International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-2, July 2019

Development of Document Clustering Technique for Gurmukhi Script using Fuzzy Term Weight

Mukesh Kumar, Amandeep Verma

Abstract: ***Document clustering is an unsupervised machine learning technique which designates the creation of classes of a certain number of similar objects without prior knowledge of data-sets. These classes of similar objects are known as clusters; each cluster consists unlabeled data objects in such a way that data objects within the same cluster have maximum similarity and have dissimilarity to the data objects of other groups. The purpose of this research work is to develop domain independent Gurmukhi script clustering technique. It is the first ever effort as no prior work has been done to develop domain independent clustering technique for Gurmukhi script. In this paper, a hybrid algorithm for the development of document clustering technique for Gurmukhi script has been developed. The experimental results of proposed document clustering technique reveal that the proposed hybrid technique performs better in terms of defining number of clusters, creation of meaningful cluster titles, and in terms of performance regarding assignment of real time unlabeled data sets to the relevant cluster as a result of various pre-processing steps like segmentation, stemming, normalization as well as extraction of named/noun entities, creation of cluster titles and placing text documents into relevant clusters using fuzzy term weight.***

Index Terms: ***Data mining techniques, Document clustering, Gurmukhi script clustering technique, Machine learning, Punjabi Text document clustering, Unsupervised learning.***

I. INTRODUCTION

In the World of Information Technology (IT), huge amount of digital data is being generated every time. Due to significant increase in usage of the online communication, a lot of data are being collected and stored by the system. From this large volume of data, it is very difficult to identify and to extract the useful information. For the better utilization of available huge data; it is required to be transformed into valuable information. Therefore to better leverage the information; an automation is necessary to extract useful information that can be utilized for further processing. Data mining is a process of extracting or mining the useful information from the large amount of data. Data mining is an intelligent process that helps to turn data into information, and information into knowledge. Data Mining is a multi-disciplinary field which deals with various approaches like pattern matching, artificial intelligence, visualization, expert systems, machine learning and so on. Machine learning is the study of computer algorithms that learns itself by training data or by its experience. There are two types of machine learning approaches namely, supervised learning and unsupervised learning[1,8]. In supervised learning, an algorithm is trained using previously classified data known as training set of data. After the algorithm is trained on previously known data, it is applied to unknown input data sets to get useful information. Neural network, classification and regression are the best examples of supervised machine learning. Whereas, in unsupervised learning, an algorithm is trained without providing any prior information about the data sets. In this approach, the machine is trained itself (without any training set) in an iterative manner based on input data sets only. Clustering, association rules, and self-organizing maps are the best examples of unsupervised machine learning . Clustering is an unsupervised learning as there are no predefined classes of training sets. It is a process of grouping similar objects together based upon its similarities [4]. Clustering is different from classification as it deals with unsupervised learning whereas classification deals with supervised learning [9]. A cluster is supposed to consist non-empty sub set of similar objects. These similar objects within the cluster are known as cluster members. Two clusters are disjoint clusters if these two clusters contain no common objects. The clustering algorithms are broadly divided into four categories namely: hierarchical clustering techniques, density based clustering techniques; grid based clustering techniques and partitional clustering techniques [2]; again partitional clustering is of two types, namely: hard clustering and soft clustering. Potential application areas of clustering include: pattern recognition, image analysis, information retrieval as well as in the domains like business, economics, chemistry and so on. For instance, clustering is applied in image processing to make the clusters of similar images based on their visual properties. Clustering can be used in marketing to develop groups of persons and products based upon various aspects that may be useful for further analysis. In web mining, the groups of users can also be developed based on their access patterns [9].

Revised Manuscript Received on 30 July 2019.
***** Correspondence Author
Mukesh Kumar, PG Dept. of Computer Science, Mata Gujri College, Fatehgarh Sahib, Punjab, India.
Amandeep Verma, Punjabi University Regional Centre for Information Technology & Management, Mohali, Punjab, India.

II. RELATED WORK

A. Punjabi Documents Clustering System

The "Punjabi Documents Clustering System" is domain based document clustering system for Punjabi text proposed by Sharma & Gupta [7] using hybrid approach. It was first ever attempt in this direction keeping in view the semantics of Punjabi language. The proposed hybrid approach for clustering of Punjabi documents performed a pre-processing phase and processing phase.

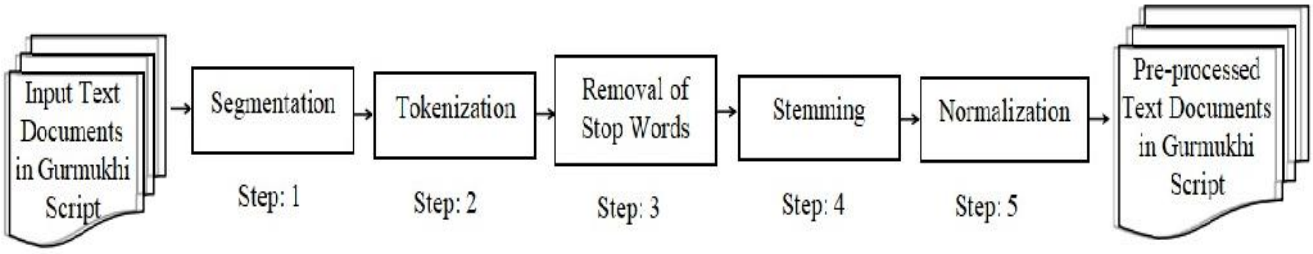

Fig.1- Steps of Pre-processing Phase

The pre-processing phase comprised removal of punctuation marks as a first step. To maintain the uniformity of spelling of words in all the input text documents, a manual normalization table has been prepared as a second step. In this process, different spellings for a single word has been stored in a table which are matched with each word of input text documents; and replaced with first spelling if match is found.

The processing phase comprised various steps. In the first step, the phrases of various lengths have been extracted from the sentences with the help of karkas. In second step, term frequency of each phrase is calculated for assigning weight to all the extracted phrases. In third step, all phrases are sorted according to its weight and top 'k' key phrases are extracted which are considered to find the similarity among the input text documents. In fourth step, similar documents are found by matching the extracted key phrases and created the initial clusters. At the last, final clusters are created by matching the key phrases of unrecognized documents with the frequent terms of each cluster.

B. *Limitations of "Punjabi Documents Clustering System"*

The major limitation of the 'Punjabi Documents Clustering System' proposed by Sharma & Gupta is that the proposed system is specific to sports domain only. There is an urgent need to develop domain independent document clustering technique for Gurmukhi script as no domain independent document clustering technique for Gurmukhi script is available. Secondly, during pre-processing phase; the normalization process of Punjabi word spellings is performed using a manual normalization table and is the main drawback of proposed system which averts the basic requirement of unsupervised learning. Moreover, during processing phase; the cluster title list is created manually which again averts the basic requirement of unsupervised learning. The performance of the system proposed by Sharma & Gupta is not desirable in respect to assignment of real time unlabeled data sets to the relevant cluster as there is an immense need to perform pre-processing steps like removal of stop words, automatic stemming and normalization of Punjabi text. At the most; only the text documents which consist adequate number of key-terms should be placed into the concerned clusters. A technique like 'Fuzzy Term Weight' is an essential requirement to achieve such performance.

III. PROPOSED WORK

As per the review of literature done to carry out this research work, the proposed system i.e. 'Development of Document Clustering Technique for Gurmukhi Script' is first ever effort as no prior work has been done to develop domain independent Gurmukhi script clustering technique. Development of proposed system comprises of two main phases namely: pre-processing phase and processing phase.

A. Pre-processing Phase

The input text documents to be processed are generally unstructured in nature. So, before further processing, it is the first requirement to convert unstructured text into structured text format. To do so, various steps are performed during pre-processing phase which are shown in Fig.-1.

1) ***Segmentation:*** Segmentation is the first step of pre-processing phase in which text of input document is segmented into sentences based upon the boundary of sentence. In Gurmukhi script, the "।" (known as "dandi") is used as boundary of sentence. So, in this step, sentences are segmented on occurrence of every sentence boundary. This step results in collection of sentences in a list.

2) ***Tokenization****:* The process of tokenization breaks down the sentences into words based upon space and punctuation marks (e.g. hyphen, comma etc.). In this step, the sentences in Gurmukhi script are broken-down into words on occurrence of every space or punctuation mark.

3) Removal of Stop Words: Stop words are common words which do not provide any useful information in taking decision regarding selection of text documents for further processing. So, these words must not be included as indexing terms and must be eliminated from text. For instance, in the sentence, "□□□□ □□□□ □□□ □□ □□□□ □□□ □□□□ □□□□ □□ □□□ □□□□□ □□□ □□□" the words "□□□□", "□□", " □□□", "□□", "□□□□□", "□□□", "□□" do not provide any useful information and must be eliminated from the text so as to lesser the weight of text.

In this step, occurrence of every stop word in the text document of Gurmukhi script is detected and removed if found. For this purpose, apart from self created list of stop words; a list of 184 stop words in Gurmukhi is also used [5].

4) Stemming: Stemming is the process of getting root of word's grammatical forms known as 'stem'. Stemming has its own significant role in proving the enhancement of overall performance of clustering process. For e.g. grammatical forms of root word "□□□□□□□" are "□□□□□□□" and "□□□□□□□□□". The stemming process will convert "□□□□□□□" and "□□□□□□□" word forms to its root word "□□□□□□□". Thus, the words "□□□□□□□", "□□□□□□□" and "□□□□□□□" will be considered as similar words. So, after performing stemming. the text documents consisting various grammatical forms of a stem; will also represent similarity among these text documents and will be placed in the same cluster. In this step automatic stemming of noun words in Gurmukhi script is performed [10]. The process of stemming has a significant role in creating list of meaningful cluster titles also. A list of cluster titles created from the root words, for instance: □□□□, □□□□□, □□□□□, □□□□□□□□, □□□□□□□□, □□□, □□□□□□ represent more meaningful cluster titles than a list of cluster titles created from its grammatical word forms, for instance: □□□□□□, □□□□□□□, □□□□□□□, □□□□□□□□□□, □□□□□□□□□□, □□□□□, □□□□□□□□.

*5) **Normalization:*** Normalization is the process of maintaining uniformity of spellings of words among all the input text documents. In Gurmukhi script, most of the words are written in different manners for e.g. "□□□□□" can also be written as "□□□□", "□□□□□" can also be written as "□□□□", "□□□□□□" can also be written as "□□□□□", "□□□□□□□" can also be written as "□□□□□□" etc. To overcome the problem of non-uniformity among various spellings of words, standardization of words among input text documents is maintained using rule based automatic normalization of words in Gurmukhi script [11]. Normalization also has its own role to achieve high accuracy in clustering process. For e.g. the word "□□□□□" may be written with different spellings as "□□□□" in another input text document. The "normalization" process will provide uniformity of word spellings by converting the word "□□□□□" to its uniform word "□□□□". Thus, the two words with different spellings "□□□□□" and "□□□□" will be considered as similar words and after performing normalization process, the text documents consisting non-uniform spellings of words; will also represent similarity among these text documents and will be placed in the same cluster.

B. Processing Phase

After performing the pre-processing steps, the input text documents in Gurmukhi script which were in unstructured text format; have been converted into structured text format and as per the requirements for further processing. There are numerous steps performed during the processing phase which are shown in Fig.-2.

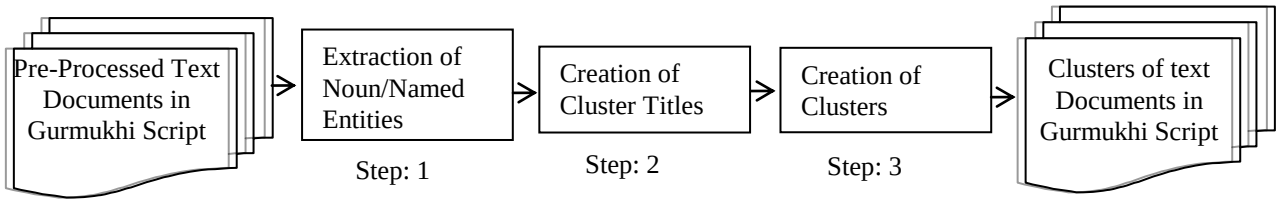

Fig.2- Steps of processing Phase

1) ***Extraction of Named/Noun Entities:*** A noun is the name of a person, place or thing. Nouns represent more informative features and provide compact representation of document's content. In this step, named/noun entities are extracted from text documents and the list of extracted nouns/names will be used further to create a list of cluster titles as well as for extraction of weighted keywords from the document. There are numerous methods that help in extracting named/noun entities from the text documents in Gurmukhi script:

- Using dictionary of Punjabi names/nouns or Punjabi noun corpus: To identify noun entities using dictionary of Punjabi names/nouns or Punjabi noun corpus; the root words from text documents in Gurmukhi script are matched with the words in dictionary of Punjabi names/nouns or Punjabi noun corpus.

- Using NER module: Named entities can be extracted using various types of NER modules for Punjabi language developed by the research scholars.

- Using POS tagger: The named/noun entities can be extracted using POS tagger modules for Punjabi language developed by the research scholars as well as by research oriented organizations.

In this work, Stanford POS tagger for Punjabi language is used to extract named/noun entities from the text documents in Gurmukhi script. This POS tagger module has been developed by KBCS, C-DAC, Mumbai [13].

2) *Creation of Cluster Titles:* In this step, a list of cluster titles is created by calculating term frequency (TF) of each noun entity (noun entities are extracted from all the pre-processed text documents during earlier step). Term Frequency is a numerical statistic which assigns a weight to every term of text documents. The weight assigned to each term reflects importance of the term; based on number of times it appeared in the text documents.

a) **Fuzzy Term Weight:** As associated weight represents importance of term; hence proves an important factor in creating list of cluster titles. Classical TF method is not adequate enough for assigning importance to the terms. Consider an instance,

where a term occurs only once in a text document or where only a term occurs repeatedly in a whole text document; should assign least importance. Whereas, most importance should be assigned to a term when it occurs for adequate number of times in a text document. Considering another instance, where a term for e.g "□□□□" is occurred in a single text document 40 number of times. On the other hand, a term for e.g "□□□□□□□□" is occurred 4 number of times each in 8 text documents. In this situation, because a term "□□□□□□□□" occurred in more number of documents; a cluster title of term "□□□□□□□□" is desired to be created giving less weight to a term occurred more number of times in single document. The use of classical TF method can not meet with such standards. To overcome such problems, and meet with such desired standards 'Fuzzy Logic based Term Weighting' method is used.

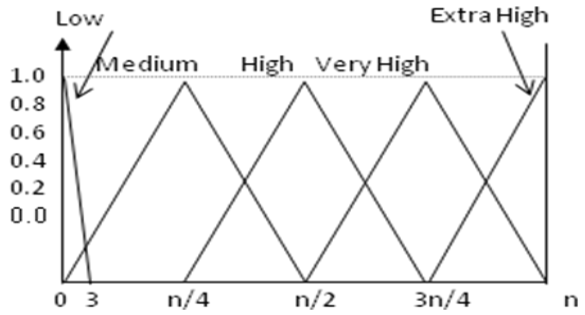

Fig.3-Membership Functions of Input Variable 'Term Frequency'

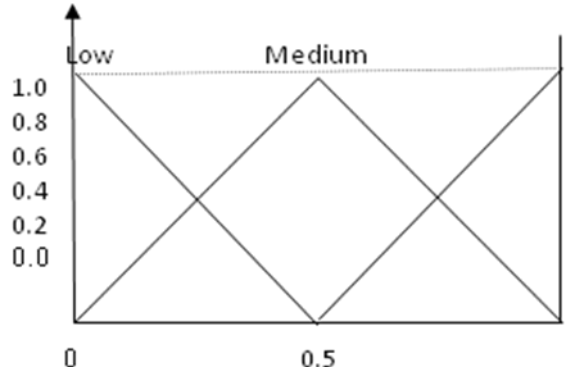

Fig.4- Membership Functions of Output Variable 'Term eight'

Fig.3 shows the labels and their associated membership functions for an input variable 'Term Frequency'. Values for the input variable 'Term Frequency' are normalized in the range of 0 to 'n', where 'n' is the number of total terms in concerned text document. Fig.4 shows the labels and their associated membership functions for an output variable 'Term Weight'. Values for the output variable 'Term Weight' are normalized in the range of 0 to 1, where 1 represents maximum term weight.

Once membership functions have been defined for input and output variables, fuzzy mapping rules can be developed to relate output actions to the observed inputs. This phase is also known as rule base or knowledge base of fuzzy logic system.The decision making capabilities are coded in a set of rules. The typical intuitive rules are as given below:

R1: IF *Term Frequency = Low* THEN *Term Weight = Low*

R2: IF *Term Frequency = Medium* THEN *Term Weight = High*

R3: IF *Term Frequency = High* THEN *Term Weight = Medium*

R4: IF *Term Frequency = Very High* THEN *Term Weight = Low*

R5: IF *Term Frequency = Extra High* THEN *Term Weight = Low*

To convert the fuzzy output back to its corresponding crisp output, the mean of maximum (MOM) defuzzification process is performed [3]. The defuzzification process using MOM can be expressed as:

$$MOM(A) = \frac{\sum_{y^* \in P} y^*}{|P|}$$

Where 'P' is the set of output values 'y' with highest possibility degree in fuzzy output 'A'.

After computing term weight from each text document, a sum of term weights from all the text documents is calculated and assigned to a particular term. The noun entities (i.e. terms) with its assigned weight are sorted by term weight in descending order and selected top 'k' entities to be defined as cluster titles. Here the value of 'k' represents the number of entities in percentage. Here the value of 'k' is an important factor for defining the list of cluster titles. A desirable list of cluster titles can be created by setting the value of 'k' ranging from 0.5 to 'n'. Using this technique, a list of meaningful cluster titles can be created as the term frequency of meaningless noun entities will be very low and will not fall within top 'k' entities sorted by term frequency weight.

3) **Creation of Clusters:** The clusters of similar text documents are created based upon term frequency weight of each noun entity from all the pre-processed text documents. The process of calculating term frequency weight and selecting top 'k' entities has already been performed during the earlier step for creating the cluster titles. The pre-processed text documents consisting same terms (with valid weight) as cluster title; are placed into same cluster. It should be noted that the text documents assigned to concerned cluster are input text documents corresponding to pre-processed text documents. After this process, the input text documents which could not be assigned to any cluster; are placed into a different cluster; named as "Miscellaneous". Text documents assigned to the cluster named as "Miscellaneous" are the text files which do not belong to any specific cluster.

IV. ALGORITHM

Step 1. Read input text documents in Gurmukhi script from the Corpus.

Step 2. For each input text document 'D_i':

a) [Perform Segmentation]:

i. Read 'D_i', replace every occurrence of "|" and "||" (known as "dandi") by end of line character i.e. '\n'.

ii. Save it as 'S_i'.

b) [Perform Tokenization]:

i. Read 'S_i', create list of words by splitting sentences into words at every occurrence of "white space" and punctuation marks like "!", ",", ":", "(", ")".

ii. Save it as 'T_i'.

c) [Perform Removal of Stop Words]:
i. Read 'T_i', replace every occurrence of "stop word" by null character.
ii. Save it as 'R_i'.
d) [Perform Stemming]:
i. Read 'R_i', perform stemming of noun/named entities using "Punjabi Language Stemmer for Nouns and Proper Names".
ii. Save it as 'ST_i'.
e) [Perform Normalization]:
i. Read 'ST_i', for every word, replace occurrence of letters "Adhak" (i.e. '□'), "Bindi on top" (i.e. '□'), "Bindi in foot" (i.e. '□') by null character and replace foot letter (i.e. '□') by normal letter.
ii. Save it as 'N_i'.
f) [Extract Noun/Named Entities]:
i. Read 'N_i', extract list of noun/named entities using POS Tagger for Punjabi language.
ii. Save it as 'NN_i', where 'NN_i' is list of noun/named entities separated by comma (i.e. ',').
Step 3. [Calculate Term Weight of each Noun/Named Entity]
a) [Create set of unique noun/named entities from all text documents]
i. For each text document 'NN_i' do:
ii. Read 'NN_i' as 'd'
iii. Create set of unique terms as 's' from all the text documents.
iv. For end
b) [Create 'list' of terms in the form of: 'Term', Total_Term_Weight, Term_Weight in 1st Document, Term_Weight in 2nd Document, … Term_Weight in 'n'th Document]
i. Define 'Term_Count', 'Tot_term_Weight' as initial parameters and 'TF_List' as list.
ii. For each 'term' in set 's' do:
iii. Set 'Tot_TF':= 0
iv. Append 'term' to 'list'
v. For each text-document 'NN_i' do:
vi. Count number of terms in text-document 'NN_i' as 'Tot_Count'
vii. For each noun/named-entity 'NE_i' in text-document 'NN_i' do:
viii. If term == NE_i
ix. Increment Term_Count as Term_Count+=1
x. End if
xi. End for
xii. Set Ct := Term_Count and Term_Count := 0
xiii. [Perform Fuzzification and De-fuzzification to calculate Term_Weight]
Call FUZZ_DEFUZZ() defined in 'Procedure-1' passing number of matched terms 'Ct' and total number of terms 'Tot_Count' occurred in text-document 'NN_i' as arguments and getting 'TermWeight' as returning value.
xiv. Append Term_Weight to 'List'.
xv. Calculate term weight in all the text-documents as Total_Term_Weight+=Term_Weight.
xvi. End for
i. Insert 'Total_Term_Weight' in 'List' as first item.
ii. [Now 'List' consists term, total term weight in all the text-documents, and term weight (for a term in set 's') in each text-document 'NNi'].
iii. Append 'List' in 'TF_List' as a record.
iv. Empty list 'List'
v. End for
vi. [Sort list 'TF_List' of Noun/Named entities by 'Total_Term_Weight' in descending order and select first 'k' entities. Where 'k' is numeric value in %]
vii. Sort 'TF_List' by 'Total_Term_Weight' in descending order.
viii. Count total number of entities 'Tot_Num_Entities' in 'TF_List'.
ix. Select first 'k' number of entities as:
x. First_K_Num = (K_Val * Tot_Num_Entites)/100
xi. (Where 'K_Val' is user defined numeric value in %)
xii. Create list of selected first 'k' entities as 'First_k_Entities_List'
xiii. [Define Cluster Titles]
xiv. For each 'record' in 'First_K_Entities_List' do:
xv. [Create list of cluster titles by extracting 'term' part from the 'record' i.e. first element of 'record']
xvi. Append record[0] to 'Cluster_Titles_List'
xvii. End for
xviii. [Create Clusters]
xix. [Create clusters and assign concerned text documents in each cluster]
xx. Count number of terms in list 'Cluster_Titles_List' as 'Number_of_Clusters'
xxi. For i = 0: Number_of_Clusters -1
xxii. Single_Record= First_K_entities_List[i]
xxiii. Cluster_Title=Single_Record[0]
xxiv.[Creation of cluster]
xxv. Create folder; name it as contents of 'Cluster_Title'
xxvi.[Placing concerned text-documents into cluster]
xxvii. For i = 1: Num_input_docs:
xxviii. [Place text document with high term weight (term weight > 0.5) into the concerned cluster; here term weight of each document starts from 3rd element of list 'Single_Record'; as first element is 'term' and second element is total weight of term from all the text documents]
xxix.If (Single_Record[i+1]>0.5:
xxx. Move input text document 'Di' to folder 'Cluster_Title'
xxxi.End if
xxxii. End for
xxxiii. End for

c) Create folder 'Miscellaneous' and place unrecognized/remaining files (if any) from the corpus of input text files.

Procedure-1

FUZZ_DEFUZZ(int Ct, int Tot_Ct)

[Fuzzification: Map crisp input values to fuzzy values].

a) [Define membership functions for an input variable 'Term Frequency']

Low=trimf(0,0,3)

Medium=trimf(0, Tot_Ct/4, Tot_Ct/2)

High=trimf(Tot_Ct/4, Tot_Ct/2, 3*Tot_Ct/4)

Very High=trimf(Tot_Ct/2, 3*Tot_Ct/4, Tot_Ct)

Extra High=trimf(3*Tot_Ct/4, Tot_Ct, Tot_Ct)

b) [Define membership functions for an output variable 'Term Weight']

Low=trimf(0, 0, 0.5)

Medium=trimf(0, 0.5, 1.0)

High=trimf(0.5, 1.0, 1.0)

Step 2. [Evaluate IF-THEN Rules]

R1: IF Term Frequency = Low THEN Term Weight = Low

R2: IF Term Frequency = Medium THEN Term Weight = High

R3: IF Term Frequency = High THEN Term Weight = Medium

R4: IF Term Frequency = Very High THEN Term Weight = Low

R5: IF Term Frequency = Extra High THEN Term Weight = Low

Step 3. [Defuzzification: Transpose fuzzy outputs to crisp output using MOM]

a) Crisp output (Y) = $\frac{\sum_{y^* \in P} y^*}{|P|}$

b) Return 'Y'

V. EXPERIMENTAL ANALYSIS AND RESULT

To evaluate the performance of proposed clustering technique and to compare with existing technique, we have implemented the proposed algorithm using Python 3.4.

A. Data Set

To analyze proposed hybrid algorithm, a data-set comprising of 200 text documents in Gurmukhi script is taken as an input to the system. This data-set is self created by collecting text documents in Gurmukhi script from various Punjabi news websites available online as no such standard data-set is available for text document in Gurmukhi script. The self created data-set is categorized into various domains.

B. Experimental Discussions

To execute proposed algorithm of document clustering technique for Gurmukhi script, the input text documents in Gurmukhi script are selected. After selecting input text documents; first, the steps of pre-processing phase are performed in sequence and then steps of processing phase are performed in sequence. The screen shots of system are shown in Fig.5.

Cluster validation is the process of evaluating quality of clusters produced by any clustering technique. To evaluate the accuracy of results obtained by the proposed clustering technique, F-Measure is applied[12].

F-Measure is a standard evaluation method i.e. harmonic mean of Precision and Recall. The Precision, Recall and F-Measure for natural classes 'Ki' (representing relevant set of text documents) and cluster 'C_j' are calculated as:

$$\text{F-Measure}(K_i,C_j) = \frac{2 * [\text{Precision}(K_i,C_j) * \text{Recall}(K_i,C_j)]}{[\text{Precision}(K_i,C_j) + \text{Recall}(K_i,C_j)]}$$

Where:

$\text{Precision}(K_i,C_j) = n_{ij} / |C_j|$ and $\text{Recall}(K_i,C_j) = n_{ij} / |Ki|$

Here, 'n_{ij}' is the total number of natural classes 'Ki' in cluster 'C_j'. We treat each cluster as the result of a query and each class as the desired set of documents for a query.per the review of literature done to carry out this research work, the proposed system i.e. 'Development of Document Clustering

TABLE-1: CLUSTERS FOR EXISTING PUNJABI DOCUMENT CLUSTERING SYSTEM

Sr. No.	Cluster Name	Precision	Recall	F-Measure
1.	□□□□□□□ (Football)	0.66	0.92	0.76
2.	□□□□ (Hockey)	0.93	0.88	0.90
3.	□□□□□□ (Cricket)	0.41	1.00	0.58
4.	□□□□□ (Tenis)	0.20	1.00	0.33
5.	□□□□□□□ (Volleyball)	0.66	1.00	0.79

TABLE-2: CLUSTERS FOR PROPOSED DOCUMENT CLUSTERING TECHNIQUE FOR GURMUKHI SCRIPT WITHOUT USING FUZZY TERM WEIGHT

Sr. No.	Cluster Name	Precision	Recall	F-Measure
1.	□□□□□□□□ (Vidiarthi)	0.75	1.00	0.86
2.	□□□□□□□ (Loktantar)	0.00	1.00	0.00
3.	□□□□□ (Kisan)	0.60	1.00	0.75
4.	□□□ (Desh)	0.33	1.00	0.49
5.	□□□□ (School)	0.00	1.00	0.00
6.	□□□□□□□ (Football)	0.66	1.00	0.79
7.	□□□ (Team)	0.51	1.00	0.67
8.	□□□ (Khed)	0.59	1.00	0.74
9.	□□□□□□ (Khidari)	0.32	1.00	0.48
10.	□□□□ (Hockey)	0.93	1.00	0.96
11.	□□□□□□ (England)	0.00	1.00	0.00
12.	□□□□□ (Sarkar)	0.37	1.00	0.54

TABLE-3: CLUSTERS FOR PROPOSED DOCUMENT CLUSTERING TECHNIQUE FOR GURMUKHI SCRIPT USING FUZZY TERM WEIGHT.

Sr. No.	Cluster Name	Precision	Recall	F-Measure
1.	□□□□□□□□ (Vidiarthi)	0.90	1.00	0.94
2.	□□□□ (Vishav)	1.00	1.00	1.00
3.	□□□□□□□ (Football)	1.00	1.00	1.00
4.	□□□□□ (Punjab)	1.00	1.00	1.00
5.	□□□ (Desh)	0.87	1.00	0.93
6.	□□□ (Team)	1.00	1.00	1.00
7.	□□□ (Khed)	1.00	1.00	1.00
8.	□□□□□□ (Khidari)	1.00	1.00	1.00
9.	□□□□□□ (Cricket)	0.87	1.00	0.93

10.	□□□□□ (Kisan)	1.00	1.00	1.00
11.	□□□□ (Hockey)		0.93	1.00
0.96				
12.	□□□□□ (Sarkar)	1.00	1.00	1.00

TABLE-4: OVERALL EFFICIENCY OF DOCUMENT CLUSTERING SYSTEMS.

Sr. No	Clustering System Algorithm	Precision	Recall	F-Measure
1.	Existing Punjabi Document Clustering System	0.57	0.96	0.67
2.	Proposed Clustering System without using Fuzzy Term Weight	0.42	1.00	0.52
3.	Proposed Clustering System using Fuzzy Term Weigh	0.96	1.00	0.98

As shown in Table-1, the existing clustering system resulted in creation of clusters from the sports domain only. The significant pre-processing phases like stemming and normalization have not been performed in this clustering system, as a result some of the text documents have not been recognized and could not be placed in relevant cluster.

On the other hand, as shown in Table-2, domain independent document clustering system using Gurmukhi script created clusters of various domains but again this system resulted in very low 'Precision' as it created three fake clusters namely: '□□□□□□□', '□□□□', and '□□□□□□' because the system could not recognized fake text documents. Here, the text documents consisting only a term/sentence repeatedly is not considered to be a fake text document. Moreover, the text document consisting inadequate number of key-terms are also placed in the clusters resulting in low 'Precision'.

In proposed clustering system using fuzzy term weight, as shown in Table-3, no such fake cluster has been created. As well as, with the use of fuzzy term weight, only the text document consisting adequate number of key-terms are placed in relevant cluster, resulting in higher 'Precision'.

Fig.5(a)- Screenshot of Main Screen

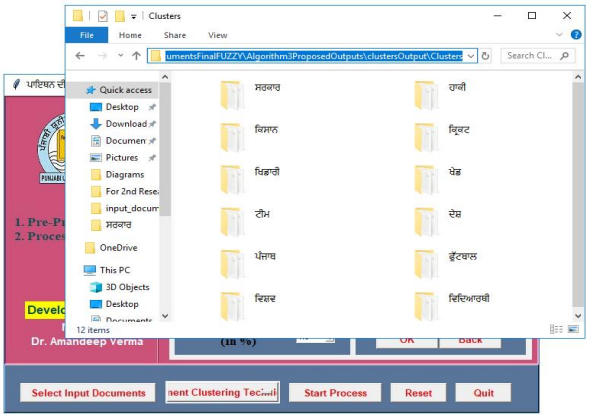

Fig.5(b)- Screenshot of Selecting Input Text Documents

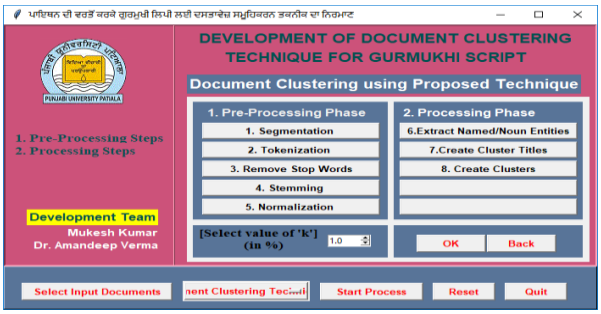

Fig.5(c)- Screenshot of various Steps of Proposed Technique

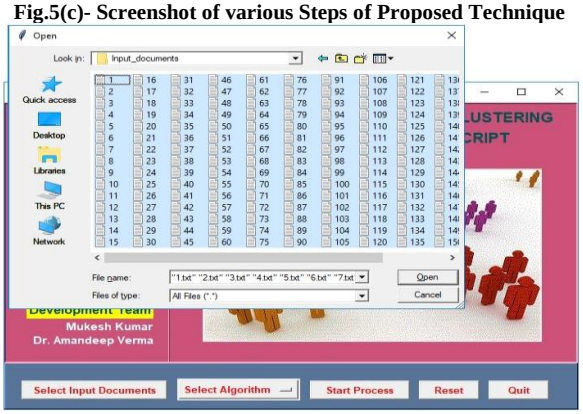

Fig.5(d)- Screenshot of Resultant Clusters as output

VI. CONCLUSIONS AND FUTURE SCOPE

The proposed hybrid algorithm for the development of document clustering technique for Gurmukhi script is first ever effort as no prior work has been done to develop domain independent clustering technique for Gurmukhi script.

The experimental results reveal that the proposed hybrid algorithm is more efficient and performs better than existing domain based Punjabi text document clustering system. In the development of proposed clustering system, the most challenging task is the creation of meaningful cluster titles. Some of the important aspects that should take into consideration, are given below:

^ Maintaining a list of stop words is continual process; as some new stop words occur during the submission of various types of input text documents in Gurmukhi script. A detailed list of stop words plays an important role in avoiding meaningless cluster title. To do so, apart from a list of 184 stop words [5], more number of stop words have been added afterwards.

^ Selection of an appropriate value of 'k' plays an important role in defining the list of cluster titles. Here, the value of 'k' represents number of selected top keywords arranged in descending order based upon term weight. For better results, set the value of 'k' to minimal when more number of input text documents are selected for clustering process. Whereas, it might be increased slightly when less number of input text documents are selected for clustering process.

^ It is assumed that number of input text documents belonging to each domain maintain nearby equal proportion otherwise it may not carry desired results. For an instance, out of 200 input text documents, if each lot of 10-12 text documents are concerned to specific domains; on the other hand only 2-3 text documents are concerned with another specific domain. In this situation, the key-terms from a domain consisting less number of documents may not fall under top 'k' selected key-terms sorted based upon term weight and the text documents from this specific domain might be placed under 'miscellaneous' cluster. Handling such an instance may be a task of future research.

Further improvements in document clustering technique for Gurmukhi script may lead to further research in developing more efficient normalization, stemming and noun/named entities extraction modules for text documents in Gurmukhi scripts. As these modules play a significant role proving an overall performance of clustering process.

Handling critical situations during the clustering process is also a subject of further research work. Considering an instance, where a fake text document, consisting only a term/sentence repeatedly along with an adequate number of key-terms also. If such fake documents are placed in a specific cluster, it results to low 'Precision' as in case of clusters '□□□□□□□□', '□□□', '□□□□□□', and '□□□□' in Table-3. Handling such instances is a subject of future research work.

REFERENCES

1. Ahmad A, Dey. L, K-Mean clustering algorithm for mixed numeric and categorical data, Data & Knowledge Engineering, 63(2), (2007) 503-527.
2. A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, ACM Computing Surveys, 31(3), (1999) 1-60.
3. A. Kakoti Mahanta, and Kabita Thaoroijam, A document clustering algorithm using a fuzzy approach", NCTAC 2007, Tezpur, (2007).
4. Caliński T, Harabasz J , A dendrite method for cluster analysis, Communications in Statistics-theory and Methods, 41(12),(2012), 1-27.
5. Jasleen Kaur,Dr. Jatinderkumar R. Saini, Punjabi stop words: a Gurmukhi, Shahmukhi and Roman scripted chronicle" ACM, WIR '16, Indore, India, ISBN 978-1-4503-4278-0/16/03, March 21-22, (2016).
6. Kishore C, Srinivasulu Asadi and Anusha G, Comparative study of software module clustering algorithms: hill-climbing, MCA and ECA, IJARCET, International Journal of Advanced Research in Computer Engineering & Technology, 1(3), (2012), 2278–1323.
7. Saurabh Sharma, Vishal Gupta, Punjabi documents clustering system, Journal of Emerging Technologies in Web Intelligence, 5(2), (2013), 171-187.
8. Srinivasulu Asadi and Dr.Ch.D.V.Subbarao, Clustering large data with mixed values using extended fuzzy adaptive resonance theory, Indonesian Journal of Electrical Engineering and Computer Science, 4(3), (2016), 617-628.
9. Srinivasulu Asadi, Dr. Ch.D.V.Subbarao and Y. Jeevan Kumar, Survey on cluster and classification analysis in hadoop mahout, ERCICA-2014, Elsevier Journal Digital Library, 7(3), (2014).
10. Vishal Gupta , Automatic stemming of words for Punjabi language, Advances in Intelligent Systems and Computing, Springer International Publishing Switzerland, (2014), 73-84.
11. Vishal Gupta , Automatic normalization of Punjabi words, International Journal of Engineering Trends and Technology (IJETT) – 6(7), (2013), 353-357.
12. Xiong, H, Wu, J & Chen J, K-means clustering versus validation measures: a data-distribution perspective, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 39(2),(2009), 318-331.
13. Link: http://kbcs.in/tools.html

AUTHORS PROFILE

Mukesh Kumar is working in PG Department of Computer Science, Mata Gujri College, Fatehgarh Sahib, Punjab, INDIA. He is currently pursuing Ph.D. from Punjabi University, Patiala. His area of research is document clustering for Gurumukhi script. He has published a number of books with Kalyani Publishers on the titles: Operating Systems, UNIX operating system, Object Oriented Programming Using C++ . He has 16 years of teaching experience.

Amandeep Verma is working as Assistanct Professor in PG Department of Computer science, Punjabi University Regional Centre for Information Technology and Management, Mohali Punjab, INDIA. He completed Ph.D. from Punjabi University, Patiala. His areas of research are: Ontological Engineering and Formal Methods, Image Processing and Cloud Computing. He has published number of research papers in reputed International journals.