Hybrid Approach for Punjabi Text Clustering

Saurabh Sharma
M.E. Research Scholar,
Computer Science & Engineering,
University Institute of Engineering & Technology,
Panjab University, Chandigarh, India

Vishal Gupta
Assistant Professor,
Computer Science & Engineering,
University Institute of Engineering & Technology,
Panjab University Chandigarh, India

ABSTRACT
Text Clustering is a text mining technique which is used to group similar documents into single cluster by using some sort of similarity measure and placing dissimilar documents into different clusters. Most of the popular clustering algorithms treats document as conglomeration of words and do not consider the syntactic or semantic relations between words. To overcome this drawback, some algorithms were proposed which aimed at trying to find connections among different words in a sentence by using different concepts, e.g. Frequent Itemsets, Frequent Words Sequences, Frequent Word Meaning Sequences, Ontology based clustering. In this paper, we proposed a hybrid algorithm for clustering of Punjabi text document, which uses semantic relations among words in a sentence for extracting phrases. Phrases extracted create a feature vector of the document which is used for finding similarity among all documents. Results on experiment data reveal that hybrid algorithm is more reasonable and has a better performance with real time data sets.

General Terms
Natural Language Processing, Text Mining, Text Document Clustering, Punjabi Language.

Keywords
Punjabi Text Clustering, Vector Space Model, Frequent Itemsets, Frequent Word Sequences, Karaka Theory, Ontology.

1. INTRODUCTION
To be able to access[1] the wide plethora of documents that are available in electronic form in Punjabi (Gurumukhi script) a growing need is constantly being felt for a multi-lingual text retrieval system. With the growing volume of text in various vernacular languages including Punjabi, which is one of the most spoken languages in India, language processing as an active area of research has gained impetus. Initially maximum research was done in the area of English. However, In recent times Research on Asian language processing has undergone a complete metamorphosis. The major hurdle on research work in this direction is the limited availability of tools and other lexical resources for these languages (i.e. other than English and major European languages). It holds more relevance for languages from Indian sub-continent. The current work focuses on development of one such tool used for clustering of Punjabi text documents.

A new method for generating feature vectors, using the semantic relations between the words in a sentence is discussed here. The semantic relations are captured by the Karaka Theory[2], which is a recently proposed semantic representation for sentences. The clustering method applied to the feature vectors is the hybrid of Vector Space Model and Frequent Sequence Concepts[3][4].

There are many algorithms for automatic clustering like the K Means algorithm [5], hierarchical clustering [6] which can be applied to a set of vectors to form the clusters. Traditionally the document is represented by the frequency of the words that make up the document (the Vector space model and the Self-organizing semantic map [7]). Different words are then given importance according to different criteria like Inverse Document frequency and Information Gain. These methods consider the document as a bag of words, and do not exploit the relations that may exist between the words[8].

However, this can cause problems[8]. For example, if we consider the two sentences John is manager of the bank and John is standing on the bank of a river. Bag of words representation treat both sentences as similar vector with two common terms i.e. {John, bank}. On the other hand, there may be some sentences, which have the same meaning but have been constructed from different sets of words. For example in the sentences, John is an intelligent boy and John is a brilliant lad, mean more or less the same thing. To solve these types of problems, few algorithms were proposed which try to find correlation among different terms in a sentence, which will be duly discussed in next section.

2. RELATED WORK
Clustering algorithms that are widely being used can be broadly divided into two categories; Hierarchical clustering algorithms and Partitional clustering algorithms. All popular clustering algorithms uses Vector Space Model to represent document vector [9]. VSM treat document as “bag of words”. Occurrence of each word in the document is counted. Each word form a dimension in document vector. Thus, documents can be compared by use of simple vector operations and even queries can be performed by encoding the query terms similar to the documents in a query vector. The query vector can then be compared to each document and a result list can be obtained by ordering the documents according to the computed similarity. The main drawback of this approach is that all the information contained in a sentence about semantics is lost.

To overcome the drawback of VSM model a new concept of Frequent Item Sets [3] was proposed which uses Apriori algorithm[10] for finding Frequent Item Sets from all document terms and only top k Frequent Item Sets are used for creating document vector. The reason, for not considering every term of the document as a dimension of Document vector, is that not every term is important from the point of view of clustering and only single word frequent terms are not sufficient for finding similarity among documents. Frequent Item Sets represents a document much better than VSM model. The main drawback of this approach was that it uses

concept of Frequent Item Sets which was originally proposed for transactional databases. Frequent Item Sets generated by this method do not consider the semantics between terms and create Frequent Item Set based on their frequent co-occurrences.

For removing drawback of Frequent Item Sets by using Frequent Word Sequences and Frequent Word Meaning Sequences another approach was proposed [4]. In this approach, document vector consists of Frequent Word Sequences rather than Frequent Item Sets. Frequent Word Sequence is defined as set of Frequent Terms, which are occurring in a sequence but not necessarily following each other immediately, in a text document. There could be words between them as long as the words between them are not frequent. This approach performs better for English language but do not perform well for Punjabi language. The reason behind this is that the sentence structure of Punjabi is different from English, so Frequent Word Sequences created with this approach, does not represent the document correctly.

Positional languages, which comes in category of Context Free Grammars (CFGs) used all these approaches discussed above. The structure of a sentence in English is different than Punjabi because Punjabi comes in the category of Free Order languages. For clustering of Punjabi text, features of free order languages were to be taken into consideration.

A majority [11] of human languages including Indian and other languages have relatively free word order. In free word order languages, order of words contains only secondary information such as emphasis etc. Primary information relating to 'gross' meaning (e.g., one that includes semantic relationships) is contained elsewhere. Most existing computational grammars are based on context free grammars which are basically positional grammars. It is important to use a suitable computational grammar formalism for free word order languages for 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 a 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.

We have used such a formalism, called the Paninian framework, that has been successfully applied to Indian languages. It uses the notion of karaka relations between verbs and nouns in a sentence. The notion of karaka relations is central to the Paninian model. The karaka relations are syntactico-semantic (or semantico-syntactic) relations between the verbals and other related constituents in a sentence. They by themselves do not give the semantics. Instead they specify relations which mediate between vibhakti of nominals and verb forms on one hand and semantic relations on the other [12][13][14]. See Fig. 1. 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.

As part of this framework, a mapping is specified between *karaka* relations and vibhakti (which covers collectively case endings, post-positional markers, etc.). This mapping between *karakas* and vibhakti depends on the verb and its tense aspect modality (TAM) label. The mapping is represented by two structures: default *karaka* charts and *karaka* chart transformations. The default *karaka* chart for a verb or a class of verbs gives the mapping for the TAM label tA_hE called basic. It specifies the vibhakti permitted for the applicable *karaka* relations for a verb when the verb has the basic TAM label. This basic TAM label roughly corresponds to present indefinite tense and is purely syntactic in nature[11] see Table 1.

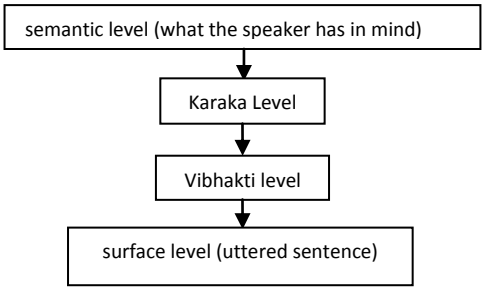

Fig. 1: Levels in the Paninian mode

Table 1. Default Karaka Chart

KARAKA	VIBHAKTI	PRESENCE
Karta	Φ	mandatory
Karma	ko or Φ	mandatory
Karana	se or dvArA	optional

3. PROPOSED APPROACH

3.1 Preprocessing

Preprocessing is defined as number of steps applied on the input text for converting it from a 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).

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 is being used for generating phrases[2]. *Karaka* theory works only on complete sentences that necessarily includes 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.

3.2 Hybrid Algorithm

3.2.1 Definitions

Karaka symbols can be defined as any of various words in languages such as Hindi, Punjabi, Japanese, Hungarian, 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. **Karaka List** is the collection of *Karaka* symbols which are used to identify the phrases from a sentence.

3.2.2 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

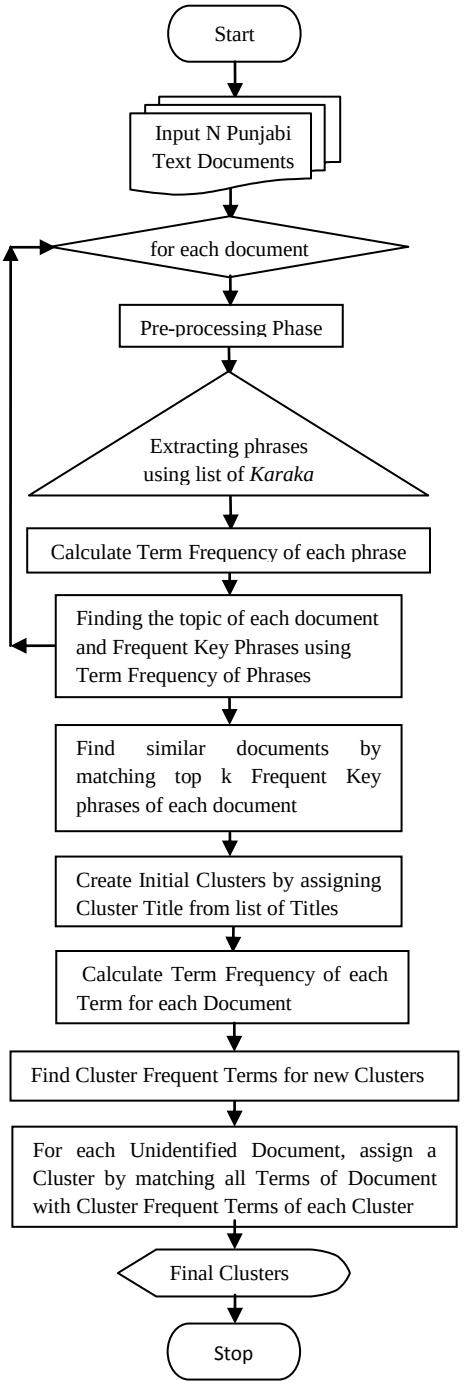

Fig 1: Flowchart of Hybrid Algorithm

list is to overcome the drawback of Frequent Item Sets[3] and Frequent Word Sequences[4], generating long Sequences by trying all combinations of 2-word sequences, using Apriori algorithm[10]. For example, we have a 4-word sequence in input file, "Panjab University Football Club". In case of Frequent sequence algorithms, after pre-processing step, using Apriori algorithm [10], initially we have 2-word sequences, {Panjab University},{University Football} and {Football, Club}. Then, we try to find longer sequence of 3-word length by combination of 2-word sequences {Panjab University Football}, {University Football Club}.

ਪੰਜਾਬ ਯੂਨੀਵਰਸਿਟੀ ਫੁਟਬਾਲ ਕਲੱਬ ਨੇ ਫੁਟਬਾਲ ਕਪ ਜੀਤਾ|

pañjāb yūnīvrasiṭī phūṭbāl kalabb (nē) phuṭbāl kap jītā

(Panjab University Football Club won the football cup.)

In our hybrid algorithm, by using *Karaka* List (see Table 2), we 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 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 }

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

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

Table 2. Karaka List

KARAKA	SYMBOL
ਕਰਤਾ (karatā)	ਨੇ (nē)
ਕਰਮ (karam)	ਨੂੰ (nūṃ)
ਕਰਣ (karaṇ)	ਨਾਲ (nāl)
ਸੰਪਰਦਾਨ (sampradān)	ਲਈ (laī)
ਅਪਾਦਾਨ (apādān)	ਤੋਂ (tōṃ)
ਸੰਬੰਧਕ (sambandhak)	ਦਾ/ਦੇ (dā/dē)
ਅਧਿਕਰਣ (adhikraṇ)	ਪਾਸ, ਕੋਲ (pās, kōl)

Extraction of phrases from the document with the help of *Karaka* list generates a document vector containing phrases of various length 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.

3.2.3 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.

3.2.4 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.

3.2.5 Finding Similar Documents and Creating Initial Clusters

In this step, we will create initial clusters 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.

Cluster Title List is list of words which are candidate terms for a cluster title. 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 numbers 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. For conducting experiments, sports domain has been taken and list of Cluster Titles specific to sports have been created.

Files with same Cluster Title are placed into same cluster. If two files contain same 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.

3.2.6 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, will be placed in a cluster named Unidentified. Since, some files may contain cluster titles but did not appear in top k Frequent Phrases, for those unidentified files, VSM model is used i.e. now document is represented as a collection of terms, obtained from all phrases. For each unidentified document, Term Frequency for each term in 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 unidentified document. Higher value is beneficial for those documents in which term frequency of cluster title is very low. Higher value of k showed better results as compare to low value of k.

3.2.7 Find Cluster Frequent Terms for new Clusters

After calculating top k frequent terms for each unidentified 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.

3.2.8 For each Unidentified Document, assign a Cluster by matching all Terms of Document with Cluster Frequent Terms of each Cluster

Cluster for unidentified document, by matching top k Frequent Terms of documents with top k cluster Frequent Terms of each document is identified. If a match found with a cluster, then document is moved from unidentified cluster to that identified cluster. If a match is found with more than one cluster, then we the number of matched terms for each cluster is counted. Document is placed in that cluster, which has maximum number of matched terms.

3.2.9 Final Clusters

After processing of Unidentified documents, final clusters containing documents from initial cluster and documents from Unidentified documents are created

4. EXPERIMENTAL EVALUATION

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

4.1 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 large dimensional data elements. The system is tested with 500 text documents collected from various Punjabi News websites relating to sports news articles which were used in the evaluation of the proposed algorithm. It is important to measure the efficiency of the proposed method. The proposed method of the research adopted the most commonly used measures in the data mining, namely, precision and recall for the general assessment.

4.2 Experimental Results

A commonly used external measurement, the F-measure is employed to evaluate the accuracy of the clustering result thus generated by the proposed algorithm. It is a standard evaluation method for both flat and hierarchical clustering structures. This is further illustrated in table 3:

Table 3. Accuracy of Proposed Algorithm

Precision	Recall	F-Measure
0.90	0.82	0.86

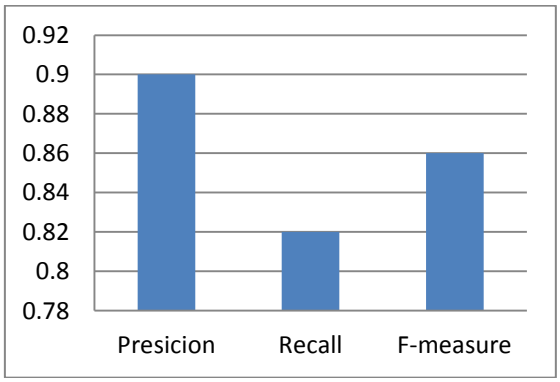

Fig 3: Experimental Results

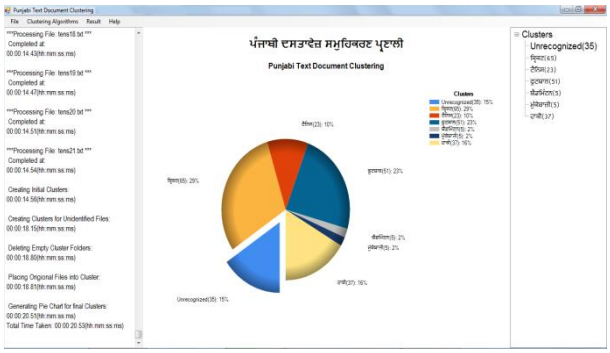

Fig 4: Screenshot of Clustering Application

5. LESSON LEARNED

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

Different Spelling 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 spelling, they do not match. To overcome this problem, we have used normalization of Cluster Titles in preprocessing 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 extremely 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 cricket, but uses some terms from other sports, 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.

6. CONCLUSION

The experimental results showed that proposed algorithm is logically feasible, efficient and practical for analysis of semantics of Punjabi text documents. Experimental data also reveals that hybrid algorithm is more feasible and has a better performance even with real time data sets.

7. REFERENCES

[1] Pandey, A.K. and Siddiqui, T.J.2008.An unsupervised Hindi stemmer with heuristic improvements. In Proceedings of the second workshop on Analytics for noisy unstructured text data, pp. 99-105, ACM New York, NY, USA. ISBN: 978-1-60558-196-5 doi>10.1145/1390749.1390765.

[2] Bharati, A., Sangal, R. 1990. A karaka based approach to parsing of Indian languages. In Proceedings of the 13th conference on Computational linguistics - Volume 3, pp. 25-29. Association for Computational Linguistics Stroudsburg, PA, USA. ISBN:952-90-2028-7 doi>10.3115/991146.991151

[3] Benjamin C.M. Fung, Ke Wang, Martin Ester. 2003. Hierarchical Document Clustering Using Frequent Itemsets. IN Proceedings of SIAM International Conference on Data Mining.

[4] Yanjun Li, Soon M. Chung, John D. Holt. 2008. Text document clustering based on frequent word meaning sequences. Data & Knowledge Engineering, Volume 64 Issue 1, (Jan. 2008), 381-404. Elsevier Science Publishers B. V. Amsterdam, The Netherlands, The Netherlands. doi>10.1016/j.datak.2007.08.001

[5] Hartigan, J. A. and Wong, M. A. 1979. Algorithm AS 136: A K-Means Clustering Algorithm. Journal of the Royal Statistical Society, Series C (Applied Statistics) Volume 28, No. 1, (1979), 100-108. Blackwell Publishing.

[6] Anil K. Jain, Richard C. Dubes. 1988. Algorithms for clustering data. Prentice-Hall, Inc. Upper Saddle River, NJ, USA. ISBN:0-13-022278-X

[7] T. Kohonen. 1995. Self-organizing Maps, Series in Information Sciences, vol. 30, Springer.

[8] Choudhary, B. and Bhattacharyya, P. 2002. Text clustering using semantics. In Proceedings of the 11th International World Wide Web Conference.

[9] Salton, G., Wong, A. and Yang, C. S. 1975. A vector space model for automatic indexing. Communications of the ACM, Volume 18 Issue 11, Nov. 1975, pp. 613 - 620. ACM New York, NY, USA. doi>10.1145/361219.361220

[10] Agrawal R. and Srikant, R. 1994. Fast Algorithms for Mining Association Rules. In Proceedings of the 20th International Conference on Very Large Data Bases. pp. 487 - 499. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA. ISBN:1-55860-153-8.

[11] Bharati, A. and Sangal, R. 1993. Parsing free word order languages in the Paninian framework. In Proceedings of the 31st annual meeting on Association for Computational Linguistics, pp. 105-111. Association for Computational Linguistics Stroudsburg, PA, USA. doi>10.3115/981574.981589

[12] Kiparsky, P. 1982. Some Theoretical Problems in Panini's Grammar, Bhandarkar Oriental Research Institute, Poona, India.

[13] Cardona, G. 1976. Panini: A Survey of Research, Mouton, Hague-Paris.

[14] Cardona, G. 1988. Panini: His Work and Its Tradition (Vol. 1: Background and Introduction), Motilal Banarsidas, Delhi.