International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2018 IJSRCSEIT | Volume 3 | Issue 1 | ISSN : 2456-3307

A Two Steps Approach for Afan Oromo Nonfiction Text Categorization

Naol Bakala Defersha[1], Getachow Mamo[2]

[1]Department of Computer Science, Wollega University school of Graduate study, Post Graduate Coordinator, College of Engineering and Technology, Nekemte, Ethiopia, India

[2]Assistant Professor, Department of Computer Science, Wollega University school of Graduate study, Post Graduate Coordinator, College of Engineering and Technology, Nekemte, Ethiopia, India

ABSTRACT

This study presents Afan Oromo text categorizations which use clustering & classification approaches. In natural language such as Afan Oromo, as amount of text documents in electronic format increases, it become difficult to filter, manage, store and process the desired content of information in natural language text. The solution of this problem is developing a tool that categorizes text documents according to their contents. The aim of this study was to design, and implement Afan Oromo nonfiction text categorization model & examining the application of machine learning techniques for automatic Afan Oromo nonfiction text categorization system. Data was collected from Oromia Culture and Tourism Bureau, Oromo cultural center, online electronic documents and other nonfiction books available. In current study, python programming language applied to tokenize, remove stop words and stem Afan Oromo nonfiction text words whereas R programming language was utilized for document indexing, Normalization, cosine similarity, and preparing documents for machine learning. Weka with java is utilized to split Afan Oromo nonfiction text document data set into train set and test set whereas weka tool was utilized for clustering and classification of Afan Oromo nonfiction texts. By using kmean algorithm Afan Oromo nonfiction text document clustering tasks were performed four times to get classes of documents. Among those clustering tasks, one clustering was resulted in cluster with 8 main categories were obtained as good clusters. J48, NaïveBayes, BayesNet, and SMO classifier algorithms were implemented for training text classification model depending on 8 main classes of documents. Among those classifications algorithms, J48 algorithm shows higher performance 94.3755% and hence it was utilized for constructing classification model. From this work it was possible to conclude that machine learning techniques can be applied for Afan Oromo nonfiction text categorization. Further researches also recommend for Afan Oromo nonfiction text Categorization to upgrade the findings.

Keywords : Afan Oromo, Nonfiction Text, Text Clustering, Text Classification, Natural Language Processing.

I. INTRODUCTION

In information age, users are subjected to continuous flow of information whether or not they actively want it which is resulted in overloaded information (Edmunds and Morris, 2000). Information overload occurs when information users are unable to access relevant information due to voluminous information (Mostak, 2014). In general, overload information has challenges like a difficulty to organize text document according to their topic or contents. Therefore, it is important to use different techniques to solve the problem of overloaded information using text Categorization. One of the techniques that solve the problem of information overload is text categorization.

Text categorization is a system that takes huge texts of a natural language and categorizes them into various clusters based on their relationship. This natural language text is divided and categorized into subsets of text and labeled according to main idea or subject (Faraz, 2015). Text categorization involves Machine learning approaches to solve problem of information

overloaded. Currently, machine learning methods have been mainly used to develop text categorization model. They are applied to develop a model that divides and categorizes a text into its categories. Constructed automatic text categorization model helps to decide and label topical labels to content to solve the problem of overloaded information (Addis, 2010). In addition to machine learning approach, text categorization also uses knowledge engineering. A knowledge engineering approach is that manually defines set of rules for expert knowledge for classification of document into given categories. A knowledge engineering becomes useless comparing it function with function of machine learning approach (Sebastian, 2002).

Machine learning approaches can be categorized as unsupervised and supervised approaches based on training datasets. An unsupervised approach is based on clustering and supervised based on classification. Those approaches are concerned with clustering and classification respectively.

Text clustering is a mechanism that breakdown existing collection of text documents into important clusters (Grace and Desika, 2014). In clustering techniques, more similar clusters are grouped together than in other clusters. It improves efficiency and effectiveness of text categorization system which resulted in saving space, time and increase quality (McCallum et al., 2000). It works with unlabeled texts those are easily available in the world. Text clustering uses algorithms like simple k-means and repeat bisection algorithm

In Text Classification approach, data sets are first manually classified and labeled as predefined categories. Learning algorithm is applied to each category to build classifier. The classifier automatically decides categories of data whose category is unknown. Flat text classification and hierarchical text classification are two main categories of text classification (Addis, 2010). In flat text classification, there is no linkage that defines the relationship of each category as each category is processed separately. Single classifier is trained to categorize a new document to certain classes. On the other hand, hierarchical text classification is used to classify large text documents by using divide-and-conquer approach to overcome a problem of large classification (Sun & Lim, 2001). It decomposes the classification task into a set of simpler problems, one at each node in the classification tree that leads to more accurate classifier. Document classification to their predefined categories requires a large amount of hand labeled texts which is difficult. To fill this gap, using text clustering approach that uses the unlabeled text collections in addition to text classification is important (McCallum et al., 2000).

Therefore, the aim of this study is to develop and implement Afan Oromo nonfiction text categorization model using two steps clustering approach and classification approach. The result of this research has different significances for users of Afan Oromo language text both at organization and individual levels and for Afan Oromo language speakers.

Text categorization tasks need sequences of procedures like collecting datasets (electronic format), preprocessing, representing, applying algorithm to high dimensionality resulted from indexing, and finally using different classifier algorithms and performance measures to create and measure performance of classifier models (Addis, 2010). Since Afan Oromo language is under-resourced language and it is difficult to store, filter, manage and classify documents manually for training dataset, we used nonfiction text categorization for our study.

1.1 Statement of the Problem

Afan Oromo is used as an official language in Oromia regional state so that there are a lot of nonfiction documents that are produced in various organizations of the state. The language is also spoken in northern parts of Kenya and Somalia. As a result, various nonfiction documents that are produced in various organizations are in both hard and electronic formats even though our study only focuses on electronic format. As amount of the text documents in electronic format increases, challenges of identifying relevant documents related to a specific topic increases. This is also true for Afan Oromo language because, as amount of Afan Oromo electronic text documents increase from time to time they become overloaded and accessing, categorizing, organizing and selecting valuable information manually from collection of text documents become difficult. There are also relevant keywords used in irrelevant document and vice versa in

Afan Oromo text documents. Those make searching for Afan Oromo text documents in electronic format more difficult, prone to error, time consuming and tedious. Giving manually given set of electronic document by looking its context is impractical, ineffective, inconsistent and error prone. As a solution, organizing text documents into certain categories according to their content is essential.

Therefore, this work plans to design and implement Afan Oromo nonfiction text categorization system for electronic format of text documents.

II. METHODS AND MATERIAL

2.1 Literature Review

To understand different techniques and algorithm of text categorization, the researcher has reviewed the relevant published articles, research thesis and electronic publications on:-

- ✓ Text clustering techniques, algorithms and its applications.
- ✓ Text classification techniques, algorithms and its application.
- ✓ Machine learning approach.
- ✓ Automatic text document categorization mechanisms.

2.1.1 Data Source and Data-set Preparation

Data sources for this research are Oromia Culture and Tourism Bureau, Oromo cultural center online electronic document and other nonfiction books available. Data obtained from data sources can be available as hard copy and electronic formats. Researchers collect data from Oromia Culture and Tourism Bureau, Oromo cultural center online electronic document and other nonfiction books available (electronic format and hardcopy format). Hard copy format of data collected will be typed and converted into electronic format that is suitable for preprocessing. Data available electronic format is in word format and converted into text (*.txt) format for text preprocessing purpose. Text processing helps for data set preparation from collected data. It will be performed by using Python and R programming language. Python helps to remove unwanted characters and words from collected data whereas R will be utilized for creating document term matrix that is means preparing data for machine learning. Python and R programming language are used in current research due to the researcher is familiarity with them.

2.1.2 Tools

We used Weka (3.8.1) as a tool to design, implement and test Afan Oromo nonfiction categorization models. It was a tool that used for various operations like data preprocessing, attribute selection, classification, clustering and improving the knowledge discovery using different Meta classifiers (Shweta, 2014). Weka tool was selected for this research due to the researchers is familiar with it and it is freely available tool. It is easy to access clustering and classification benefits. Weka tool supports data set in ARFF and CSV formats. In present data research, after data set was loaded into weka tool, text clustering and classifications were performed by Weka tool.

2.2 Scope and Limitation of the Study

This research works was intended to design and implement automatic Afan Oromo nonfiction categorization model depend on collected data. Even if data collected from hardcopy and softcopy resources the experiment in current research was conducted on electronic format of data. Due to the time constraints, not all available text clustering algorithms and text classification algorithms are going to be implemented and tested. Kmeans clustering algorithm is utilized for text clustering and J48, NaiveBayes, BayesNet, and SMO classifier algorithms are tested for text classification. For tokenization of Afan Oromo words white space was used delimiter. This white space is cannot tokenize entire Afan Oromo nonfiction text document words due to some words in Afan Oromo nonfiction text document composed from two independent words. Therefore, in current work, the word composed from two independent words will be considered. In Afan Oromo language only the postfix type of affix is mainly used. Therefore, only postfix of Afan Oromo words were applied in current research for stemming Afan Oromo words.

This study is attempted for single label nonfiction text categorization (not multi-label text categorization) in Afan Oromo language. Besides this, Afan Oromo documents like acronym, abbreviation, audio, scanned

and video documents will not be considered in this work.

2.3 Significance of the Study

The final result of this research has the following significances for the following beneficiaries.

Benefit of the Research:-

- ✓ It results a system used to categorize nonfiction Afan Oromo texts available in electronic format.
- ✓ The final result of this research will be used as an input for further researches that is conducted on Afan Oromo language.
- ✓ The system enables non experts to classify/categorize nonfiction Afan Oromo nonfiction text documents.

Beneficiaries of the Proposed Model

- ✓ All Afan Oromo users for their daily activities.
- ✓ Governmental and non-governmental institutions.
- ✓ Individual user of Afan Oromo nonfiction text documents.

2.4 2.4 Automatic Afan Oromo Nonfiction Text Categorization Techniques

2.4.1 Data Acquisition

As it has been discussed in previous section (1.5.2) sources of data for this research work was Oromia Culture and Tourism Bureau, Oromo cultural center online electronic document and other nonfiction books available. Data was collected from mentioned sources in electronic format and converted to format suitable format for preprocessing tasks. Afan Oromo nonfiction text preprocessing includes tokenization, stemming, and stop word removal. Text documents representation and dimensionality reduction tasks were applied on preprocessed Afan Oromo nonfiction text documents to make data ready for machine learning techniques. After Afan Oromo nonfiction text documents became ready for Machine learning, text clustering and text classification techniques were applied to build Afan Oromo nonfiction categorization model..

2.4.2 Text Preprocessing

Text preprocessing is phases of study implemented to convert raw data in a natural language to the most important text-features that help to identify between text-categories (Chaudhari et al, 2013). It was performed on collected data those were input for text clustering and Classification. It involved set of steps such that one steps done after another step completed to generate important terms and allocates weights that show their importance for representing the document. Before performing text preprocessing task on collected data, we tried to correct spelling error of some Afan Oromo words in collected nonfiction text documents. In Afan Oromo langauge ~~words~~ spelling letters of word wrongly typed has great impacts on current research work. Therefore, researchers attempted correct spelling error words in these documents. For instance, researchers faced with the sentences such as “sirni gaddaa kalaqa uummata oromooti and “Sirni gadaa madda diimokiraasii ammayaati” during data collection for Afan Oromo nonfiction text documents. In “sirni gaddaa kalaqa uummata oromooti” sentence word “gaddaa” wrongly typed. It was corrected to word gadaa by researchers. “Gadaa” is differnegt from gaddaa when they stemmed to stem or root. Gadaa stemmed to “gad” root whereas “gaddaa” stemmed to “gadd” root. The term “gad” and “gad” are used as different terms to build document terms matrix or term document matrix that represent documents. Such problems lead our entire work to low performance. After we walk through entire text documents and correct spelling of words in the text document in word format converted into “*.txt” format that is suitable for data preparation tasks. Here after, Afan Oromo nonfiction text corpus built from sets of documents in “txt” format and researchers performed text preprocessing tasks on this corpus. Text preprocessing tasks were performed on Afan Oromo nonfiction text document corpus to clean and to make it ready for machine learning. Particularly in current research it includes tokenization, stop word removal and stemming. In addition to those processes document representation and dimensionality reduction are processes helps to prepare data for machine learning. Each text categorization tasks need tools and programming language. In current research, Tokenization, stop word removal and stemming performed on Afan Oromo Corpus were implemented by using python programming language whereas

document representation (indexing document) and dimensionality reduction was performed by R programming language.

2.4.3 Tokenization

Tokenization is process of breaking down strings into tokens of words. Khan (2010) defines tokens as elements of string. Strings were disintegrated into words, digits and punctuations in natural language. From tokenized strings tokens are generated. Tokenization was performed by using white space. Afan Oromo phrases, clause and sentence were tokenized by using algorithm adopted from (Abera Driba, 2009; Zelalem, 2001) works with little modification on the length of the word. We used python programming language for the implementation of this algorithm.

Algorithm 3.1 to REMOVE number from Afan Oromo nonfiction text documents.

```
Open the file for processing
Do
    Read the content of the file line by line
    Assign the content to string
    For word in string split by space
        If word contains number
            Replace number marks with space
    End for
While end file
```

Algorithm 3.1 above used to remove numbers from document through sets of steps. Those series of steps are: - first, it opens file to read its contents. Second, it divides the strings to tokens depending on space. Third, check the contents whether it contains digits or not. Fourth, if there is digits delete and replace it by white space and check up to the end of documents.

Algorithm 3.2 to REMOVE Afan Oromo Punctuation marks from Afan Oromo nonfiction text document

```
Open the file for processing
Do
    Read the content of the file line by line
    Assign the content to string
    For word in string split by space
        If word contains punctuation marks
            Replace punctuation marks with space
    End for
While end file
```

Algorithm 3.2 shows how to remove punctuation marks from Afan corpus in current.

Algorithm 3.2 above designed to remove Afan Oromo punctu following sets of steps. First, it opens file to read its contents. Sec check whether content is Afan Oromo punctuation mark or not. F marks replace it by white space and check up to the end of docume

After Afan Oromo digits and punctuation marks were replaced b word identification were performed by using below algorithm 3.3.

Algorithm 3.3 to identify words

1. Initialize the variable to hold the word
2. Read a character from the sentence (document)
3. Check if the character is Afan Oromo word delimiter
4. If not, concatenate the character to the variable,
5. Else if the number of characters is above two characters report th
6. If there is more data to process go to step 1

By using the above algorithm 3.3, the list of Afan Oromo words were generated by walking those following steps. First it initializes variable that holds values. Second, it read character from document starting from the beginning of the document. Third, check whether character if is Afan Oromo word delimiter or not. Fourth, if not, concatenate character to variable. Fifth, if the length of character is greater than or equal to 2 generate it as a word and repeats this process until no data content to be read. Finally, data tokenized and cleaned from digits and punctuation marks were used in python source code that removed Afan Oromo stop words.

2.4.4 Stop Word Removal

There is no standardized stop words list prepared for Afan Oromo nonfiction text document. Therefore, we manually collected Afan Oromo nonfiction text document stop words lists depending on corpus using Afan Oromo dictionaries. Afan Oromo nonfiction text document stop words are the most frequently used words in nonfiction text documents. They carry no information and include pronouns, prepositions, conjunctions, articles, and particles. For instance, stop words of Afan Oromo nonfiction text document are kana, sun, fi, inni, ana, akka, ishee, isaan, nuti and etc. The lists of Afan Oromo nonfiction text documents stop words collected from collected corpus and saved in one file. By using this file name in which stop words were saved, entire stop words imported into python source code, code executed and stop words removed from the corpus.

Here under algorithm stated by (Abera Diriba, 2009) implemented to remove Afan Oromo nonfiction text document stop words.

An algorithm for stop words removal from a given documents.
1. Get the next word until the last word in the document
2. Check the word against the stop words list
3. If not a word exists in the stop words list then write it as a candidate for document representation
4. Else drop it
5. Go to step 1.

After stop words were cleaned from corpus stemming operation was implemented on corpus using python programming language.

2.4.5 Afan Oromo nonfiction word Stemming

In Afan Oromo nonfiction text document, one word appears in different forms to refer singular or plural, and to show tense. This various forms of words may have one root or stem. The aim of Afan Oromo nonfiction text document word stemming is to obtain this root or stem of word. Stemming is process of delete affixes from a given word. Among various approaches of stemming word, Affix removal was used for stemming Afan Oromo nonfiction text document words. Still it confusion that whether Afan Oromo language words have prefix or not. Different persons write one Afan Oromo words in different styles. for example, other person can write as "walgargaaruu", "hindanda'u", "nidanda'ama", "niguddifame", "nigadoome", "hingadoomne" whereas other person write as wal gargaaruu", "hin danda'u", "ni danda'ama", "ni guddifame", "ni gaddome", "hin gadoomne". Writing Afan Oromo words in above two ways prevent researcher to consider other affixes except postfix. In current research, "wal", "ni", "hin" and etc were taken as stop words than as prefixes. Infixes type of affixes are also not known in Afan Oromo language. Therefore, we particularly used postfix removal techniques for stemming Afan Oromo nonfiction text document words. The selection of only postfix removal techniques this due to known affixes in Afan Oromo language is post fix (maxxantuu boodaa). For instance, in Afan Oromo language Gaachana, gaachanaan, gaachanatti, gaachanni, gaachanaaf and etc words are diffent forms of stem "hundee jechaa" Gaachan and postfixes were aan, a, atti, and naaf. Researchers removed Afan Oromo nonfiction postfixes through several steps. Postfixes of each word in Afan Oromo nonfiction text documents corpus were identified and the length of post-fixes to be removed from root words were decided by researchers. We wrote source code of python programming language depending on the identified post-fixes and length the postfixes (see appendix B). Finally, Afan Oromo word stemming tasks completed and the entire word reduced to its stem or root and this stemmed word used for document representation.

2.5 Document Representation

Document Representation task was applied after text preprocessing Afan Oromo nonfiction text documents. It used term weighting to represent documents. Term frequency was computed from number of times Afan Oromo nonfiction text word w terms found in Afan Oromo nonfiction text document d. Term weighting computed to decides the degree of importance of a given term to a given document. The term that occurs always in the Afan Oromo nonfiction text document was more closely connected to document comparing with term that occurs rarely in the document; but term that appears in almost entire of collected documents cannot identify classes and low weight will be assigned to this like terms. Afan Oromo nonfiction text document occurred more frequently and word occurred in such document less frequently components adjusted properly. From this, it is possible to conclude that document frequency and weight of terms inversely proportional to each other in Afan Oromo nonfiction text document corpus. The highest modest cost is achievable by inverse document frequency function idf. IDF is the entire number of Afan Oromo nonfiction text documents in corpus by number of Afan Oromo nonfiction text document word occurs in.

The need of Afan Oromo nonfiction text document word to given document was related to number of its availability and the identification power of Afan Oromo nonfiction text document word was inversely proportional to number document in which the word available. Depending on this idea important term weight determination generated as tf-idf and formula to calculate the weight of a Afan Oromo nonfiction text document word ω in a Afan Oromo nonfiction text document d is given by the following:-

$\omega ik = fik * \log (N/ni)$

Where:

ω_{ik} is the weight of term i in the k^{th} document in Afan Oromo nonfiction text document corpus.
f_{ik} is the frequency of the i^{th} term in the k^{th} document in Afan Oromo nonfiction text document corpus.
N is the number of documents in the Afan Oromo nonfiction text document corpus
n_i is the number of documents in which the i^{th} term occurs.
Document representation in current research was implemented by using R programming language. By using R programming language, tf and idf first computed using corpus. Here after, document term matrix computed by using that equation in 3.1. (See appendix C).

2.6 Dimensionality Reduction

Even if text preprocessing tasks were performed on Afan Oromo nonfiction text document, still there was high dimensionality of data. Thefore, dimensionality reduction techniques utilized to dimension of current data. Among dimensionality reduction techniques such as Information Gain, Mutual Information, Chi-Square Statistic, Term Strength, and Document Frequency Thresholding (Ozgur, 2004) we used Information Gain.

2.7 Document Similarity Measure

Document similarities measure is an important in text categorization approaches (text clustering and text classification) to measure similarity between documents. It is computed using cosine (Ozgur, 2004) as follows: -

$$cos(d1, d2) = \frac{d1.d2}{||d1||\ ||d2||}$$

In the above formula, $d_1.d_2$ is the dot product of d1 and d2 divided by the lengths of d1 and d2. This formula describes that similarity of two documents $d_1$and d_2 is cosine of the angle between document vectors. Result achieved from this formula is -1(opposite) to 1(exactly same), 0 usually independence and between those indicate that intermediate similarity or dissimilarity (Gebrehiwot Asefa, 2011).

2.8 Document Clustering

As discussed in previous section (2.2.1), Document clustering is a mechanism that breaks down existing collection of a text documents into important clusters (Grace and Desika, 2014). In clustering process most similar clusters are grouped together than in other clusters. It works with unlabeled documents that are freely available. This is due to it is an unsupervised learning which does not work with "pre-defined categories and labeled documents" (Dhillon, 2003). Document Clustering has advantages of working with unlabeled data that does not need manually labeling text document. It helps to find natural groups in data sets without knowing behavior of data within documents. Text clustering algorithms are used to cluster document into clusters. Among different clustering techniques discussed in previous section (2.1.1), partitioning techniques are used for Afan Oromo nonfiction text document clustering. Kmeans clustering algorithms were implemented by researcher. In partitioning techniques, initially division number of document is decided and operation further performed to achieve those all k numbers of division. As compared by Karypis et al (n.d.) hierarchical technique is not good due to its time quadratic complexity and whereas partitioning techniques use linear time complexity. Due to those reasons, partitioning techniques is better to use than hierarchical techniques clustering Afan Oromo text document with corresponding its algorithms such as direct k-means.

2.8.1 Basic K-means Algorithm

K-means Algorithm is the most widely document clustering algorithm that focused on center point and it uses concept of mean or median point of a group of points which is centroid (Karypis et al, n.d.). This centroid in referred to real data point. This centroid c is computed as the following formula for a set, S, of documents and their corresponding vector representations.

$$\boldsymbol{c} = \frac{1}{|S|}\sum_{d \in S} \boldsymbol{d}$$

For Afan Oromo text clustering Basic K-means clustering algorithm pointed out by Karypis et al (n.d.) adopted and used.

Basic K-means Algorithm for finding K clusters.

1. Select K points as the initial centroids.
2. Assign all points to the closest centroid.
3. Recompute the centroid of each cluster.
4. Repeat steps 2 and 3 until the centroids don't change

2.9 Document Classification

A discussed in previous section (2.2.2), classification is second approach for Afan Oromo nonfiction text categorization system. Output clustering processes is utilized as input for classification approaches. Afan Oromo nonfiction text document classification approaches utilized classification algorithm like SMO, NaiveBayes, BayesNet, and J48 decision Tree to build classifier for Afan Oromo text document.

As discussed in previous section Support Vector Machines is one the algorithm that used for text classification. It is supervised machine learning algorithm that helps to train data. The working principle of SVM depends on support vectors in training data (Briicher et al, n.d.). When the document that support vector unavailable from training data the working of Support Vector Machines remains the same.

2.10 Programming language

As discussed in last section researchers utilized python and R programming language in current research.

Python is a general purpose programming language created in the late 1980s, and named after Monty Python, that's used by thousands of people to do things from testing microchips at Intel, to powering Instagram, to building video games with the PyGame library. In present day study, researchers used python programming language for tokenization, stop word removal and stemming of Afan Oromo text documents.

R is powerful machine learning and statistical environment with large number of functions and libraries. RStudio is an integrated development environment (IDE) for R where essential source code typed on. RStudio is available as a commercial product and free product. We used free version of RStudio in this work due to it is freely available tools. In this study, R programming language utilized by researchers for computing term frequency (tf), inverse document term frequency (idf) and document term matrix depending the corpus to document representation (document indexing).

2.11 Tools

2.11.1 Waikato Environment for Knowledge Analysis (weka 3.8.1)

Weka is open source and freely available tool under the GNU General Public License. It was developed by New Zealand at Waikato University by using java programming language. Weka tool also supports different machine learning algorithms. Weka tool was used for text clustering (Jain et al., 2010). In current work, Weka tool was used for Afan Oromo nonfiction text clustering and classification. It contains tools for:-

- ✓ data pre-processing,
- ✓ data classification,
- ✓ regression,
- ✓ clustering,
- ✓ association rules, and
- ✓ Visualization.

Weka start window contains Explorer, Experimenter, KnowledgeFlow and SimpleCLI application that contain different function to work with text data (Bouckaert et al., 2010). Explorer used for opening and browsing data, experimenter for conducting experiment, knowledgeFlow used for drag-and-drop interface the work explorer and SimpleCLI performing execution using command line. Weka also has panels such as preprocess, classify, cluster, Associate, selet attribute and visualize. Those panels used for different purposes. For instance in current research, preprocess panel used for browsing data in different format (CSV or ARFF) that supported by weka tool. Classify and cluster panel used to give services of classification and clustering respectively. Cluster panel enable researchers to apply clustering algorithms in weka and test accuracy of model. Classify panel also used to apply classifiers algorithms in weka tool, and test accuracy predicting model. In current research, kmean clustering algorithm for clustering and BayesNaive, NaïveBayes algorithms, Sequential Minimal Optimization and J48 were applied from weka tool for text categorization purpose. In addition to classification tasks, researchers also used, classify panel to evaluate Afan Oromo nonfiction text categorization model by using weka built performance measurements for text categorization.

2.12 Performance Measures

The performance measurements are tested experimentally as discussed in previous section (2.6) for text clustering and text classification. For Afan Oromo nonfiction text document categorization system performance of text clustering and text classifier are tested separately and independently. The performance of text clustering was evaluated by using F-measure, percentage values of correctness, purity and entropy (Gottschalg & Ribeiro, 2014). In current research, we evaluated the clusters using percentage values of correctness of instances correctly clustered in clusters in weka tool. As discussed in previous section, accuracy, recall, precision, and Fmeasure were utilized to evaluate performance of Afan Oromo nonfiction text document. Accuracy, recall, precision, and F-measure computed by different equation. The percentage of document assigned to category c that are accurately gave to category c and it computed as follows:

$$Pi = \frac{TPi}{TPi + FPi} \quad \text{.....1}$$

Percentage of entire document allocated to category c is recall and it is defined as:

$$Ri = \frac{TPi}{TPi + FNi} \quad \text{.....2}$$

From this equation TPi for stands true positives, FNi stands for false negatives and FPi stands for false positives. True positives entire number of document categorized properly to category ci whereas false positives is the number of documents given to category ci that should have been given to other categories. Amount of document given to other categories that should have been given to category ci is false negatives. For precision (Pi) and recall (Ri) for category ci average of them is F-measure (F). It is computed as follows

$$Fi = \frac{2 \times Pi \times Ri}{Pi + Ri} \quad \text{.....3}$$

Since F-measure is joined method of precision and recall that favors point registering highest recall and precision, it used measure performance of the text classifier.

III. RESULTS AND DISCUSSION

Results

BayesNet classifier did classify 92.038% of 1369 instances correctly by taking 0.13 seconds. The performance of each catgory evaluated using precision, recall, ROC-Area and F-measure.

Correctly Classified Instances 1260 92.038 %
Incorrectly Classified Instances 109 7.962 %

=== Detailed Accuracy By Class ===

TP Rate	FPRate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.899	0.012	0.967	0.899	0.932	0.907	0.992	0.985	Guddifachaa
0.934	0.004	0.966	0.934	0.949	0.944	0.996	0.983	OromoCulturalSport
0.971	0.000	1.000	0.971	0.985	0.985	0.982	0.972	OromoCulturalDrug
0.907	0.002	0.951	0.907	0.929	0.927	0.984	0.957	OromoCulturalWeapons
0.885	0.007	0.719	0.885	0.793	0.793	0.966	0.870	Food&DrinkofOromoo
0.897	0.010	0.958	0.897	0.927	0.909	0.995	0.972	OromoConflictResolutitionStyle
0.961	0.054	0.840	0.961	0.897	0.867	0.984	0.927	OromoCulturalCloth
0.919	0.011	0.905	0.919	0.912	0.902	0.984	0.957	OromoMarryingCulturalStyle
Weighted Avg.	0.920	0.019	0.926	0.920	0.921	0.902	0.989	0.963

Table 1: Detailed Accuracy by Class from BayesNet

Detailed Accuracy by Class in table 2 was computed from confusion matrix in table 2. Referring the above detailed accuracy by Class in table 4.11, category with the highest score of accuracy was OromoCulturalDrug with 98.5% whereas Food&DrinkofOromoo scored the lowest accuracy 79.3%.

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	<-- classified as
348	1	0	0	6	5	24	3	a = Guddifachaa
0	141	0	0	0	0	7	3	b = OromoCulturalSport
0	1	33	0	0	0	0	0	c = OromoCulturalDrug
1	1	0	39	1	0	1	0	d = OromoCulturalWeapons
0	1	0	0	23	0	2	0	e = Food&DrinkofOromoo
8	0	0	1	1	253	15	4	f = OromoConflictResolutitionStyle
2	0	0	0	1	6	299	3	g = OromoCulturalCloth

1 1 0 1 0 0 8 124 | h = OromoMarryingCulturalStyle

Table 2: Confusion matrix using BayesNet

Discussion

NaiveBayes, J48, SMO, and BayesNet are text classifier algorithms used for Afan Oromo nonfiction text classification in current works. Each classifier algorithm has been shown different accuracy in with different time. NaiveBayes classifier had classified 80.7889%, J48 classifier has classified 94.3755%, SMO classifier algorithm had classify 92.9876% and BayesNet classifier had classify 92.038% of 1369 instances correctly. Classifier algorithms consumed some of CPU time during text classification tasks. As described in table 4.12, 0.02, 0.03, 0.36 and 0.13 seconds were consumed by NaiveBayes, J48, SMO andBayesNet classifiers algorithms respectively. In current research, to build text categorization model the researcher compared accuracy of classifier algorithms and selected classifier algorithms with higher acxuracy. Hence, J48 classifier algorithms has been selected for bulding Afan Oromo nonfiction text categotization due to its accuracy was highest than accuracy of other classifier algorithm in shown in Table 4.13 below.

Algorithms	Accuracy	Time consumed
NaiveBayes	80.7889%	0.02 seconds
J48	94.3755 %	0.03 seconds
SMO	92.9876%	0.36 seconds
BayesNet	92.038%	0.13 seconds

Table 3: Average Accuracy of NaïveBayes, J48, SMO, BayesNet, and classifiers algorithms

As shown in table 3, J48 classifier algorithms had highest performance for classification of Afan Oromo non-fiction text document than NaiveBayes, SMO, and BayesNet classifiers algorithms. By comparing classification algorithms in above table 4.13, depending on time each algorithm consumed to build Afan Oromo nonfiction text classifier model, NaiveBayes classifier algorithm consumed 0.02 seconds. In current study, accuracy was methods to choose classifier algorithms for testing Afan Oromo nonfiction Text categorization model. Therefore, researcher selected J48 text classifier algorithms for testing Afan Oromo nonfiction text documents categorization system as shown table 3

Testing Afan Oromo nonfiction Text categorization system

Afan Oromo text categorization model was built by algorithm that shows the highest performance during classification of Afan Oromo nonfiction text documents. We utilized testing set already prepared intentionally for text categorization.

Test set was split from training as 20% of data set previous section was used for testing Afan Oromo nonfiction text documents categorization system. It included entire 8 main categories. Test set was in ARFF format to be supported by weka tool and contains two sections header section and data section. It was manually labeled by language experts. Test set with arff format was used as input for weka tool for testing purpose on trained model by using training data set. The content of test set was shown in figure 4.6.

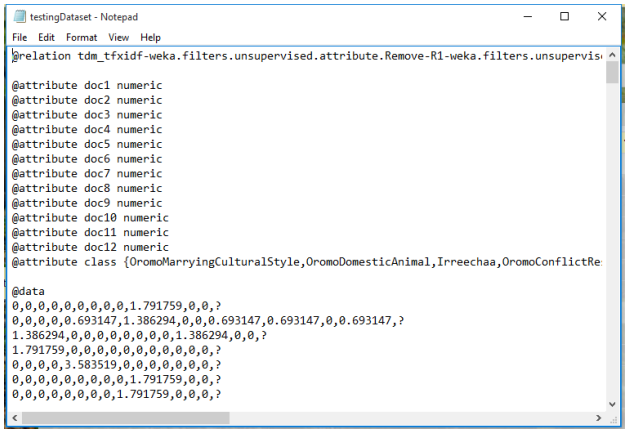

Figure 1: Test set format for weka tool

By using trained model, it is possible to classify new data using weka (Rodríguez D, n.d.). In weka tool, classification of the new text data was done by using weka Explorer and simple command line. In current study, we used weka Explorer for testing Afan Oromo nonfiction text categorization model built by using J48 classifier algorithm. Before loading test set into weka tool for testing purpose through weka Explorer, researchers prepared test set with structure suitable for learning model. Class label assigned to class were removed from data section of test set and replaced by "?" as shown in 1. "?" has been used as unknown

class label. Test set with unknown class label was loaded into weka tool using "testDataset.arff" file name and tested whether the model predict the class of new text data set with unknown class label or not. Class of each test set was tested and predicted by text categorization model that was built by using J48 classifiers algorithms. Researchers checked if the nonfiction categorization model correctly assigned or predicted the class of new text data in "testDataset.arff" or not. Afan Oromo nonfiction text categorization system correctly predicted 77.2% of instances into categories. Finally, result was summarized as in Table below.

Class Number	Accuracy	Class Name
0	82.1%	Guddifachaa
1	73.7%	OromoCulturalCloth
2	72.7%	OromoConflictResolutitionStyle
3	75.0%	OromoMarryingCulturalStyle
4	83.3%	OromoCulturalDrug
5	88.0%	OromoCulturalSport
6	83.3%	OromoCulturalWeapons
7	36.4%	Food&DrinkofOromoo
Average accuracy	77.2%	

Table 3: Afan Oromo nonfiction text Categorization testing results using Text categorization Model

Discussion

Referring accuracy in the table 4.15 above, researchers tried to evaluate categorization system by using built model. From this evaluation we concluded that the performance of entire system is influenced by performance of clustering algorithm. In current research, the category "OromoCulturalSport" has highest performance of 88.0 %(F-measure) than other categories. The higest performance of OromoCulturalSport was due to almost all of it instances were resides on its class and instances from other class distributed to this class (see table 4.2 above). By referring table 4.2 above, the category "OromoCulturalSport" which was cluster cluster "1" also have comon words those distributed it. On the other hand, "Food&DrinkofOromoo" has lowest performance than others. This lowest performance of "Food&DrinkofOromoo" arised from few common words it contained (see table 4.2 above). From result of testing Afan Oromo nonfiction text categorization system experiments above, we also generalized that about document that clustered and classified in different clusters and classes during classification and clustering. Some documents that clustered under one class by clustering algorithm are classified in other class by text classifiers. Now we illustrated this idea by using the hereunder statement.

Statement "**ingicc waaq nyaat dubar gamt ayyaan araar**"

This statement was taken as example from one of the Afan Oromo nonfiction text document utilized in current research. This statement contained terms (stem of words) those represent documents. From this statement the word "gamt" which mean team was clustered in "Guddifachaa" cluster in clustering tasks performed by kmeans text clustering algorithm. On other hand this word classified in "OromoCulturalSport classes" during classification algorithms during classification tasks. The over all performance of builted model was 77.2% as we understood from table 4.15. The performance of the this text categorization model was degraded due to different problems such stemming, and spelling error Afan Oromo nonfiction text documents used in this particular research.

IV. CONCLUSION

In natural language, as amount of text documents in electronic format increases, challenges of identifying relevant documents to a specific topic increases. Those Challenges lead to information overloaded. Hence, using certain mechanisms those reduce the problem of overloaded information is the main concern in natural language processing. One of the mechanisms that reduces problem of overloaded information is using text categorization. Text categorization is mechanism that enables the intended user to filter, manage, access and use information by minimizing challenges of information overloading. Text categorization utilizes machine learning approaches (i.e text clustering and text classification) to overcome problems of overloaded information. Using text clustering and text classification sequentially has its own advantages in text categorization. For instance, text clustering minimizes challenges of manually labeling text task

and helps to work with unlabeled data those are freely available in large amount for text classification. In general, text clustering has advantages of saving time, and cost for Afan Oromo nonfiction text categorization. It utilizes Kmeans clustering algorithms for clustering Afan Oromo nonfiction text documents. Several numbers of experiments were conducted using Kmeans algorithm. Different numbers of clusters were assigned at each experiment and experiment accuracy was evaluated. Finally, 8 numbers of classes were obtained from clustered documents as classes of Afan Oromo nonfiction text documents. Those 8 main classes were utilized for training Afan Oromo nonfiction text categorization model using classifier algorithms such as J48, NaïveBayes, BayesNet, and SMO. Among those classifiers algorithms, J48 classifier algorithm shows highest performance than others. Numerically the accuracy of J48 was 94.3755%, accuracy of NaïveBayes was 80.7889%, accuracy of BayesNet was 92.038%, and accuracy of SMO was 92.9876%. Referring accuracy of each algorithm, it is possible to conclude that j48 classifier algorithm is good classifier algorithm for building Afan Oromo nonfiction text categorization model. The testing set was manually labeled. Class label of testing set was removed and replaced by "?" unknown class label and tested wether the the constructed training model correctly assigned class label or not. For loaded test set this model predicts the class of laoded test set by 77.2059 % accuracy. In general, depending on this accuracy we conclude that the model is usable for categorizaing Afan Oromo nonfiction text documents.

V. RECOMMENDATION

The following works are recommended for from this thesis:-

- ✓ Spelling word of Afan Oromo words play vital role in Afan Oromo nonfiction text processing. It is solution to design spell corrector for word of Afan Oromo nonfiction text document and incorporate it to check and correct error of spelling words. Therefore, it is recommended to design and implement spell corrector for words of Afan Oromo nonfiction text document.
- ✓ In current study, KMeans clustering algorithms was utilized for clustering. Therefore others clustering were recommended to be used for clustering Afan Oromo nonfiction texts.
- ✓ Using and testing other classification algorithms for classifying Afan Oromo nonfiction text document without using text clustering approaches.
- ✓ Current study deals with only single label Afan Oromo nonfiction text classification and multi-label classification of Afan Oromo nonfiction is recommended.
- ✓ No corpus for Afan Oromo nonfiction text documents, therefore preparing internationally acceptable corpus for Afan Oromo nonfiction text documents is recommended future work.
- ✓ In current work, we utililized white space as Afan Oromo nonfiction text document word delimiter to tokenize Afan Oromo nonfiction text document word. This white space cannot tokenize Afan Oromo nonfiction text document word those composed two independent words separated from weach other by white space. Therefore, it is recommended for tokenizing Afan Oromo nonfiction text document word by developing other techniques.
- ✓ We used only postfix for stemming Afan Oromo nonfiction text document word. Still it is important properly identify other Afan Oromo nonfiction text document word affixes with language in possible ways and implement it for stemming Afan Oromo nonfiction text document word.
- ✓ We write python programming for stemming Afan Oromo nonfiction text document word due to lack already developed Afan Oromo nonfiction text document word stemmer tool. It will be future work to develop Afan Oromo nonfiction text document word stemmer tool.

VI. REFERENCES

[1]. Prof. Abera Diriba, (2009). Classification of Afan Oromo News Text: The Case of Radio Fana (master's thesis). Addis Ababa University, Addis Ababa.

[2]. Adel, D. S., (2007). Dimensionality Reduction Techniques for Enhancing Automatic Text Categorization (master's Thesis).

[3]. Andrea Addis, (2010). Study and Development of Novel Techniques for Hierarchical Text Categorization. University of Cagliari, Italy.

[4]. Baker, L.D, & Kachites, A. M, (1998). Distributional Clustering of Words for Text

Classification: ACM SIGIR, Cluster Quality. Journal of mathematics and computer science (2014).

[5]. Bouckaert, R.R., Frank,E., Kirkby, R.,Reutemann,P., seewald,.A., 7 Scuse,D., (2012). WEKA Manual for Version 3.6.1. University of Waikato, Hamilton, New Zealand.

[6]. Clsuterting, (n.d.). Clustering Example using RStudio (WIne example). (https://www.youtube.com/results?search_query=Clustering+Example+using+RStudio+%28WIne+example%29) Accessed on July, 2017.

[7]. Debela Tesfaye, (2011). A rule-based Afan Oromo Grammar Checker.IJACSA.Vol. 2, No. 8, 2011.

[8]. Edmunds, A., & Morris, A., (2000). The problem of information overload in business organizations: a review of the literature. International Journal of Information Management.

[9]. Faraz, A., (2015). An elaboration of text Categorization and automatic Text Classification through Mathematical and Graphical Modeling. Computer Science & Engineering: An International Journal (CSEIJ), Vol.5, No.2/3, June 2015.

[10]. Gebrehiwot Asefa, (2011). A two steps approach for tigrigna text categorization (master's thesis). Addis Ababa University, Addis Ababa.

[11]. Getachow Rabirra., (2016). Oromo Grammar 5th edition. Addis Ababa, Ethiopia.

[12]. Hannah, G.G., & Desika, K., (2014). Experimental Estimation of Number of Clusters Based on Cluster Quality. Journal of mathematics and computer science 12 (2014).

[13]. Jain, Y., & Kumar, A.N., (2014). A Theoretical Study of Text Document Clustering. Yogesh Jain et al, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2), 2014, 2246-2251

[14]. Jain,S., Afshar, M. A., & Doja, M.N., (2010). K-Means Clustering Using Weka Interface. Proceedings of the 4th National Conference; INDIACom-2010. Computing For Nation Development, February 25 – 26, 2010.

[15]. John, C. (1998). Technical Paper Review Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. USA: Morgan Kaufmann.

[16]. Khan, A., Baharudin, B., Hong, L.L., & Khan, K., (2010). A Review of Machine Learning Algorithms for Text-Documents Classification. Journal of Advances in Information Technology, Vol. 1, No. 1, February 2010.

[17]. Maron, M.E & Kuhns, J.L (1960). Probabilistic Indexing and Information Retrieval. Tehe RAND Corporation. Sanat Monica, California.

[18]. McCallum, A., Nigam, K., Thrun, S. & Mitchell, T. (2000) Text Classification from Labeled and Unlabeled Documents Using EM. Boston: Kluwer Academic Publishers, 39(2),

[19]. Meenakshi & Singla, S., (2015). Review Paper on Text Categorization Techniques. SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE) – EFES April 2015. ISSN: 2348 – 8387. Available at www.internationaljournalssrg.org

[20]. Mostak, K., & Hoq, G., (2014). Information overload: causes, consequences, and remedies: A study, vols LV-LV1.

[21]. Raj, B., S., & Paul, A., (2013). Clustering Algorithms: Study and Performance Evaluation Using Weka Tool. International Journal of Current Engineering and Technology ISSN 2277 – 4106 © 2013 INPRESSCO. Available at http://inpressco.com/category/ijcet

[22]. Ribeiro, A. A., Gottschalg, C. D, (2014). Automated text clustering of newspaper and scientific texts in brazilianportuguese: analysis and comparison of methods. University of Brasilia (Universidade de Brasilia–UnB), Brasilia, DF, Brazil. JISTEM J.Inf.Syst.Technol. Manag. vol.11 no.2 Sao Paulo May/Aug. 2014.

[23]. Rodriguez, (n.d.). Making predictions on new data using Weka. University of Alcala available at http://www.cc.uah.es/drg/courses/datamining/ClassifyingNewDataWeka.pdf

[24]. Sebastiani, F., (2002). Machine Learning in Automated Text Categorization.ACM Computing Surveys.ConsiglioNazionaledelleRicerche, Italy: ACM, p.10-15.

[25]. Seffi Gebeyehu & Sreenivasa, .V.R, (2014). A Two Step Data Mining Approach for Amharic Text Classification. American Journal of Engineering Research (AJER). e-ISSN : 2320-0847 p-ISSN : 2320-0936 Volume-03, Issue-04, available at www.ajer.org.

[26]. Sun, A., & Lim, E., (2001). Hierarchical Text Classification and Evaluation. Proceeding of International conference on data mining. Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 2001), Pages 521--528, California, USA, November 2001.

[27]. Tiwari, M., & Singh, R., (2012). Comparative Investigation of KMeans and K-Medoid Algorithm of IRIS Data. In the International Journal of Engineering Research and Development.

[28]. Wakshum mekonnen. (2000), Development ofa stemming algorithm for Afaan Oromo University, Addis Ababa, Ethiopia.

[29]. Witten, H., (n.d.). More data mining with weka (3.6 Evaluating clusters). Department of computer science univrsity of Waikato New Zealand. (https://www.youtube.com/watch?v=9aODdNSAauI&t=21s) Accessed on June, 2017.

[30]. Zelalem Sintayehu, (2001). Amharic News Classification. MSc Thesis. Addis Ababa University, Addis Ababa, Ethipia