.txt”)
      * Stop list, suffix list and prefix list files is read from the file that contains them.
      * Initializes Normalization and SadisReplacer dictionaries by their values.
      * By iterating (looping) through all the files in the selected folder
        + Reads the file using “ReadFromFile” method
        + If the file can be read without any problem, the following preprocessing tasks of information retrieval is conducted.

The following list of punctuations is removed.

"፤", "፣", "፥", "!", ".", ":", "።", "(", ")", "-", "\_", "?", "/", "[", "]", ",", "<", ">", ";", "{", "}", "|", "'" and Double Quote

* The special character "-" is removed in order to consider words like ‘ሥነ-ሥርዓት’ as ‘ሥነሥርዓት’

Normalization

This is: all characters that contain the first element of the dictionary is replaced by the second element.

E.g. "ዐ" will be replaced by "አ" and this holds true for all the elements of "ዐ"

"ጸ" will be replaced by "ፀ"

"ሠ" will be replaced by "ሰ"

"ሐ" will be replaced by "ሀ"

"ኀ" will be replaced by "ሀ"

“ሃ" will be replaced by "ሀ"

"ዉ" will be replaced by "ው"

* The following abbreviations will also be replaced. The following process is not normalization, but it is best and easier to implement it here.

"ዓ.ም" will be replaced by "አመተ ምህረት"

“አአዩ" will be replaced by "አዲስ አበባ ዩኒቨርሲቲ"

“አ.አ" will be replaced by "አዲስ አበባ"

“ክ.ከ" will be replaced by "ክፍለ ከተማ"

"አቢሲኒያ" will be replaced by "ኢትዮጵያ"

"መኪና" will be replaced by "ተሽከርካሪ"

"አስተማሪ" will be replaced by "መምህር"

* + - Tokenize the stop list, suffix list, and prefix list and store them in a data type called “List”.
    - Tokenize the file that is read from the corpus
    - Stop words will be removed

Checks for' words that are not contained in the stop list, words that are not " " (blank), which are not numbers, the length of the word is greater than 1 and puts them in a new array which will be returned

* + - Stemming will be performed by
    - Removing Prefix and
    - Removing Suffix. Here if a word ends with “ች”, it will be removed and the character before it will be replaced by its ‘Sadis’ (6th Character in Amharic alphabet) using the SadisReplacer method.
* After all preprocessing is completed, the system assigns the result to a variable called < ourVocabularyList>

***This operation iterates until there is no more ‘.txt’ file in the selected folder.***

NEXT, calculation and construction of Inverted Index File.

1. Term frequency (TF) will be calculated
2. Collection frequency (CF) will be calculated
3. Location of the index term will be identified and saved
4. Document frequency (DF) will be calculated
5. TF, Document Name and Location will be stored in objPosting and Term ,DF and CF will be stored in objVocabulary

Each element in objPosting will be saved on a file named “**PostingFile.txt**” and element of objVocabulary will be saved on a file named “**VocabularyFile.txt**”

**The searching module works as follows:**

* First the user enters the query and clicks on “ፈልግ” button.
* Then, the contents of ‘VocabularyFile.txt’ and ‘PostingFile.txt’ will be read and stored in a List that contains Structure.
* If stop list, suffix list and prefix list are not initialized, then the system reads from the files that contains them. This will be used to remove stop words from the query and stem it.
* Any punctuation mark will be removed from the query and Normalization will be performed. This operation is the same as the one done in the Indexing module.
* The query term(s) will be tokenized and will be stored in a List data type.
* Stop words in the query will be removed. Here, if all the query terms are stop words, all will be removed. So, “ይቅርታ: ከመጠይቅዎ ጋር የሚስማማ ፋይል አልተገኘም።” message will be displaced. Otherwise the following operations will be performed.
* Prefixes and suffixes will be removed and the stem word will be identified.
* Term frequency will be calculated. The result will be normalized over all the query terms.
* For all query terms, Term weight will be calculated. In order to do this, Document frequency of the query term will be retrieved from the objVocabulary, which contains the retrieved result from the ‘VocabularyFile.txt’.

Then Inverse Document Frequency will be calculated by dividing the Document Frequency by the total number of documents indexed. The Term Weight is calculated as:

termWeight = 0.5 + (0.5 \* tf)) \* idf

The final result will be stored in a List of Structures variable called objTermDocumentMatrix.

* Term weight will be calculated for each of the terms that are extracted from the Vocabulary.
* From the variable ‘objTermDocumentMatrix’, all weights of the query is extracted and stored in a List. This is used for calculating Cosine Similarity.
  1. Then, by iterating through all the data in objTermDocumentMatrix, all weights of a specific document is identified and stored in a list which is used for calculating Cosine Similarity.
  2. After the list of weight of the query and a document is stored and the numerator for the Cosine Similarity function is calculated (as numerator += query.weight \* document.weight), Calculate\_COS\_SIMILARITY is called by passing three arguments, i.e. (listDocument, listQuery, numerator).
  3. The result of the Cosine Similarity is stored in a variable called objResultOfCOS which contains documentName and Cosine Value.
* The operations from i upto iii will be performed for all documents that contain any of the query terms and when EndOfFile is reached or when there is no more document to process, we continue to the next step.
* The contents of the variable called objResultOfCOS will be Sorted first and then Reversed in order to put them in descending order.
* The data that is contained in is displayed on the user interface through the two lists (lstBoxOutput and lstBoxCos).
* When the user selects one document from the list, the text box which is found below the list boxes i.e. ‘txtSampleText’, will be filled by summary data that contains the query terms. The specific query terms will be highlighted to help the user identify the position of the term.
* If the user selects one document name from the list and clicks on the “ፋይሉን ክፈት” button, then the whole content of that specific document is displayed on a new notepad like window where the query terms are highlighted.

**Our Specialty:**

1. Opening the content of a file that is selected from the ranked list.
2. In the opened file, the query terms will be **highlighted**. If multiple terms in a document are found, **ALL** will be highlighted.
3. Provide summary of the document that is selected without the need to open the whole text and highlight the query term
4. Provides users with an easy-to-use interface to change the location of Stop list file, suffix file and prefix file.
5. Normalization of characters
6. For terms which end with the letter “ች”, the character before “ች” will be replaced by its ‘Sadis’ letter.

For example: ሰዎች is stemmed to ሰው, ባንኮች is replaced by ባንክ, …

1. Users can open Vocabulary and Posting file from within the application.
2. Users can switch to Indexing Form or to Searching Form in one click - using the menu provided.
3. Attractive and user-friendly user interface is designed using Visual Basic.Net 2010