Punjabi Documents Clustering System

Saurabh Sharma
University Institute of Engineering & Technology, Panjab University, Chandigarh, India
Email: saurabhsharma381@gmail.com

Vishal Gupta
University Institute of Engineering & Technology, Panjab University, Chandigarh, India
Email: vishal@pu.ac.in

***Abstract*—Text document clustering inherits its qualities from Natural Languages Processing, Machine Learning and Information Retrieval. For unsupervised document organization, automatic topic extraction and fast information filtering and accuracy in retrieval, this is an effective method. Many clustering algorithms are available for unsupervised document organization and its retrieval thereof. The documents for text clustering are merely considered as an assortment of words in traditional approaches to clustering. The semantic relationship of the words should form the decisive base for clustering, which is generally conveniently forgotten albeit the information is vital for the purpose. A new method for generating frequent phrases by analyzing the semantic relations between the words in a sentence is discussed. Karaka list captures the semantic relations, which is a grammatical connector for connecting Nouns, Pronouns and Verbs in a sentence. This new clustering method utilizes an amalgamation of the theories behind Karaka Analyzer, Frequent Item sets and Frequent Word Sequences. Results are indicative of the fact that New Hybrid approach performs better in terms of Number of Clusters, Meaningful label of Clusters and effectiveness of clustering for those documents which do not have desired information in frequent phrases. Use of semantic features is the key to better results.**

***Index Terms*—Punjabi Document Clustering, Karaka Theory, Frequent Phrases**

I. INTRODUCTION

With the onset of the revolution and knowledge explosion in the field of electronic resources, the documentation these days is ubiquitous in electronic form in addition to the normal documentation that exists on paper. The data/documentation in electronic form facilitates quick access to these electronic documents which are stored, usually in a very large text database. Considering the vast abundance of resources available on a vast text database like World Wide Web, exploration and utilization of these resources for information retrieval and text mining requires an effective solution. For search and retrieval from World Wide Web, with accuracy in the relevance of retrieved document, Web Search Engines are hugely accepted tools. Several methods have been projected to achieve this precision. Clustering the retrieval results before they are displayed to the user is advocated by one of the commonly used methods. The user may be interested in just one of the various results retrieved, by search engines that usually cover a variety of topics.

Text document clustering is a clustering technique which is specifically used for clustering of text document format. Clusters are formed and the text documents are grouped together in them on the basis of their similarities and into different groups on the basis of dissimilarities between them. The foundation of text document clustering is based on this concept.

Many effective clustering methods [1] on the structured or semi structured data are available however the way how to convert a document into structured data and the way how to calculate these structured data is essential as to clustering of text documents. The process of text document clustering can be separated into three sub-procedures namely, document representation, measurement and clustering. Document representation requires compression of the document. Swapping between the compression ratio and the integrality of the document features takes place. Structured data is the general output of this part. Various methods are employed to calculate different kinds of data. The measurement procedure smoothes this difference and by designing mathematical algorithms, a normalized start point is provided to the clustering process. The clustering procedure in the end gathers the documents into different groups. The user directly or indirectly sets a criteria and the number of groups are formed in accordance with this criteria. The way of document representation decides the success of text document clustering. Various techniques or methods are suggested by researchers to tackle this problem. Out of these options the simplest is N gram model where some words in document are selected to characterize the document. Often the stem of the selected words is used as a substitute for the words themselves in order to minimize the dimension of feature vector. A Very high effectiveness is the best advantage of this model. [2] [3]. The inherent meaning conveyed by the document is lost as N-gram model only uses bag of words. NLP extraction is also utilized by certain researchers to get phrases or part the sentences as the element of the representation such as sequences or basic element [4] [5]. Widyantoro [6] proposed the Fuzzy set and predication reasoning method in the clustering problem, and this idea was also propagated by Miyamoto [7] and Ridvan [8]. Based on Vector Space Model, most of the measurements are designed where the feature

doi:10.4304/jetwi.5.2.171-187

elements of document are organized as a vector. Design of measure method with respect to the weight age of every member of the vector is made. Some other measurements such as suffix tree were also adopted and they proved to be effective [5].Clustering methods can be generally categorize into two groups: hierarchy based methods and partitioning methods. Various algorithms that predominantly belong to the mentioned two categories are discussed. A generalized overlook of them and comparison among them has been studied by [9] [10].

II. Related work

A. Document Representation using Term Frequency

One of the most common methods for document representation considers the document as a bag of words. By using the frequency count of each word in the document, the document vector is created. The total number of distinct words in the whole set of documents to be clustered is the size of the vector. Methods that use the term count are discussed below.

1). Vector Space Model: The basic method of representing a document is by considering it an element in a vector space [11]. Each component of the vector is the frequency of occurrence of a word in the document. By fixing a logical criteria for the selection to shortlist a subset of most important words ,the size of the vector can be reduced. To find a suitable subset of words that represents the essential characteristics of the documents remains a difficult problem. Removal of non informative words is important, hence most common words like and, with, to etc., which are also known as stop words, are removed from the text during the process of creating the vector.

2). Word Category Maps: In the Self-organizing semantic map [12] method the words are clustered onto neighbouring grid points of a Self Organizing Map. Synonyms and closely related words are often mapped onto the same grid point or neighbouring grid points. This clustering scheme is even more effective than the thesaurus method where sets of synonyms are found manually. Adjacent words in the text taken over a moving window, forms the input to the self organizing map. All the words from all the documents are input interactively enough number of times to make the word category map. The next step involves labelling of each grid point is by all those words, the vector of which are mapped to that point. The grid points usually get multiple labels. For creating a Vector for a document, the words of the document are scanned and counted at those grid points of the SOM that were labelled by that word. These two methods identify a basic approach for representing a set of documents in the form of a vector which can be used effectively. The frequency count of different words is also weighted to improvise the performance of clustering. A comparative evaluation of feature selection methods for text documents is done by Yang and Pedersen [13].

3). Shortcomings of the Frequency Based Approach: Similar documents should also be similar in the representation. The representation of documents must reflect the knowledge meant to be conveyed by the documents. The methods that have been discussed above do not consider the semantic relations of the words for representation of documents. This gives rise to ambiguity in many cases where some combination of sentences, which have the same set of words having different meanings. These documents should fall in different clusters. For example, if we consider two sentences: Raj works as a manager in a Bank which is near the Tree house and There are lush green Trees on the Bank of the river that flows next to Raj's house, they have the same set of words but pertain to completely different meaning. There may be some sentences which have the same meaning but have different sets of words as constituents. This is prevalent in cases where synonymous words are used in the sentences. For example in the sentences, Beggar was famished and Beggar was starving convey more or less the same thing. A solution to such cases was proposed by Deerwester [14], which is known as Latent Semantic Indexing. Another method "Word Category Map" method can also be used. For a document even a word, which has a relatively lower frequency of occurrence in the document, can be more accurate in describing the document, whereas a word, which occurs more frequently, may have less importance. Frequency based methods do not take this into account. Semantic as well as the syntactic information present in the documents is required to be considered for solving the above problems.

B. Frequent Item sets Based Text Clustering

A new concept of frequent item sets was proposed [15] to overcome the drawback of VSM model [11]. Review of the literature reveals that frequent item sets based text clustering has received a lot of attention among the researchers. A method known as Frequent Item set-based Clustering with Window (FICW) which employs the semantic information for text clustering with a window constraint was presented by Zhou Chong [16]. FICW performed better in terms of both clustering accuracy and efficiency. This was depicted by the conclusions drawn from the experimental results obtained from three (hypertext) text sets. Frequent Term Set-based Clustering (FTSC) which is a text clustering algorithm and employs frequent term sets to cluster texts was proposed by Xiangwei Liu and Pilian He [17].The significant information is extracted from documents and stored into databases. To mine the frequent item sets Apriori was used. In the end, It clusters the documents as per the frequent words in subsets of the frequent term sets. To enhance the accuracy and speed of the clustering algorithm for extremely large databases, this algorithm can lessen the dimension of the text data FTSC and FTSHC algorithms are comparatively more efficient than K-Means algorithm in the clustering performance as per the experimental results obtained from these algorithms.

A simple hybrid algorithm (SHDC) on the basis of top-k frequent term sets and k-means was proposed by Le Wang [18].This was designed to overcome the main challenges of current web document clustering. Top-k frequent term sets were employed to provide k initial

means, which were regarded as initial clusters and later refined by k-means. The final optimal clustering was returned by k-means and the clear description of clustering was given by k frequent term sets. SHDC performed better other than two representative clustering algorithms (the farthest first k-means and random initial k-means) both on efficiency and effectiveness. These findings were drawn from the experimental results on two public datasets. Based on maximal frequent item sets, [19] introduced a web-text clustering method for personalized e-learning. Web documents were represented by vector space model, initially. Then, maximal frequent word sets were determined. Lastly, maximal item sets were employed for clustering documents on the basis of a new similarity measure of item sets. It was an effective and efficient method and the same was aptly supported by the experimental results. To cluster short documents in very large text database, Yongheng Wang [20] have introduced a frequent term based parallel clustering algorithm. To enhance the accuracy of clustering, a semantic classification method is also employed. The algorithm was more precise and efficient than other clustering algorithms when clustering large scale short documents. The algorithm has good scalability & can also be employed to process voluminous data. Liu and Zheng [21] proposed the documents clustering algorithm on the basis of frequent term sets. As per the Vector Space Model (VSM) every term is sorted in accordance with their relative frequency documents. Frequent term sets are, then, mined using frequent-pattern growth (FP growth). Finally, Documents are clustered on the basis of these frequent term sets. This approach gave a clear explanation of the determined clusters by their frequent term sets. It is efficient for very large databases. With the aid of experimental results, the efficiency and suitability of the proposed algorithm has been demonstrated. Henry Anaya-Sanchez [22] have proposed a clustering algorithm for Text Clustering based on Frequent Item sets for discovering and unfolding the topics included in a text collection. The algorithm depends on the most probable term pairs generated from the collection and on the estimation of the topic homogeneity related to these pairs. Topics and their descriptions whose support sets were homogeneous for denoting collection topics were produced from those term pairs. The efficacy and usefulness of the approach was demonstrated by the experimental results over three benchmark text collections. Florian Beil [2] proposed an approach which employed frequent item (term) sets for text clustering.

Algorithms for association rule mining were used to determine such frequent sets. The mutual overlap of frequent sets with regard to the sets of supporting documents was calibrated to cluster on the basis of frequent term sets. Two algorithms for frequent term-based text clustering were given, FTC which produced flat clustering and HFTC for hierarchical clustering. Clustering of more efficiency and comparable quality were obtained by the presented algorithms. These results were in consonance with the experimental results obtained from the test data.

C. Shortcomings of the Frequent Item set Based Approach.

The major limitation for the frequent item sets based approach is the difference in performance with respect to the language of the data. It performs better for English language but do not perform well for many Asian languages like Punjabi language. This can be attributed to the fact that the sentence structure of Punjabi is different from English. Frequent item sets created with this approach, do not represent the document correctly. It uses concept of "item sets", which was originally proposed for transactional databases [23] and was the main drawback of frequent item sets approach. The semantics [36] between terms is not considered by frequent item sets generated by this method and frequent item sets are created based on their frequent co-occurrences.

D. Indian Languages

Last few years have witnessed an increase in interest in Asian languages [24], especially those spoken in the Far East (e.g. Mandarin, Japanese, and Korean) and in the Indian subcontinent. Of late, with the increase in the amount of volume in term of websites and the number of internet pages available in these vernacular languages, It is required to have a better understanding of the procedures applied on them for processing. Many online users are working with these websites. To acquaint them and to draw the best usage out of these websites, an insight is required in this direction. The Indian subcontinent has a very rich tradition of many languages which have co-existed since time immemorial. The Indian constitution recognizes 18 languages that are spoken in various parts of the country. This general view however does not take into account that approximately 29 languages are spoken by more than 1.22 billion native speakers there, most of which have official status in the various Indian states. The situation in India is more complex than in Europe, as evidenced by the four main families to which the various languages belong: the Indo-European (more precisely the Indo-Aryan branch [25] including Bengali, Hindi, Marathi, and Punjabi among others) located mainly in the northern part, the Dravidian family (e.g. Kannada, Malayalam, Tamil, and Telugu) in the southern part, the Sino-Tibetan (e.g., Bodo and Manipur) in the northeastern part, and the Austra-Asiatic group (Santali) in the eastern part of this subcontinent, from a linguistic perspective. India's proportion of non-Indo-European languages is much greater than that of Europe. Moreover, compared to the three alphabets used in Europe (Latin, Greek, and Cyrillic), the various Indian languages use at least seven different writing systems "lipis" like Devnagari, Gurumukhi, kanarese (Dravidian family) etc.

E. Punjabi Language

Punjabi language is a member of the Indo-Aryan family of languages. Punjabi is spoken in India, Pakistan, USA, Canada, England, and other countries with Punjabi

immigrants. It is the official language of the state of Punjab in India. Punjabi is written in 'Gurmukhi' script in eastern Punjab (India), and in 'Shahmukhi' script in western Punjab (Pakistan). English language has a very complex system of derivational morphology but Punjabi does not have such a rich derivational morphology. However, it has a comparatively rich system of inflectional morphology when compared with English. It can be clear from the fact that a typical English verb usually has five different inflectional forms, e.g. forms of English verb 'go' are go, went, gone, goes, and going. On the other hand, an average Punjabi verb can take about 48 forms depending upon gender, number, person, tense, and aspect values, in a sentence. Some Punjabi verbs can have up to two causative forms and each such causative form will further have on an average 48 forms. Syntax and grammar of Punjabi language is entirely different from other languages in the world. Although, Statistical based language independent techniques can be easily applied for Punjabi Text Clustering. But linguistics based text clustering for Punjabi is entirely different from other languages of world as Punjabi is having entirely different syntax and grammar. For good clustering results both statistical and linguistic based approaches are needed. For Punjabi text clustering, most of the lexical resources are not available and need to be developed first.

F. Key Features of Punjabi Morphology

Punjabi descended from the Shauraseni language of medieval northern India and became a distinct language during the 11th century. In India, Punjabi is written with the Gurmukhi (ਗੁਰਮੁਖੀ) alphabet, while in Pakistan it is written with a version of the Urdu alphabet known as Shahmukhi. The written standard for Punjabi in both India and Pakistan is known as Majhi, which is named after the Majha region of Punjab. Punjabi is one of India's 22 official languages and it is the first official language in East Punjab. In Pakistan Punjabi is the second most widely spoken language but has no official status.

Gurumukhi script (**ਗੁਰਮੁਖੀ**) The Gurumukhi alphabet developed from the Landa alphabet and was standardized during the 16th century by Guru Angad Dev Ji, the second Sikh guru. The name Gurumukhi means "from the mouth of the Guru" and comes from the Old Punjabi word Gurumukhi. Notable Features of Gurumukhi are:

· Type of writing system: syllabic alphabet
· Direction of writing: left to right in horizontal lines
· Used to write: Punjabi

Consonants have an inherent vowel. Diacritics, which can appear above, below, before or after the consonant they belong to, are used to change the inherent vowel. When they appear at the beginning of a syllable, vowels are written as independent letters. When certain consonants occur together, special conjunct symbols are used which combine the essential parts of each letter. Punjabi is a tonal language with three tones. These are indicated in writing using the voiced aspirates consonants (gh, dh, bh, etc) and the intervocal h. The Shahmukhi alphabet is a version of the Urdu alphabet used to write Punjabi in Pakistan. It is normally written in Nastaʿlīq style and has been used since the second half of the 20th century. The name Shahmukhi means "from the King's mouth". Notable Features of Shahmukhi are:

· Type of writing system: alphabet
· Direction of writing: right to left in horizontal lines
· Used to write: Punjabi

The sounds ñ (ਞ), ng (ਙ), ṇ (ਣ) and nh (◌ੱ/◌ੰ) are all written with ں(noon ghunna). In initial and medial positions they are written with ن. ے(bari ye) is only found in the final position, when writing the sounds e (ਏ) or æ (ਐ), and in initial and medial positions, it takes the form of ۓ. Short vowels are written with: ◌ (ਅ), ◌ (ਉ), ◌ (ਇ): a, u, i. The term "Punjabi language" however does not refer to a well-defined and clearly standardized language but rather to a relatively large group of dialects wherein interlingual understanding is always possible (just as English is in the UK and the U.S.).

Punjabi sentence structure follows the Subject - Object - Verb (or SOV) pattern. Lexicons in Indian languages such as, Punjabi, is never free from the influence of other languages and vice versa. English language borrows some words from the Indian languages, such as "jungle" (from a Sanskrit stem), "punch" (drink, from Hindi or Marathi), "jute" (vegetable fibre, from Bengali) or "curry" (from the Tamil). Similarly, to a larger extent, many word forms in Indian languages are borrowed from English, especially given its dominant presence in commerce (e.g., time, taxi, company, bank, budget, ice cream, and gasoline) and in technology (e.g., computer and Internet).

Punjabi is written in Gurumukhi script, comprising 10 vowels and 38 simple consonants. Generally no distinction is made between uppercase and lowercase letters. In Punjabi grammar there are only two genders, masculine and feminine, while the neuter found in Sanskrit has disappeared. Feminine nouns are usually formed from the masculine, either by replacing the final '-ਆ' ('◌ਾ') by '-ਈ' ('◌ੀ') (e.g., "ਚਾਚਾ" (horse), "ਚਾਚੀ" (mare)) or by adding '-ਈ' for nouns ending with a consonant "ਤਿੱਤਰ" (partridge), "ਤਿੱਤਰੀ" (female partridge). Number is expressed through distinctive singular and plural forms. This language does not have a definite article (the), and instead of placing prepositions before the noun, it positions them after in the form of postpositions (e.g., "on the table"→"table on"). These are used in certain Western European languages such as German, as in the expression "den Fluss *entlang*" (*along* the river), while the use of this linguistic construction in other Indo-European languages is clearly the exception.

III. Problem Formulation And Objective Of The Study

Clustering of huge, diverse and rapidly changing text documents is a very complex task. The results thus achieved are largely dependent on the document set on

which clustering is applied and the parameters used for clustering criteria. Precise selection of the parameters is extremely vital for getting better clustering results. With the ongoing transformations in the field of online searching, clustering has gained much attention in last few years. However, the field of clustering holds many promises and much more research is still needed in this direction specifically in vernaculars like Punjabi language.

A. Objectives

The study was undertaken specifically for Clustering of text documents in Punjabi Language as no prior work has been done in this language as per review of literature done to carry out this study. It is a first ever attempt in this direction to provide a solution for text clustering in Punjabi Language, by proposing a new hybrid approach, which will be immensely useful to the researchers who will wish to undertake study and research in vernacular languages.

1. To implement existing Item set based Data Clustering Algorithms, namely: Clustering with frequent Item Sets and Clustering with Frequent Word Sequences, for Punjabi text documents.
2. Propose and implement a new algorithm for clustering of Punjabi text documents by combining best features of the algorithms mentioned above keeping in view the semantics of Punjabi language.
3. Compare the Efficiency of the three algorithms for Punjabi Text Document Clustering using: Precision, Recall, F-Measure, Number of Clusters, Percentage of unrecognized documents and Processing Time.

IV. Punjabi Text Clustering Using Frequent Item sets

This method [26] utilizes the concept of cohesiveness of a cluster directly by making frequent item sets. These item sets are identified on the basis of the cohesiveness i.e. union of parts of the same kind that exists between the terms used in the data. The documents under the same topic share more cohesiveness and hence more common item sets than those which are grouped under different topics.

Global Frequent Item set: A global frequent item set is a set of items that appear together in more than a minimum fraction of the whole document set. A minimum global support, in a percentage of all documents, can be specified for this purpose. It uses Apripori algorithm presented by Agrawal [23] for finding global frequent item sets. It is important to note here that item sets are found based on word presence, not on TF and IDF.

Global Frequent Item: A global frequent item refers to an item that belongs to some global frequent item set. A global frequent item set containing k items is called a global frequent k-item set.

Global Support: The global support of an item set is the percentage of documents containing the item set. A global frequent item is cluster frequent in a cluster Ci if the item is contained in some minimum fraction of documents in Ci. A minimum cluster support, in a percentage of the documents in Ci, can be specified for this purpose.

Cluster Support: The cluster support of an item in Ci is the percentage of the documents in Ci that contain the item. This method constructs clusters in two steps: constructing initial clusters, then making initial clusters disjoint.

A. Constructing Initial Clusters

An initial cluster to contain all the documents, that contain this Item set, is constructed for each global frequent item set. One document may contain several global frequent item sets hence the initial clusters are not disjoint. The next step involves the removal of overlapping of clusters. An important property of initial clusters is that all documents in a cluster contain all the items in the global frequent item set that defines the cluster, i.e. these items are mandatory for the cluster. To identify the cluster, global frequent item set is used as the cluster label.

TABLE I.
DOCUMENT SETS

Doc. name	Feature vector (ਟੇਸਟ, ਮੈਚ, ਟੀਮ, ਖੇਡ, ਖਿਡਾਰੀ, ਹਾਕੀ)					
Cric5	(1	1	2	0	0	0)
Cric14	(1	1	1	0	0	0)
Cric15	(1	1	1	0	0	0)
Cric16	(1	1	0	0	0	0)
Cric34	(1	0	0	0	0	0)
Cric12	(0	1	0	0	0	0)
Fball4	(0	0	1	1	0	0)
Fball13	(0	0	2	0	0	0)
Fball16	(0	0	0	1	0	0)
Fball17	(0	1	1	1	1	0)
Hoky12	(0	2	1	0	0	1)
Hoky15	(0	1	1	0	0	1)

B. Making Clusters Disjoint

In this step, clusters are made disjoint. For each document, it identifies the “best” initial cluster and keeps the document only in the best initial cluster.

TABLE II.
ITEM SETS

Itemset	Global support
{ਟੇਸਟ}	42%
{ਮੈਚ}	67%
{ਟੀਮ}	67%
{ਖੇਡ}	25%
{ਖਿਡਾਰੀ}	08%
{ਹਾਕੀ}	17%
{ਟੇਸਟ,ਮੈਚ}	54%
{ ਟੀਮ, ਮੈਚ}	67%
{ਟੇਸਟ, ਟੀਮ}	54%

TABLE III.
GLOBAL FREQUENT ITEM SETS (MINIMUM GLOBAL SUPPORT=35%)

Global frequent item set	Global support
{ ਟੇਸਟ}	42%
{ਮੈਚ}	67%
{ਟੀਮ}	67%
{ਟੇਸਟ, ਮੈਚ}	54%
{ਮੈਚ, ਟੀਮ}	67%
{ਟੇਸਟ, ਟੀਮ}	54%

Suppose that *Score*($C_i \leftarrow doc_j$) measures the goodness of a cluster C_i for a document doc_j . For each doc_j , we remove doc_j from all the initial clusters C_i that contain doc_j but one for which *Score*($C_i \leftarrow doc_j$) is maximized. If there are more than one C_i that maximizes *Score*($C_i \leftarrow doc_j$), choose the one that has the most number of items in the cluster label. After this step, each document belongs to exactly one cluster. Now, we define the score function *Score*($C_i \leftarrow doc_j$). Score($C_i \leftarrow doc_j$) = [$\sum$ n(x) * cluster_support(x)] - $\sum$ [n(x') * global_support(x')]
where x represents a global frequent item in doc_j and the item is also cluster frequent in C_i ; x' represents a global frequent item in doc_j that is not cluster frequent in C_i ; $n(x)$ and $n(x')$ are the weighted frequency of x and x' in the feature vector of doc_j. The cluster label is a set of mandatory items in the cluster. Every document in the cluster must contain all the items in the cluster label. This forms a major distinguishing factor in the cluster label and the set of cluster frequent items associated with a cluster. To construct an initial cluster and to identify the cluster, cluster label is used. Whereas, some minimum proportion of documents in a cluster should contain the cluster frequent item. The topic description of the cluster is therefore derived from the cluster frequent items. Re-computation of the cluster frequent items is necessary for each cluster as it will help to reveal the updated clustering because some documents are removed from initial clusters. While re-computing the cluster frequent items of a cluster Ci, also include all the documents from all "descendants" of Ci. A cluster is a descendant of Ci if its cluster label is a superset of the cluster label of Ci. The rationale is that descendants are likely to be subtopics of a parent; therefore, it is sensible to include them.

C. Tree Construction

This step helps to understand the hierarchy of the process and hence a hierarchical cluster tree is constructed in consonance with this idea. The resulting hierarchical tree has two major objectives behind its creation namely, to form a foundation for pruning & to provide a natural structure for browsing. Each cluster has exactly one parent in the cluster tree. There is an exception to this general rule in the case of cluster with the empty cluster label. Parent cluster has a more general topic than a child cluster. Albeit different, they are "similar" to a certain degree. It is important to remember here that each cluster uses one global frequent k-item set as its cluster label. Such clusters are called k-clusters. The root node appears at level 0 in cluster tree, which corresponds to the cluster with the cluster label "null". This node collects the un clustered documents. However when the actual user interface is run, the un clustered documents are put in a cluster marked "Unrecognized" at level 1. The 1- clusters appear in level 1 of the tree, and so forth for every level.

D. Tree Pruning, Child Pruning and Sibling Merging

When a small minimum global support is used, the cluster tree thus formed can be broad and deep. It makes the documents of the same topic, distributed over several small clusters, which eventually lead to poor clustering accuracy. This step aims at merging similar clusters in order to produce a natural topic hierarchy for browsing and it also enhances the clustering accuracy. Creation of final clusters is the outcome of this step.

V. PUNJABI TEXT CLUSTERING USING FREQUENT WORD SEQUENCES

In this method [5], sequence of words constitutes a text document d, so that it can be represented as d = (w1, w2, w3 . . .), where w1, w2, w3 . . . are words appearing in d. Like a frequent item set in the association rule mining of a transaction data set [23], a word set is frequent when its support is at least the user-specified minimum support. It implies that, at least the specified minimum number (or percentage) of documents are available that contain this word set. A frequent k-word set is a frequent word set containing k words. All the words in frequent 2-word sets

are called frequent words. Definition 5.1 (Word Sequence): An ordered sequence of two or more words is called a word sequence. A word sequence S is represented as (w1, w2 . . .). A frequent word sequence is denoted by FS. For example, FS = (w1, w2, w3, w4), in which w2 is not necessarily following w1 immediately in a text document. There could be words between them as long as w2 is after w1 and the words between them are not frequent. A text document d supports this word sequence if these four words (w1, w2, w3, and w4) appear in d in the specified order. A word sequence S is an FS when there are at least the specified minimum numbers (or percentage) of documents supporting S. Multiple occurrences of a sequence in the same document is counted as one.

A. Finding frequent 2-word sets

Reduction in the dimension of the database (i.e., the number of unique words) by eliminating those words which are not frequent enough to be in a frequent k-word sequence, for k $\geq$ 2, is the major aim of this step. An association rule miner is used to find the frequent 2-word sets that satisfy the minimum support. All the words in frequent 2-word sets are put into a set WS. Members of the frequent word sequences of all length k, k $\geq$ 2, must be in WS. After finding the frequent 2-word sets, we remove all the words in the documents that are not in WS. After the removal, the resulting documents are called compact documents.

B. Building a generalized suffix tree (GST)

To find the frequent word sequences of the database is the intent of this step. The suffix tree [27], a well known data structure for sequence pattern matching is used, to find all the frequent word sequences. Each compact document is treated as a string of words and inserted into a generalized suffix tree (GST) one by one. It can find all the frequent word sequences of the database by collecting the information stored in all the nodes of the GST. We try and understand this algorithm for Punjabi Text Documents by taking an example. D = {d1, d2, d3};

- d1: ਜਵਾਨ ਮੁੰਡੇ ਬਾਸ੍ਕੇਟਬਾਲ ਖੇਡਣਾ ਪਸੰਦ ਕਰਦੇ ਹਨ| (Young boys like to play basketball.)
- d2:.ਅਧੇ ਜਵਾਨ ਮੁੰਡੇ ਫੁਟਬਾਲ ਖੇਡਦੇ ਹਨ|(Half of young boys play football)
- d3: ਲਗਭਗ ਸਾਰੇ ਜਵਾਨ ਮੁੰਡੇ ਬਾਸ੍ਕੇਟਬਾਲ ਖੇਡਦੇ ਹਨ | (Almost all boys play basketball.)

There are 9 unique words in this database D: { ਸਾਰੇ, ਲਗਭਗ, ਬਾਸ੍ਕੇਟਬਾਲ, ਮੁੰਡੇ, ਫੁਟਬਾਲ, ਅਧੇ, ਪਸੰਦ, ਖੇਡਣਾ, ਜਵਾਨ} {all, almost, basketball, boys, football, half, like, play, young}. If we specify the minimum support as 60%, the minimum support count is 2 for this case. The set of frequent 2-word sets is {{young, boys} {ਜਵਾਨ, ਮੁੰਡੇ}, {boys, play} {ਮੁੰਡੇ, ਖੇਡਣਾ}, {boys, basketball} {ਮੁੰਡੇ, ਬਾਸ੍ਕੇਟਬਾਲ}, {young, play} {ਜਵਾਨ, ਖੇਡਣਾ}, {play, basketball} {ਖੇਡਣਾ, ਬਾਸ੍ਕੇਟਬਾਲ}}; and WS = {ਜਵਾਨ, ਮੁੰਡੇ, ਖੇਡਣਾ, ਬਾਸ੍ਕੇਟਬਾਲ}, {young, boys, play, basketball}. After removing those words not in WS, the database D becomes D'={d'1, d'2, d'3} as follows, where the removed words are shown in parentheses.

d'_1 = ਜਵਾਨ ਮੁੰਡੇ ਬਾਸ੍ਕੇਟਬਾਲ ਖੇਡਣਾ (ਪਸੰਦ) ਕਰਦੇ ਹਨ |(Young boys (like to) play basketball;)

d'_2 = (ਅਧੇ) ਜਵਾਨ ਮੁੰਡੇ(ਫੁਟਬਾਲ), ਖੇਡਦੇ ਹਨ |((Half of) young boys play (football);)

d'_3 = (ਲਗਭਗ ਸਾਰੇ)ਜਵਾਨ ਮੁੰਡੇ ਬਾਸ੍ਕੇਟਬਾਲ ਖੇਡਦੇ ਹਨ |((Almost all) boys play basketball;)

The example shows that the dimension of D is reduced from 11 to 4 which is a major deciding factor on our next step in which the generalized suffix tree is generated for the database D'.

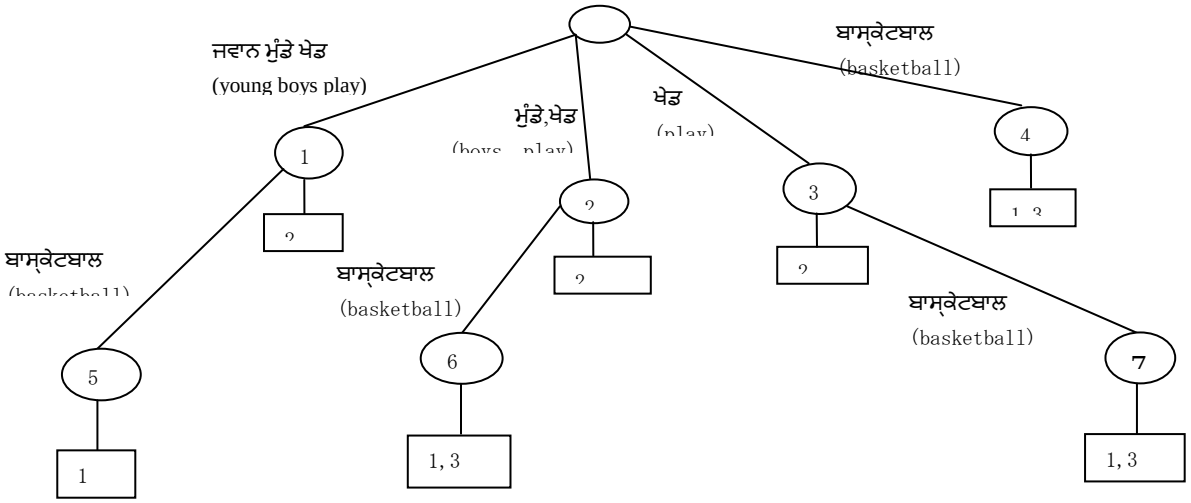

Figure 1. Generalized suffix tree for the compact document

A generalized suffix tree (GST) is a suffix tree that combines of a set of strings. In the present case, we build a GST of all the compact documents in the text database.

C. Finding frequent word sequences and collecting the cluster candidates

After building the GST, perform the depth-first traverse to collect all the cluster candidates for the database. Only the suffix nodes representing frequent word sequences can produce the cluster candidates.

D. Merging the cluster candidates based on the k-mismatch concept

The mismatches between the frequent word sequences found by building the GST of the document collection are checked. Three types of mismatches that can happen between two frequent word sequences FS_i and FS_j: insertion, deletion and substitution. Insertion means that by inserting k words into the shorter pattern FS_i, it becomes the longer pattern FSj. Deletion means that by deleting k words from the longer pattern FS_i, it becomes the shorter pattern FSj. Substitution is the relationship between two patterns, FS_i and FS_j, of the same length, such that by substituting k words in FS_i, it becomes FS_j.

E. Combining the overlapping clusters

After we merge cluster candidates into clusters, it may be observed that some clusters have too much overlap between their documents. If overlapping is larger than the specified overlap threshold value d, these two clusters are combined into one cluster. Obviously, the range of d is [0, 1]: When d = 0, these two clusters are disjoint; and when d = 1, these two clusters have the same set of documents, which does not mean these two clusters are identical because this set of documents may cover two different topics.

VI. Proposed Approach For Punjabi Text Clustering

The languages termed as Positional languages, which come in the category of Context Free Grammars (CFGs) have used all these approaches that have been discussed above. A context-free grammar is a formal system that describes a language by specifying how any legal text can be derived from a distinguished symbol called the axiom, or sentence symbol. It consists of a set of productions, each of which states that a given symbol can be replaced by a given sequence of symbols. The sentence structure of Punjabi is different as it belongs to the category of free order language, unlike in English. Hence, features of free order languages were to be taken into consideration for clustering of Punjabi text.

A majority [28] of human languages have relatively free word order. It includes Indian and other languages also. Order of words contains only secondary information such as emphasis etc. in free word order languages. Primary information pertaining to ‘gross’ meaning (e.g., one that includes semantic relationships) may not be in the exact word but is contained elsewhere. Most existing computational grammars are positional grammars & are based on context free grammars. Use of suitable computational grammar formalism for free word order languages is important for following two reasons:

1. A suitably designed formalism will be more efficient because it will be able to make use of primary sources of information directly.

2. Such formalism is also likely to be linguistically more elegant and satisfying. Since it will be able to relate to primary sources of information, the grammar is likely to be more economical and easier to write. Paninian framework, formalism, has been used for extraction of phrases. It has been successfully applied to Indian languages. A Karaka relation between verbs and nouns in a sentence is used to analyse the sentence. The notion of karaka relations is central to the Paninian model. Sudhir K Mishra [29] whose work focused on the theory of Karaka, introduced by Panini in his work in Adhikara sutra [30], for analyzing the structure of a sentence in Sanskrit Language did the prominent work in this category.

It is often a misnomer that karakas are similar to cases in English, although, they are fundamentally different: "The pivotal categories of the abstract syntactic representation are the karakas, the grammatical functions assigned to nominal’s in relation to the verbal root. They are neither semantic nor morphological categories in themselves but correspond to semantics according to rules specified in the grammar and to morphology according to other rules specified in the grammar [31]. Karaka denotes the relationship between a noun and a verb in a sentence and it literally means 'that which brings about' or the 'doer' [28]. Any factor that contributes to the accomplishment of any action, Punjabi language identifies eight sub types like Hindi and Sanskrit [28]. The karaka relations are syntactico-semantic (or semantico-syntactic) relations between the verbal and other related constituents in a sentence. They by themselves do not give the semantics. Instead they specify relations which mediate between vibhakti of nominal’s and verb forms on one hand and semantic relations on the other [31] [32] [33]. Two of the important karakas are “karta karaka” and “karma karaka”. Frequently, the karta karaka maps to agent theta role, and the karma karaka to theme or goal theta role.

Table IV.
Karaka Case Markers

Classical Case	Punjabi Karakas	Case Markers
Nominative Case	Karta karaka	ਨੇ (nē)
Accusative Case	Karma karaka	ਨੂੰ (nūṃ)
Instrumental Case	Karan karaka	ਨਾਲ (nāl)
Dative Case	Sampradaan karaka	ਲਈ (laī)

Ablative Case	Apaadaan karaka	ਤੋਂ (tōṃ)
Genitive Case	Sambandh karaka	ਦਾ/ਦੇ (dā/dē)
Case of Time-Place	Adhikaran karaka	ਪਾਸ, ਕੋਲ (pās, kōl)

Karta karaka is that participant in the action that is most independent. As part of this framework, a mapping is specified between karaka relations and case markers (which covers collectively case endings, post-positional markers, etc.) [28]. Case markers are used to identify useful phrases.

A. *Preprocessing Phase*

Preprocessing is defined as number of steps applied on the input text for converting it from free text to a structured format, which is the basic requirement of any clustering algorithm. In text clustering, some techniques used in preprocessing are removal of punctuation marks, removal of stop words, stemming of words, normalization (where the same word exists in different spellings in case of multilingual words).

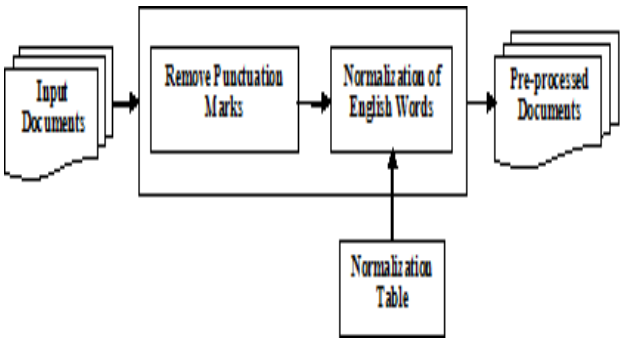

Figure 2. Pre-processing Phase of Hybrid Algorithm

For pre-processing, the Algorithm takes Punjabi text documents as input. The first step in pre-processing comprises of removal of punctuation marks. Stop words are not removed, since Karaka theory [30] is being used for generating phrases. Karaka theory works only on complete sentences that necessarily include stop words. This does away with the requirement of removal of stop words. Next step is normalization of those words which are used with different spellings.

For example, many words in Punjabi are borrowed from English, which are written as Transliteration of original spelling, without translating it into Punjabi. For those words, we have manually prepared a Normalization table, in which different spellings for a single word is stored. During Normalization step, each word is checked in normalization table for all different spellings. If a match found, then it will be replaced by spelling1 of that word. Purpose of normalization is to maintain uniformity of spelling in all documents which contain that word. This helps in better clustering results. Otherwise some documents may not be identified just because of the difference in spellings.

TABLE V.
PART OF NORMALIZATION TABLE

English Word	Spelling 1	Spelling 2	Spelling 3	Spelling 4
Boxing	ਮੁੱਕੇਬਾਜ਼ੀ	ਮੁਕੇਬਾਜੀ	ਮੁਕੇਬਾਜ਼ੀ	ਮੁੱਕੇਬਾਜੀ
Cricket	ਕ੍ਰਿਕੇਟ	ਕ੍ਰਿਕਟ	---	---
Football	ਫੁਟਬਾਲ	ਫੁੱਟਬਾਲ	ਫ਼ੁੱਟਬਾਲ	---
Hockey	ਹਾਕੀ	ਹਾੱਕੀ	---	---
Tennis	ਟੈਨਿਸ	ਟੇਨਿਸ	---	---

B. *Processing Phase*

Definition 6.1 (Karaka Symbols). Karaka symbols can be defined as any of various words in languages such as Hindi, Punjabi, Japanese, Hungarian, and Finnish which serve the same purpose as the preposition but comes after the noun. In other words, a word that show the relation of a noun and pronoun to other words in a sentence, similar in function to preposition but it follows rather that proceeds the object. Definition 6.2 (Karaka List). The Karaka List is the collection of Karaka symbols which are used to identify the phrases from a sentence.

1) Algorithm Details: After the pre-processing step is complete, phrases are extracted from sentences with the help of karaka list. Karaka List is the collection of words which are used to specify role of words as nouns, verbs, objects and gives information about semantics of the sentence.

The main purpose of using Karaka list is to overcome the drawback of Frequent Item Sets [26] and Frequent Word Sequences [5], generating long Sequences by trying all combinations of 2-word sequences, using Apriori algorithm [23]. For example, a 4-word sequence in input files, "Panjab University Football Club".

ਪੰਜਾਬ ਯੂਨੀਵਰਸਿਟੀ ਫੁਟਬਾਲ ਕਲੱਬ ਨੇ ਫੁਟਬਾਲ ਕਪ ਜਿੱਤੀਆ

Pañjāb yūnīvrasiṭī phūṭbāl kalabb (nē) phuṭbāl kap jītiyā (panjab university football club won the football cup.)

In case of Frequent sequence algorithms, after pre-processing step, using Apriori algorithm, initially it has

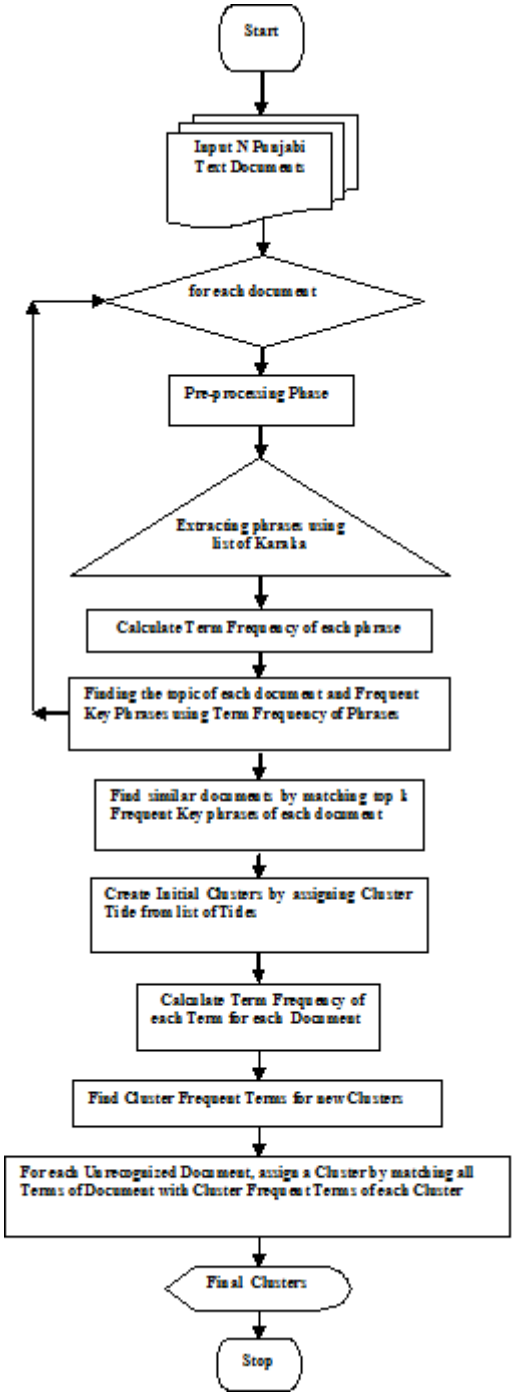

Figure 3. Flowchart of Hybrid Algorithm

2-word sequences, {Panjab University},{University Football} and {Football, Club}. Then, it try to find longer sequence of 3-word length by combination of 2-word sequences {Panjab University Football}, {University Football Club}. Finally, original 4-word sequence is found. In proposed hybrid algorithm, by using Karaka List, it break the sentence into phrases when a Karaka symbol is found and discard the Karaka symbol. In the above example, in a single step two phrases are generated from the input sentence with the advantage that this 4 word sequence is generated in a single step. Phrases from sentences:

{ਪੰਜਾਬ ਯੂਨੀਵਰਸਿਟੀ ਫੁਟਬਾਲ ਕਲੱਬ}, {ਫੁਟਬਾਲ ਕਪ ਜੀਤਾ}

{panjāb yūnīvrasiṭī phūṭbāl kalabb},{phuṭbāl kap jītā}

{Panjab University Football Club}, {won football cup}

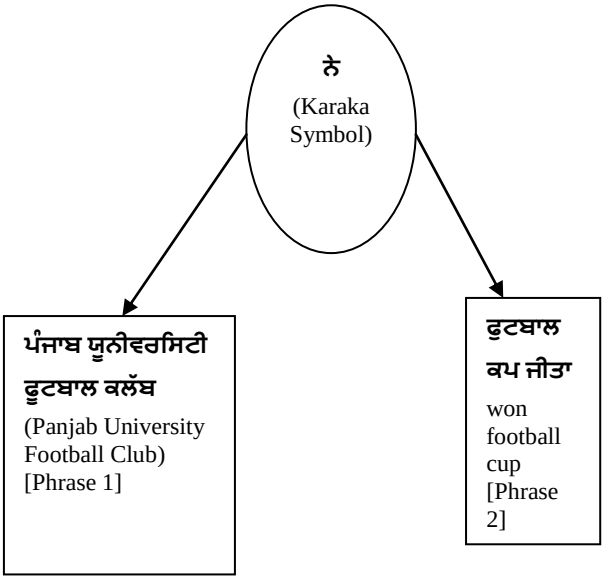

Figure 4. Extraction of Phrases using Karaka Symbol

Extraction of phrases from the document with the help of Karaka list generates a document vector containing phrases of various lengths as they were originally in input document. This dissuades the computation of k-length sequences in number of steps by trying all possible combinations of (k-1)-length sequences.

2) Calculate Term frequency of Phrases: Term Frequency is a numerical statistic which reflects how important a word is to a document in a collection. It is often used as a weighting factor in information retrieval and text mining. The value of Term Frequency increases proportionally to the number of times a word appears in the document which helps to control for the fact that some words are generally more common than others. For each phrase, we calculate the Term Frequency, by counting the total number of occurrence in the document.

3) Find top k Frequent Phrases: Sort all phrases by Term Frequency in descending order. Then declare top k phrases as Key phrases. These key phrases will be used for finding similarity among all other documents. The value of k is a very important factor for better clustering results. The valid value of k ranges from 1 to n, where n is number of phrases in a document. For experimental results, 20% of phrases are used as value of k.

4) Finding Similar Documents and Creating Initial Clusters: In this step, initial clusters are created by matching key phrases of documents with each other. If a phrase is found common between two files, then it is assumed that these files may belong to the same cluster. All matched files will be searched for each Cluster Title in the list.

5) Definition 6.3 (Cluster Title List): Cluster Title List is a list of words which are candidate terms for a cluster

title. For example, in a real life scenario, a user may wish to cluster documents from sports domain, which will be generically based broadly on common popular sports name. It therefore, becomes meaningful to provide clustering results, having small number of clusters with name of sports as cluster titles rather than generating huge number of clusters with meaningless cluster titles. Using cluster labeling by applying statistical techniques which is done by identifying "important" terms in the text that best represent the cluster topic, is normally done by the prevalent algorithms in this field. Many times, the list of significant keywords, or even phrases, will fail to provide a meaningful readable label for a set of documents. Often, the suggested terms tend to represent different aspects of the topic underlying the cluster, even when related to each other. Moreover, a good label may not occur directly in the text. Hence, to deduce a proper label from the suggested terms to successfully describe the cluster's topic, user intervention is required [34].

The main idea of using Cluster Title List is to avoid meaningless or ambiguous titles of Clusters. To avoid this major drawback, in which huge number of clusters with meaningless titles or multiple clusters on same topic are created, manually created list of Cluster Titles for specific domain have been used. Sports domain has been selected for conducting experiments on test data. List of Cluster Titles specific to sports have been created manually as no such list is available for Punjabi language. Files with same Cluster Title are placed into same cluster. If two files contain matching phrase but do not contain same Cluster Title, then it is assumed that both files do not belong to same cluster. One important property of initial clusters is that all documents in a cluster must contain Cluster Title that defines the cluster, i.e. Cluster Title is mandatory for each document of the cluster. Advantage of this property is that precision of each initial cluster is always equal to 1.

6) Calculate Term Frequency of each Term for each Document and Sort them to find Top k frequent terms: After creating initial clusters, all those files which are not placed in any cluster, are placed in a cluster named "Unrecognized". Since, some files may contain cluster titles but did not appear in top k Frequent Phrases, for those unrecognized files, VSM model is used i.e. now document is represented as a collection of terms obtained from all phrases. For each unrecognized document, Term Frequency for each term in the document is calculated. Then, all terms are sorted based on their Term Frequency in document, to find top k frequent terms of the document. The value of k can be varied as per the users discretion from 5%, 10%, 20% and so on. Higher the value of k, more terms will be considered for finding cluster for unrecognized document. Higher value is beneficial for those documents in which term frequency of cluster title is very low. Higher value of k shows better results as compared to low value of k. For experimental results, 20% of phrases are used as value of k.

7) Find Cluster Frequent Terms for new Clusters: After calculating top k frequent terms for each unrecognized document. Now, top k Cluster Frequent Terms for each cluster will be identified. Cluster Frequent Term is defined as the Term which appears in at least 80% of documents in a cluster. Cluster is treated as a conceptual document (by combining all terms of all documents in a cluster) for finding cluster frequent terms. Calculate Term Frequency of each term of the conceptual document. Top k cluster frequent terms by sorting all terms using their Term Frequency will be identified and used for the next step.

8) For each Unrecognized Document, assign a Cluster by matching all Terms of Document with Cluster Frequent Terms of each Cluster: Cluster for unrecognized document, by matching top k Frequent Terms of document with top k Cluster Frequent Terms of each document, is identified. If a match found with a cluster, then document is moved from Unrecognized cluster to that identified cluster. If a match is found with more than one cluster, then the number of matched terms for each cluster is counted. Document is placed in that cluster, which has maximum number of matched terms.

9) Final Clusters: After processing of unrecognized documents, final clusters containing documents from initial cluster and documents from unrecognized documents are created.

VII. Experimental Evaluation

This section discusses the experimental evaluation of our proposed algorithm on our test data sets.

A. Data Set

The text documents are denoted as unstructured data. It is very complex to group text documents. The document clustering requires a pre-processing task to convert the unstructured data values into a structured one. The documents are data elements with large dimensions. The system was tested for 221 text documents collected from various Punjabi News websites which comprised of news articles on sports. This dataset was categorized into 7 Natural classes, which were used for the evaluation of all three algorithms.

B. Experimental Results and Discussion

To evaluate the accuracy of the clustering results generated by clustering algorithms, F-measure is employed. A commonly used external measurement method; it is a standard evaluation method for both flat and hierarchical clustering structures. Let us assume that each cluster is treated as if it were the result of a query and each natural class is treated as if it were the relevant set of documents for a query. The recall, precision, and F-measure for natural class K_i and cluster C_j are calculated as follows:

$$\text{Precision}(K_i, C_j) = n_{ij} / |C_j| \quad (2)$$

$$\text{Recall}(K_i, C_j) = n_{ij} / |K_j| \quad (3)$$

$$\text{F-Measure}(K_i, C_j) =$$

$$\frac{2 * [Precision(K_i, C_j) * Recall(K_i, C_j)]}{[Precision(K_i, C_j) + Recall(K_i, C_j)]} \quad (4)$$

where n_{ij} is the number of members of natural class K_i in cluster C_j. Intuitively, $F(K_i;C_j)$ measures the quality of cluster C_j in describing the natural class K_i, by the harmonic mean of Recall and Precision for the "query results" C_j with respect to the "relevant documents" K_i. This is further illustrated in table 3:

TABLE VI.
CLUSTERS FOR FREQUENT ITEM SETS

Cluster	Precision	Recall	F-Measure
ਹਾਕੀ (Hockey)	1.00	0.20	0.33
ਖਿਡਾਰੀ (khidari)	0.33	0.20	0.25
ਖਿਤਾਬ (Khitaab)	0.73	0.27	0.39
ਖੇਡ (khed)	0.35	0.21	0.26
ਟੀਮ (team)	0.42	0.31	0.36
ਟੂਰਨਾਮੈਂਟ (tournament)	0.33	0.06	0.10
ਤਮਗਾ (tamga)	0.67	0.50	0.57
ਦਿੱਲੀ (dilli)	1.00	0.05	0.10
ਭਾਰਤ (bharat)	0.43	0.11	0.18
ਮੈਚ (match)	0.44	0.20	0.27
ਰਾਸ਼ਟਰਮੰਡਲ (rashtarmandal)	1.00	0.21	0.35

TABLE VII.
CLUSTERS FOR FREQUENT WORD SEQUENCES

Cluster	Precision	Recall	F-Measure
ਸੈਸ਼ਨ ਹਾਲ (session Hall)	1.00	0.04	0.08
ਹਾਕੀ (Hockey)	1.00	0.17	0.29
ਖਿਡਾਰੀ (khidari)	0.55	0.40	0.46
ਖਿਤਾਬ (Khitaab)	0.57	0.20	0.30
ਖੇਡ (khed)	0.50	0.14	0.22
ਚੰਦ (chand)	0.67	0.06	0.11
ਚੌਂਕ (chaunk)	1.00	0.05	0.10
ਜੋੜੀ (jodi)	1.00	0.13	0.24
ਟੀਮ (team)	0.41	0.39	0.40
ਟੂਰਨਾਮੈਂਟ (tournament)	0.57	0.20	0.30
ਟੈਨਿਸ (Tennis)	1.00	0.20	0.33
ਤਮਗਾ (tamga)	0.33	0.25	0.29
ਦਿੱਲੀ (dilli)	1.00	0.53	0.69
ਦੁਨੀਆ ਮਹੇਸ਼ (dunia mahesh)	1.00	0.10	0.18
ਪ੍ਰਦਰਸ਼ਨ (Pradarshan)	0.50	0.05	0.10
ਪੁਰਤਗਾਲ (Purtgaal)	1.00	0.09	0.16
ਬੋਰਡ (board)	1.00	0.07	0.13
ਭਾਰਤ (bharat)	0.55	0.25	0.34
ਮਿੰਟ ਕਾਰਨਰ (mint corner)	1.00	0.02	0.04
ਮੈਚ (match)	0.69	0.15	0.25
ਰੂਸ ਹਾਰ (roos haar)	0.50	0.03	0.06

TABLE VIII.
CLUSTERS FOR HYBRID APPROACH

Cluster	Precision	Recall	F-Measure
ਹਾਕੀ (Hockey)	0.97	1.00	0.99
ਕ੍ਰਿਕਟ (Cricket)	0.97	0.89	0.93
ਟੈਨਿਸ (Tennis)	1.00	0.77	0.87
ਫੁਟਬਾਲ (Football)	0.98	0.89	0.93
ਬੈਡਮਿੰਟਨ (Badminton)	1.00	1.00	1.00
ਮੁੱਕੇਬਾਜ਼ੀ (Boxing)	0.80	1.00	0.89

TABLE IX.
OVERALL EFFICIENCY OF ALGORITHMS

Clustering Algorithm	Precision	Recall	F-Measure
Frequent Itemsets	0.61	0.21	0.29
Frequent Word Sequences	0.75	0.17	0.24
Hybrid Approach	0.95	0.92	0.93

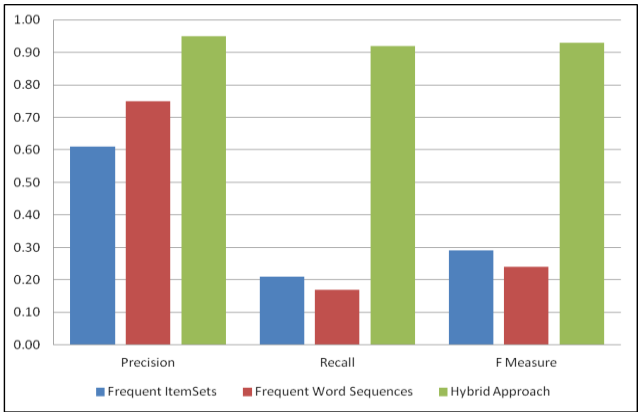

Figure 5. Precision, Recall and F-Measure

In Fig 5, the graph plotted for Precision, Recall and F-Measure for all the three algorithms that were studied for clustering of Punjabi text documents, the two algorithms namely, Frequent Item set and Frequent word sequence, shows a good precision but a very poor recall value. This leads to a very low value of F-Measure which is indicative of its overall poor performance. On the other

hand, Hybrid algorithm that shows good Precision, Recall and F-Measure, outperform other two algorithms and hence generate best clustering results.

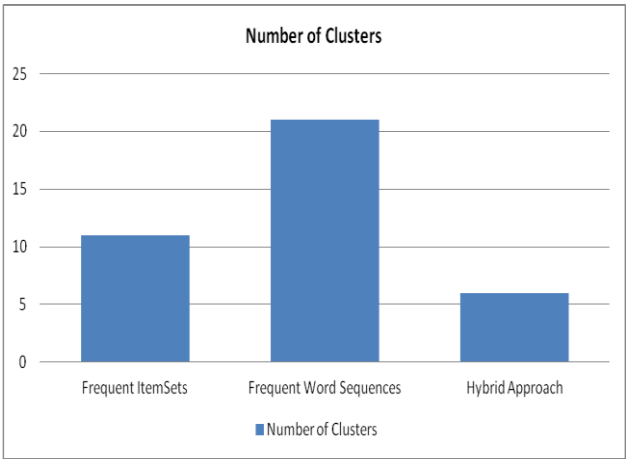

Figure 6. Number of Clusters

This figure shows that maximum number of clusters are generated by Frequent Word Sequences as compared to the other two algorithms. These clusters are ambiguous in title and there is a great deal of overlapping in the topic of the clusters that are being generated. The performance of Frequent Itemsets algorithm is better than the previous as it makes use of techniques such as tree pruning, child pruning and sibling merging, for reducing numbers of meaningful cluster. The proposed Hybrid approach gives the best results in terms of less number of clusters by utilizing the meaningful cluster labels which is achieved through utilization of manually produced list of cluster titles for sports domain.

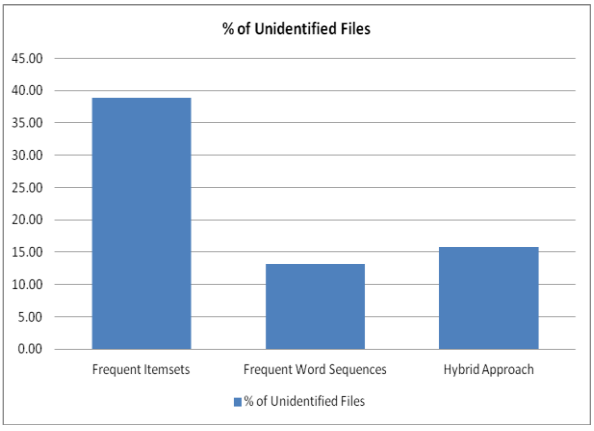

Figure 7. Percentage of Unrecognized Files

Input documents, which do not provide vital information in key frequent phrases or it does not contain that keyword at all, and are to be placed in proper clusters; the algorithms are incapable of clustering these documents. A frequent Item set is not able to cluster the maximum number of files and leaves them as Unrecognized. Frequent Word Sequence shows the best performance in this case. The performance of the Hybrid algorithm performs is very close to Frequent Word Sequence as it generates an optimum output. The reason why the Hybrid Algorithm leaves documents as unrecognized has been duly dealt with in the section on Error Analysis.

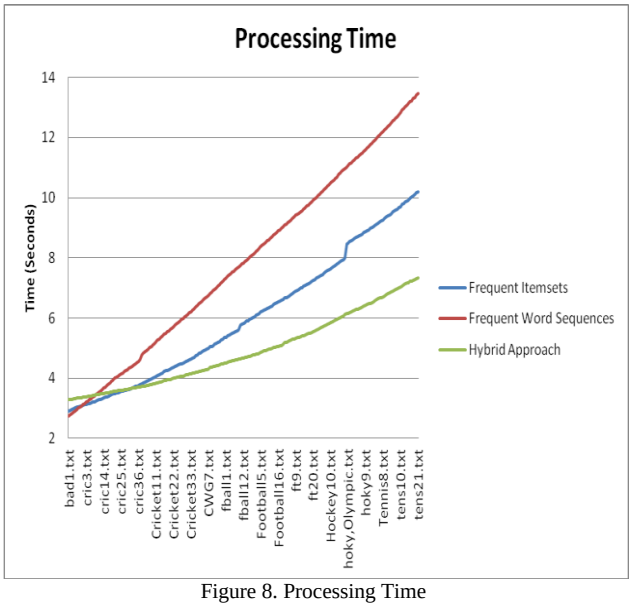

Figure 8. Processing Time

The efficiency of the algorithm is determined on the basis of the time taken for generating the clustering results. Fig. 8 shows the processing time of each algorithm. Hybrid algorithm takes the least time for processing as compared to the other two algorithms. The reason for this can be understood by the fact that both algorithms find longest sequence of k-word length by trying all possible combination of (k -1) word sequences in n number of steps. As for Hybrid algorithm, longest sequence can be found in just one step by extracting the complete phrase from input sentence with the help of karaka list.

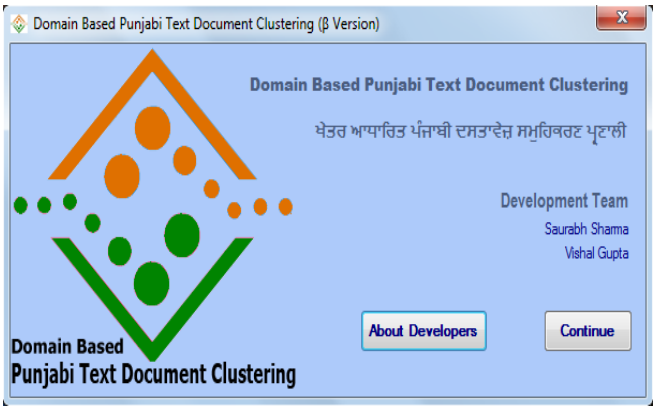

Figure 9. Welcome Screen

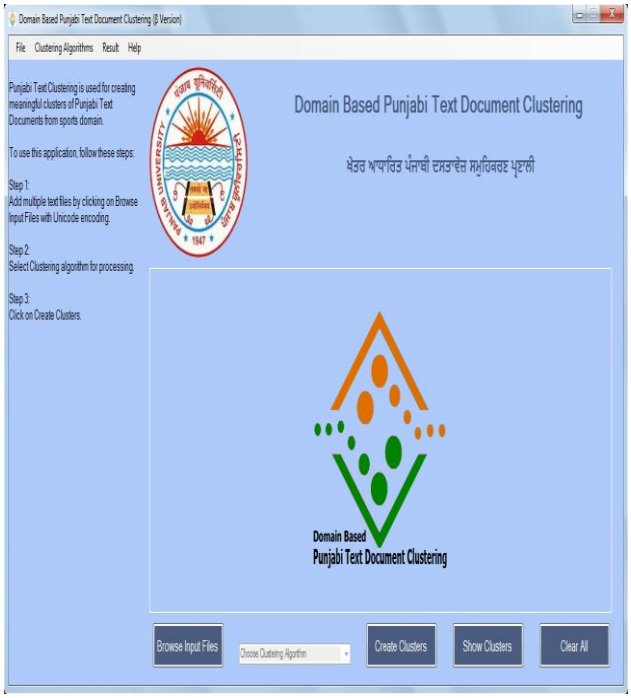

Figure 10. Screenshot of Main Window of Clustering Application

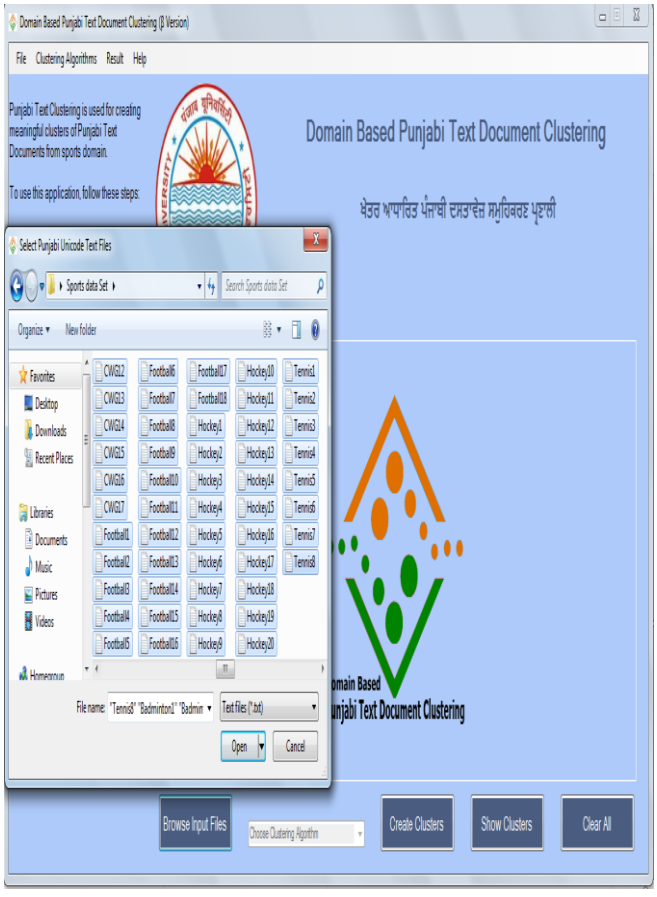

Figure 11. Screenshot of Selecting Multiple Files for Clustering

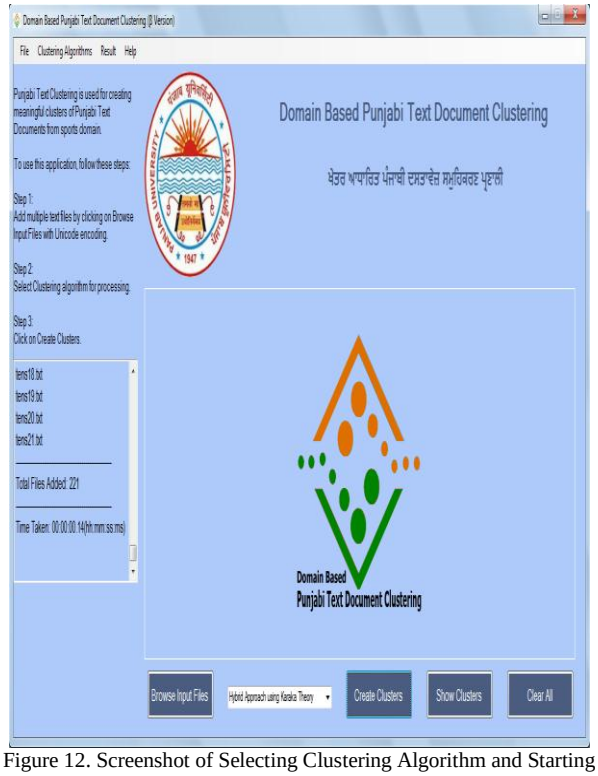

Figure 12. Screenshot of Selecting Clustering Algorithm and Starting Processing

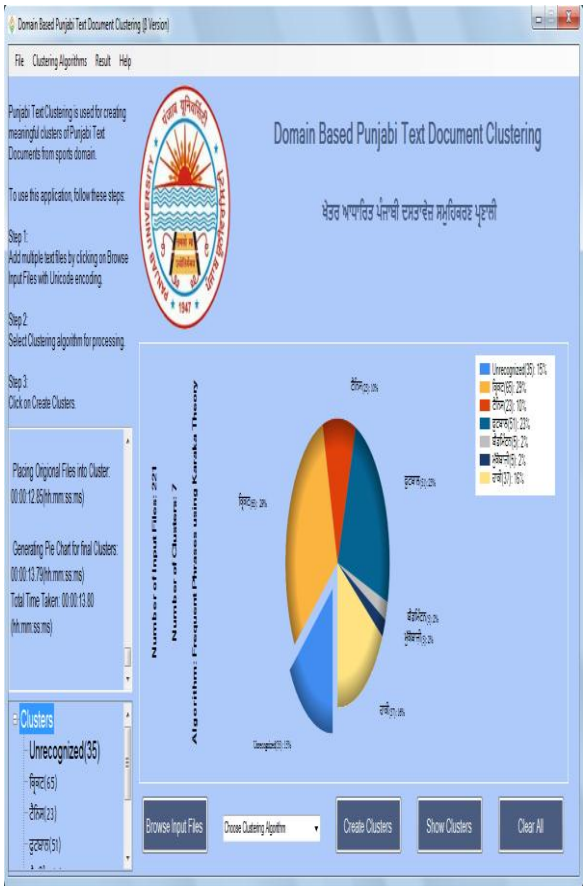

Figure 13. Screenshot of Clustering Results

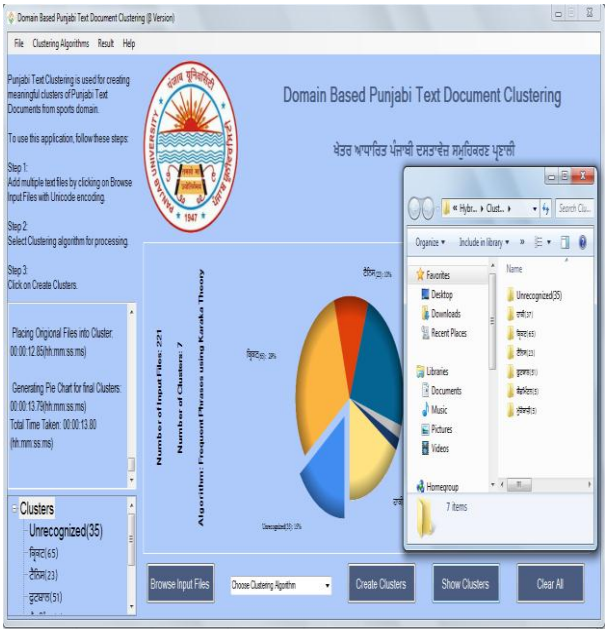

Figure 14. Screenshot of Clusters Folder

VIII. Error Analysis

During the development of this algorithm, several problems for improving clustering results were encountered. These problems and reason for errors in clustering result are discussed below.

Different Spellings in Different Documents results in True Negative. In case of words, which are originally from other languages than the one under purview, e.g. English word 'football' can be written as ਫੁਟਬਾਲ or ਫੁੱਟਬਾਲ. Now, during clustering phase, efforts are made to find similarity between two documents about football, but having different spellings, that do not match. To overcome this problem, we have used normalization of Cluster Titles in pre-processing step. Phrases containing Important Terms but not coming in Top k Frequent Phrases, results in True Negatives. For example, a document contains news about football. But word 'football' is appearing only one or two times in whole document, then it is very hard to capture this desired information in top k Frequent phrases. To overcome this problem, VSM approach is utilized after creating Initial clusters. In this step, top k Frequent Terms are identified. Advantage of applying this step is utilizing those meaningful terms which are not captured in top k Frequent phrases, but very vital for efficient, effective & correct clustering of documents.

Multiple Key Phrases matches with Multiple Cluster Titles results in False Positive and True Negative. For example, a document contains an article on football, but uses some terms common with other sports e.g. team, goal, match referee etc. then it becomes difficult to identify the exact cluster for the document. To overcome this problem, the number of matching Cluster frequent Terms are counted for each matching cluster. Document is, then, placed in that cluster which has maximum number of matching Cluster frequent Terms.

IX. Conclusion

Proposed algorithm is logically feasible, efficient and practical for Punjabi text documents. Experimental data reveals that hybrid algorithm has more feasibility and performs better than Frequent Itemsets and Frequent Word Sequences with Punjabi text data sets. The results are validated and drawn from the experimental data. The reason that can be attributed for their behaviour is that these algorithms, focus on frequent sets without considering the semantics of a sentence. The other contributing reasons for poor results are, the meaningless names of clusters and the creation of a huge number clusters. Proposed algorithm shows better results as it uses a list of Cluster Title candidates, which does not allow the construction of huge number of clusters with meaningless names.

X. Future Work

To increase the efficiency and efficacy, domain specific ontology as an external source of information in addition or in place of Cluster Titles list can be used. Those domains, where documents must contain any word which is present in Cluster Titles List, the usefulness of Cluster Titles list is noticed. But in cases, where document does not contain any word from Cluster Titles List, but still belong to same cluster and must be identified correctly, it does not serve the required purpose. For identifying those documents, use of ontology results in more correct and better clustering results. Some ontology based approaches were proposed in recent researches which have made use of Wikipedia as an external source of information. [34] [35] For further expansion of the theory and to conduct future research in the same genre, this proposed algorithm can be combined with the above approaches for more accurate and efficient clustering results.

References

[1]. T. Weixin, and Z. Fuxi, “Text Document Clustering Based on the Modifying Relations”, In *Proceedings of International Conference on Computer Science and Software Engineering*, vol. 1, pp. 256-259, 2008.

[2]. F. Beil, M. Ester, and X. XU, “Frequent term-based text clustering”, In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, Edmonton, Alberta, Canada, pp. 436 – 442, 2002.

[3]. Z. Volkovich, V. Kirzhner, A. Bolshoy, E. Nevo and A. Korol, "The method of N-grams in large-scale clustering of DNA texts”, *Pattern Recognition*, 38, 11, pp. 1902-1912, 2005.

[4]. D. Liu, Y. He, D. Ji, and H. Yang, “Multi-Document Summarization Based On BE-Vector Clustering”, CICL, *Lecture Notes in Computer Science*, 3878, pp. 470-479, 2006.

[5]. Y. Li, M. Soon, S. M. Chung, and J. D. Holt, “Text document clustering based on frequent word meaning sequences”, *Data & Knowledge Engineering*, 64, 1, pp. 381-404, 2008.

[6]. D. H. Widyantoro, and J. Yen, “A fuzzy similarity approach in text classification task”, In *Proceedings of The Ninth IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 653 – 658, 2000.

[7]. S. Miyamoto, "Fuzzy multi sets and fuzzy clustering of documents", In *Proceedings of the 10th IEEE International Conference on Fuzzy Systems*, vol. 3, pp. 1539 – 1542, 2001.
[8]. R. Saraçoglu, K. Tutuncu, and N. Allahverdi, "A fuzzy clustering approach for finding similar documents using a novel similarity measure", *Expert Systems with Applications*, 33, 3, pp. 600-605, 2007.
[9]. A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review", *ACM Computing Surveys*, ACM New York, NY, USA, 31, 3, pp. 264-323, 1999.
[10]. M. Steinbach, G. Karypis, and V. Kumar, ,"A comparison of document clustering techniques", In *Proceedings of Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Boston, MA, USA, 2000.
[11]. G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing", *Communications of the ACM*, ACM New York, NY, USA. 18, 11, pp. 613 – 620, 1975.
[12]. T. Kohonen, Self-organizing Maps, *Volume 30 of Series in Information Sciences*, Springer, Berlin, Heidelberg, 1995.
[13]. Y. Yang, and J. O. Pedersen, "A Comparative Study on Feature Selection in Text Categorization", In *Proceedings of the 14th International Conference on Machine Learning*, 1997.
[14]. S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis", *Journal of the American Society for Information Science* (1986-1998); 41, 6, pp. 391-407, 1990.
[15]. S. M. Krishna and S. D. Bhavani, "An Efficient Approach for Text Clustering Based on Frequent Itemsets", *European Journal of Scientific Research*, Euro Journals Publishing, Inc. 42, 3, pp. 399-410, 2010.
[16]. Z. Chong, L. Yansheng, Z. Lei, and H. Rong, "Frequent item set based text clustering with window constraint", *Wuhan University Journal of Natural Sciences*, 11, 5, pp. 1345-1351, 2006.
[17]. X. Liu and P. He, "A Study on Text Clustering Algorithms Based on Frequent Term Sets", *Lecture Notes in Computer Science*, 3584, pp. 347-354, 2005.
[18]. L. Wang, L. Tian, Y. Jia, and W. Han, "A Hybrid Algorithm for Web Document Clustering Based on Frequent Term Sets and k-Means", *Lecture Notes in Computer Science*, Springer Berlin, 4537, pp. 198-203, 2010.
[19]. Z. Su, W. Song, M. Lin, and J. Li, "Web Text Clustering for Personalized E-learning Based on Maximal Frequent Item sets", In *Proceedings of the 2008 International Conference on Computer Science and Software Engineering*, 06, pp. 452-455, 2008.
[20]. Y. Wang, Y. Jia, and S. Yang, "Short Documents Clustering in Very Large Text Databases", *Lecture Notes in Computer Science*, Springer Berlin, 4256, pp. 83-93, 2006.
[21]. W. L. Liu and X. S. Zheng, "Documents Clustering based on Frequent Term Sets", *Intelligent Systems and Control*, 2005.
[22]. H. A. Sánchez, A. P. Porrata, and R. B. Llavori, "A document clustering algorithm for discovering and describing topics", *Pattern Recognition Letters*, 31, 6, pp. 502-510, 2010.
[23]. R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules", In *Proceedings of the 20th International Conference on Very Large Data Bases*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA. , pp. 487 – 499, 1994.
[24]. L. Dolamic, and J. Savoy, "Comparative Study of Indexing and Search Strategies for the Hindi, Marathi, and Bengali Languages", *ACM Transactions on Asian Language Information Processing*, ACM New York, NY, USA. 9, 3, 11, 2010.
[25]. C. P. Masica, The Indo-Aryan Languages, Cambridge University Press, Cambridge, UK, 1991.
[26]. B. C. M. Fung, K. Wang, and M. Ester, „Hierarchical Document Clustering Using Frequent Item sets", In *Proceedings of SIAM International Conference on Data Mining, 2003*.
[27]. P. Weiner, "Linear pattern matching algorithms", *In Proceedings of the 14th Annual Symposium on Foundation of Computer Science*, pp. 1–11, 1973.
[28]. A. Bharati and R. Sangal, "Parsing free word order languages in the Paninian framework", In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics Stroudsburg, PA, USA, pp. 105-111, 1993.
[29]. S. K. Mishra, "Sanskrit Karaka Analyzer for Machine Translation", *M. Phil dissertation*, Jawaharlal Nehru University, New Delhi, 2007.
[30]. A. Bharati and R. Sangal, "A karaka based approach to parsing of Indian languages", In *Proceedings of the 13th conference on Computational linguistics*, Association for Computational Linguistics Stroudsburg, PA, USA, vol. 3, pp. 25-29, 1990.
[31]. P. Kiparsky, "Some Theoretical Problems in Panini's Grammar", *Bhandarkar Oriental Research Institute*, Poona, India, 1982.
[32]. G. Cardona, "Panini: A Survey of Research", Mouton, Hague-Paris.on, 38, 11, pp. 1902-1912, 1976.
[33]. G. Cardona, "Panini: His Work and Its Tradition", *Background and Introduction*, vol. 1, Motilal Banarsidas, Delhi, 1988.
[34]. D. Carmel, H. Roitman, and N. Zwerdling, "Enhancing cluster labeling using Wikipedia", In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, ACM New York, NY, USA. Pp. 139-146, 2009.
[35]. S. Banerjee, K. Ramanathan, and A. Gupta, "Clustering short texts using Wikipedia", In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM New York, NY, USA. Pp. 787 – 788, 2007.
[36]. B. Choudhary, and P. Bhattacharyya, "Text clustering using semantics", In *Proceedings of the 11th International World Wide Web Conference, 2002*.

AUTHORS' INFORMATION

Saurabh Sharma is M.E. in Computer Science & Engineering from University Institute of Engineering & Technology, Panjab University Chandigarh. He has done B.Tech. in Computer Science & Engineering from Swami Devi Dayal Institute of Engineering & Technology, Barwala in 2007. He is devoting his research work in the field of Computational Linguistics and specifically to Text Mining. His research work has been published in reputed International journals.

Vishal Gupta is Assistant Professor in Computer Science & Engineering department at University Institute of Engineering & Technology, Panjab University Chandigarh. He has done MTech. in computer science & engineering from Punjabi University Patiala in 2005. He is among University toppers. He secured 82% Marks in MTech. Vishal did his BTech. in CSE from Govt. Engineering College Ferozepur in 2003. He is also pursuing his PhD in Computer Science & Engineering. Vishal has written around thirty five research papers in international and national journals and conferences. He has developed a number of research projects in field of NLP including synonyms detection, automatic question answering and text summarization etc. One of his research papers on Punjabi language text processing was awarded as best research paper by Dr. V. Raja Raman at an International Conference at Panipat. He is also a merit holder in 10^{th} and 12^{th} classes of Punjab School education board.