MAP REDUCE BASED BAG OF PHRASES REPRESENTATION AND DISTRIBUTIONAL FEATURES INCORPORATION FOR TEXT CLASSIFICATION

M. Janaki Meena
SCSE, VIT Chennai, India
E-Mail: janakimeena.m@vit.ac.in

ABSTRACT

Text classification is the basis step for developing intelligent information systems such as language identification, biography generation, authorship verification, content filtering, search personalization, product classification, sentiment analysis, detection of malicious activities, patent classification and opinion mining. From early 90's various machine learning approaches have been applied to text classification. Document representation is the process of converting raw documents into a set of features that shall be fed into machine learning algorithms. Features for applying machine learning algorithms to text corpus shall be words, n-grams (phrases) or synsets. Distribution of features in a document is also important for deciding their importance. In this research, a MapReduce based bag of phrases representation is used for classifying text using Naïve Bayes Classifier. The proposed feature selection algorithm is converted to MapReduce programming model and the results are discussed. Precision and recall are metrics that are used in this research to compare the results. It has been observed that bag of phrases representation gives better accuracy for technical documents and including distributional features improves the accuracy of the classifier.

Keywords: MapReduce, bag-of-phrases, CHIR, distributional features, feature selection, naïve bayes classifier.

INTRODUCTION

The research made by Deerwester et al., has found that more than 80% of the time, two different people choose unlike key words to describe a familiar object [1]. This is because of the properties of the English language like polysemy and synonymy. These properties of the language have made document categorization to be a challenging task. It has also been observed that synonym of the language has led to different writing style among the people; that is people are using different words to convey the same meaning. This property of the language decreases the performance of algorithms that chooses features based on the frequency of terms. To reduce the performance hitches of feature selection algorithms due to synonymy and polysemy, in this research documents have been represented as a bag of phrases. Syntactic phrases are groups of words with certain syntactical relationships that occur in any order in a document. Since phrases reduce ambiguity and have narrower meanings, they are much desirable text attributes. The dimension of the text is still high in these representations too and needs feature selection to remove the irrelevant and redundant terms from the corpus [1].

In Bag-of-words (BOW) representation, each document is represented as a vector in which each element matches to the frequency of the word in the document [2].When texts written by people from different demographical background were analyzed, it was observed that there is no uniformity in the usage of words. It is also observed that the semantic relations among the words get destroyed in the BOW representation. Even stable phrases, such as "Information Retrieval", are separated as individual words and their meaning is lost when represented as BOW. Representing documents using phrases helps in resolving the problem to an extent. Two types of phrases have been explored in literature. They are syntactic and statistical phrases. This research tries to identify more dominant syntactic phrases of length up to two for document representation. Syntactic phrases were identified with the help of Part of Speech (POS) tagging in Wordnet.

A comparative study on the feature selection techniques proposed for text classification has reported that the features chosen based on information gain and χ^2 value are better than other techniques [3]. In this research, the performance of the Naïve Bayes classifier has been analyzed with the words selected by information gain, chi-square and CHIR methods. CHIR is a variant of chi-square method proposed by Li et al., which considers relevancy of the feature to the category for feature selection [4], [11]. It was also observed that there is an increase in the performance of the classifier when words chosen by CHIR method were used. Further it is also important to include the position and context of the words but concentrated only on their frequency of occurrences. Most of the times, human's judgment consider distributional features of terms to categorize documents.

MapReduce is a programming model proposed by Google for parallelizing the code in a distributed framework. Hadoop is open source framework which has implemented MapReduce programming model. The basic difference in MapReduce is to move the code between nodes in a cluster rather than moving data between the nodes as in the case of other distributed processing technologies [6].

FEATURE SELECTION

Not all features are important for classification, sometimes too many features fed into a classifier may lead to tainted performance of the classifier [8]. Many feature selection algorithms exists and statistical methods are more popular.

The concept of χ^2 statistic (CHI)

CHI algorithm is a well-known technique to select the features for applying machine learning algorithms to text corpus and it is based on feature-category independence test [4]. Observed frequencies are compared with the expected frequencies to calculate the χ^2 Value of a feature with respect to a category, here it is assumed that the occurrences of the terms are independent. In a corpus with ‘n’ labeled documents that fall in ‘m’ categories, for each feature in the corpus a 2x2 contingency table is formed as shown in Table-1.

Table-1. A 2x2 contingency table for ‘t’ with respect to ‘c’.

	c	¬c	∑
W	100	25	125
¬w	60	340	400
∑	160	365	525

Expected frequency is calculated as E (i, j) using Equation (1), here the presence or absence of a feature is represented by ‘i' and whether the document belongs to a category is represented by ‘j’:

$$E(i,j) = \frac{\sum_{a\in\{w,\neg w\}} O(a,j) \sum_{b\in\{c,\neg c\}} O(i,b)}{n} \quad (1)$$

For a feature ‘*w’* in the training corpus, CHI value with respect to a category ‘c’ is given by Equation (2):

$$\chi^2_{w,c} = \sum_{i\in\{w,\neg w\}} \sum_{j\in\{c,\neg c\}} \frac{(O(i,j)-E(i,j))^2}{E(i,j)} \quad (2)$$

“Degrees of freedom” describes the number of values that are free to vary in the final calculation of a statistic. For a contingency table of dimension r, c “degrees of freedom” is (r-1) * (c-1). For every term, equation (2) is used to calculate a value ‘v’. For the determined degrees of freedom of the term and with a confidence level of 0.1%, look at the value in standard χ^2 tabulation be and let it be ‘f’. Let the value in χ^2 tabulation for the degrees of freedom of ‘1’ with 0.1% confidence level is 10.83. Null hypothesis has to be rejected ‘v’ is greater than ‘f’ and alternative hypothesis has to be considered. Hence for example if $\chi^2_{w,c}$ value (v) is 72 then it is greater than the value in the tabulation, 10.83, hence null hypothesis has to be rejected and alternative hypothesis has to be considered.

Whenever there is a dependency, dependency is decided by goodness-of-fit. Discrepancy between the observed and the expected values of model in question is measured by goodness of fit. For a text corpus with m categories, term-goodness is determined as either the average as given by Equation (3) or the maximum as given by Equation (4):

$$\chi^2_{avg}(w) = \sum_{j=1}^{m} p(c_j)\chi^2_{w,c_j} \quad (3)$$

$$\chi^2_{\max}(w) = \max_j\{\chi^2_{w,c_j}\} \quad (4)$$

We have used maximum value for comparison. Terms were ranked based on their χ^2 value and terms having greater positive dependency to some category in the corpus were selected.

Drawback of χ^2 statistics

CHI algorithm can determine only the existence of a dependency of a feature to a category but not the polarity (positive or negative) of the dependency.

The concept of CHIR method

CHIR is a variation of CHI algorithm which the polarity of dependency in addition to its existence. Polarity of dependency is determined by a new relevancy measure $R_{w,c}$ as given in Equation (5):

$$R_{w,c} = \frac{O(w,c)}{E(w,c)} \quad (5)$$

If the value of $R_{w,c}$ is close to 1, then one may conclude that the term ‘w’ has no dependency to the category ‘c’. When there is a positive dependency between the term ‘w’ and the category ‘c’ then the value of $R_{w,c}$ will be larger than 1 and when there is a negative dependency between the term ‘w’ and category ‘c’ then $R_{w,c}$ is smaller than 1.

Based on χ^2 statistics and $R_{w,c}$ a new definition for term-goodness for a corpus with m classes is given in Equation (6):

$$r\chi^2(w) = \sum_{j=1}^{m} p(R_{w,cj})\chi^2_{w,cj} \text{ with } R_{w,c_j} > 1 \quad (6)$$

Here weight of $\chi^2_{w,cj}$ in the text corpus is calculated as $p(R_{w,c_j})$. In terms of R_{w,c_j}, $p(R_{w,c_j})$ is shown as Equation (7):

$$p(R_{w,c_j}) = \frac{R_{w,c_j}}{\sum_{j=1}^{m} R_{w,c_j}} \text{ with } R_{w,c_j} > 1 \quad (7)$$

Term w is more relevant to category 'c' for larger values of $r\chi^2(w)$. For each unique term in the text corpus, calculate $r\chi^2$ value, then sort the terms in descending order based on their $r\chi^2$ value and then the top 'q' terms are selected.

FEATURE SELECTION ALGORITHM TO EXTRACT UNIGRAMS AND BIGRAMS

The proposed algorithm aims at extracting prominent syntactic bigrams to supplement unigrams for classification. Syntactic phrases are semantically rich but are not able to influence the classification process due to its statistical sparseness and the phenomenon that phrases with same meaning contains different linguistic units [7], [10]. They are treated as different phrases when stored in their sequence of occurrence. The proposed algorithm uses a method similar to the one used by Caropreso *et al.* method begins with the usual preprocessing of documents such as stop word removal and stemming. Then, the terms are alphabetically ordered to factor out the morphological, syntactic, and semantic variations of n-grams [5]. Morphosyntactic variations of phrases include different forms of noun phrases, verb phrases and full sentences.

As for semantic variations, noun phrases with different meanings also give rise to the same n-gram.

Here n-grams are defined as various syntactic expressions may convey the same concept, and hence they must be seen as a form of conflation [9]. Generalizations made based on n-grams have problems such as over-generalization and under-generalization. POS tagging is the basic step for identifying syntactic phrases and it is done using syntactic rules of English language and Wordnet.

The algorithm uses the following steps in Figure-1 and extracts noun phrases and verb phrases of length up to two from the corpus.

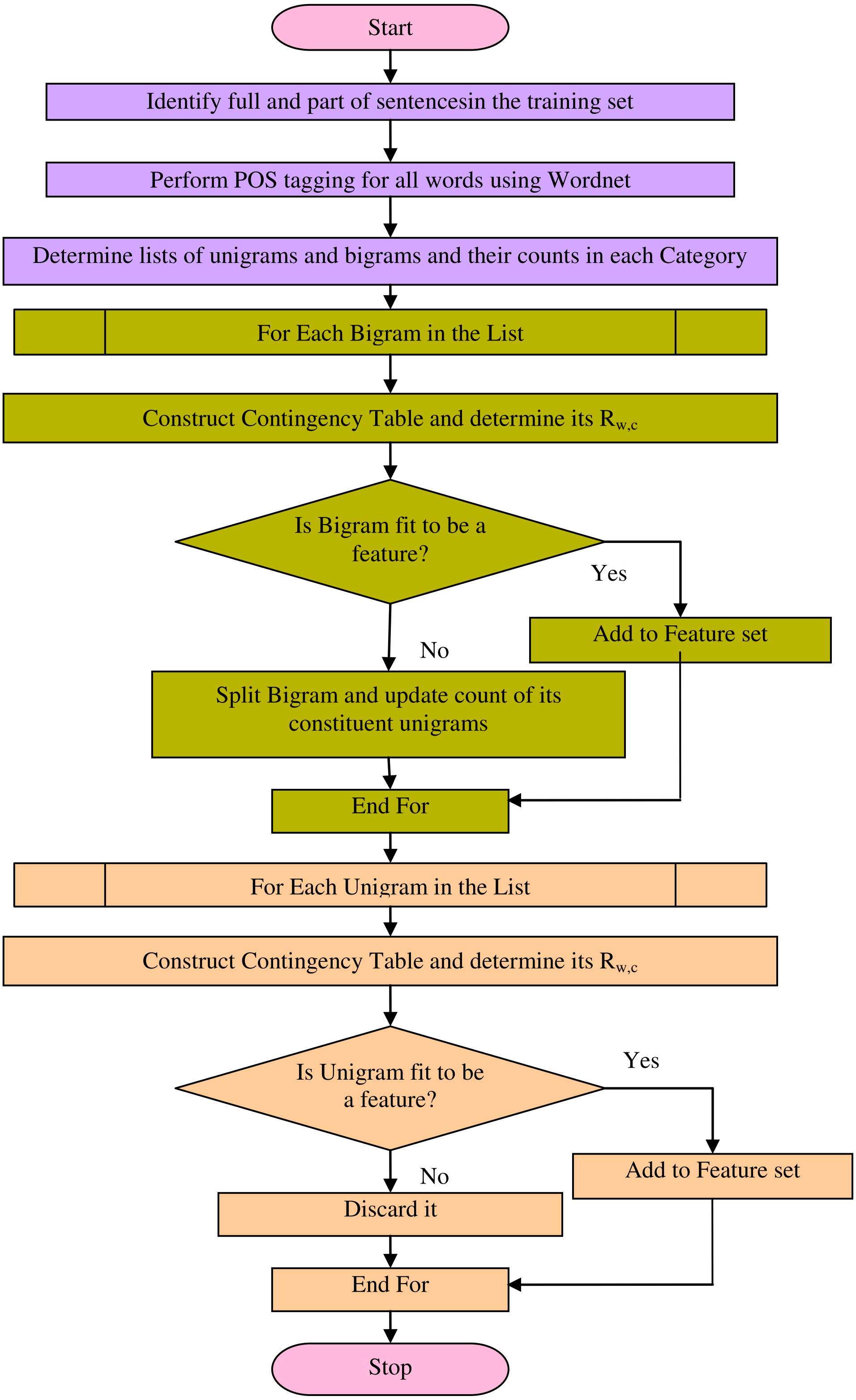
Start
Identify full and part of sentencesin the training set
Perform POS tagging for all words using Wordnet
Determine lists of unigrams and bigrams and their counts in each Category
For Each Bigram in the List
Construct Contingency Table and determine its $R_{w,c}$
Is Bigram fit to be a feature?
Yes
Add to Feature set
No
Split Bigram and update count of its constituent unigrams
End For
For Each Unigram in the List
Construct Contingency Table and determine its $R_{w,c}$
Is Unigram fit to be a feature?
Yes
Add to Feature set
No
Discard it
End For
Stop

Figure-1. Algorithm for extracting unigrams and bigrams.

ALGORITHM TO IDENTIFY SEMANTICALLY RELATED PHRASES

This section describes the algorithm to identify phrases that are semantically related but morophosyntactically varied. The input for the algorithm is either the full sentence or part of the sentence separated by a comma, semicolon, colon, and etc. The algorithm is as follows:

Procedure: Identify_Uni_and_Bigrams(S, Dic)
Input: S: Sentence or part of Sentence
Dic: Dictionary used

Output: List of uni and bigrams in sentence S
LU$_s$ - List of Unigrams in the sentence
LB$_s$ - List of Bigrams in the sentence

```
For each word $W_i$ in S
If $W_i$ is not a stop_Word then
Determine its POS using Dic
If $W_i$ and $W_{i+1}$ is a noun or adjective or verb then
Stem words $W_i$ and $W_{i+1}$
Form phrase p with $W_i$ and $W_{i+1}$ placed alphabetically with an '_' inbetween them.
Store p in $LB_s$.
else
Stem and Store $W_i$ in $LU_s$
End
Return
```

ALGORITHM TO IDENTIFY RELEVANT BIGRAMS

This section illustrates the algorithm to identify the bigrams that are with positive relevancy to the categories. The irrelevant bigrams and bigrams with negative relevancy are splitted to its constituent unigrams and then the counts of the corresponding unigrams are updated.

Procedure: Identify_Fit_Bigrams(LOBO, LOUO)
Input: List of Unigrams and Bigrams
Output: List of Features with Bigrams
LOBO - List of bigrams with count of occurrences in all categories
LOUO - List of unigrams with count of occurrences in all categories

```
For all $LB_i \in$ LOBO do
Form contingency table and determine $R_{w,c}$
If $R_{w,c} > 1$
GOT $\leftarrow$ Estimate_Goodness_Of_Term($LB_i$)
LF $\leftarrow$ LF $\cup$ $LB_i$
Else
$B_{i1}$ , $B_{i2}$ $\leftarrow$ Split($LB_i$)
LOUO $\leftarrow$ Insert or Update_Count($B_{i1}$)
LOUO $\leftarrow$ Insert or Update_Count($B_{i2}$)
Return LF
End
```

DISTRIBUTIONAL FEATURES

Sometimes distributional features of terms give a good guidance factor for feature selection. Hence in this section, the CHIR method is enhanced using two distributional features, first appearance and compactness. First appearance is given importance based on the intuition that terms that appear in the earlier part of a document are more important. Compactness of a term is measured based on the usage of the term throughout the document. More important terms are spread across the entire document and have less compactness value. For a document d, with n sentences a distributional array for each term is formed as $array(t,d) = [c_0, c_1, ..., c_{n-1}]$ and first appearance is defined as Equation (8):

$$FirstApp(t,d) = \min_{i \in \{0,...,n-1\}} c_i > 0\,?\,i : n \tag{8}$$

In this research work, compactness is calculated based on the variance of the positions of all appearances of the term. Mean position of all appearances of the term is computed and then the average distance between the positions of each appearance of the term is calculated. Mean position is determined as position variance using Equation (9).

$$Compact_{PosVar}(t,d) = \frac{\sum_{i=0}^{n-1} c_i \times |i - centroid\,(t,d)|}{count(t,d)} \tag{9}$$

Where centroid and count are calculated using Equation (10) and Equation (11)

$$centroid\,(t,d) = \frac{\sum_{i=0}^{n-1} c_i \times i}{count(t,d)} \tag{10}$$

$$count(t,d) = \sum_{i=0}^{n-1} c_i \tag{11}$$

Algorithm to Integrate DF and CHIR

This section gives the outline of the algorithm to choose some useful negative features based on their relevancy and distributional properties.

Step 1: Preprocess the training corpus and identify the terms in it.
Step 2: Contingency table is formed for each term with respect to each category and only positive features based on their goodness-of-fit value are included
Step 3: For each negative feature, compute distributional features and include those terms with values greater than the threshold.

MAPREDUCE ALGORITHM TO EXTRACT PHRASES

The code to select features for bag-of-phrases representation also has two mappers and reducers. POS tagging is the important step in the algorithm. Wordnet is used for POS tagging of words in the corpus. There are two major steps in the algorithm:

a) Remove stop words and identify the key unigrams and bigrams.

b) Compute the relevancy and CHI value of each term in the text corpus.

Raw training documents are given as input for the first mapper. The input key for the mapper is the line offset and input value is the line of text in the document and the output key are the syntactic phrases in the document and the output value is 1.

The map function of the first module does POS tagging for each word by connecting to Wordnet.

Using the grammatical rules of the language and the POS tagging of the words, valid verb and noun phrases are identified in the text corpus.

The output from the map function is sorted and the key-value pairs are grouped by key. The output of the map function enters in to the shuffle and sort module of the framework and fed as the input for the reduce function.

The reduce function iterate through the list and determine the sum of each phrase. Each reducer writes its output in a file. Figure-2 shows the logical data flow of MapReduce module for phrases extraction. The mapper creates contingency table for each phrase corresponding to each category and stores them in a file.

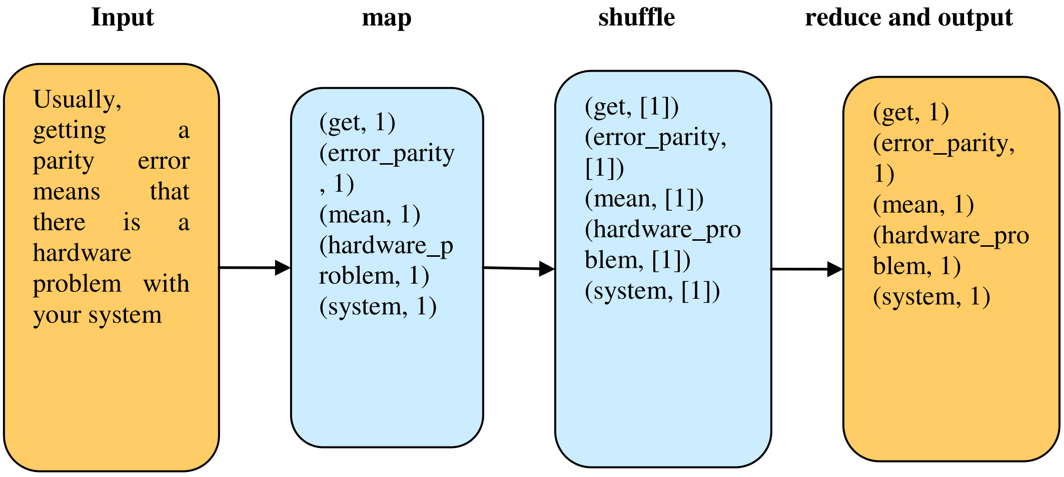

Figure-2. Logical data flow of MapReduce module for phrases extraction.

The input for the second MapReduce module is the output of the first MapReduce module. Mapper is fed with the set of phrases and their corresponding contingency table. The contingency given to the mapper is a string separated by $ symbol, the mapper splits the contingency table string and computes the CHI and $R_{w,c}$ values. The reduce function includes only phrases with positive relevancy and eliminates features with negative relevancy. Figure-3 shows the logical data flow of the second MapReduce module that applies CHIR algorithm for bag-of-phrases representation.

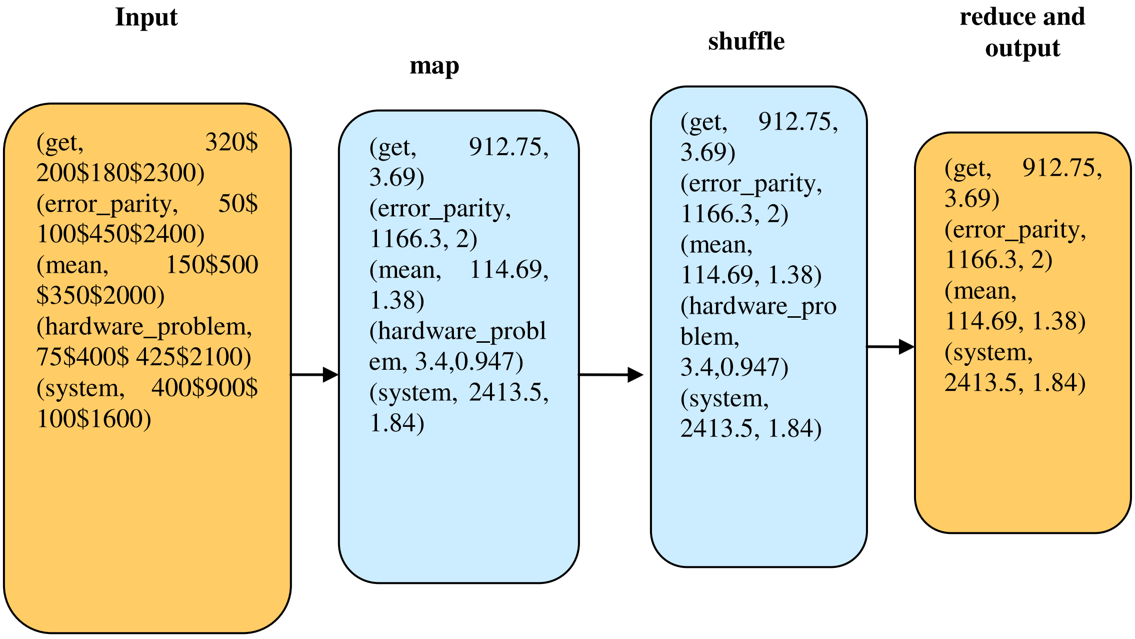
Input
map
shuffle
reduce and output
(get, 320$ 200$180$2300) (error_parity, 50$ 100$450$2400) (mean, 150$500 $350$2000) (hardware_problem, 75$400$ 425$2100) (system, 400$900$ 100$1600)
(get, 912.75, 3.69) (error_parity, 1166.3, 2) (mean, 114.69, 1.38) (hardware_problem, 3.4,0.947) (system, 2413.5, 1.84)
(get, 912.75, 3.69) (error_parity, 1166.3, 2) (mean, 114.69, 1.38) (hardware_problem, 3.4,0.947) (system, 2413.5, 1.84)
(get, 912.75, 3.69) (error_parity, 1166.3, 2) (mean, 114.69, 1.38) (system, 2413.5, 1.84)

Figure-3. Logical data flow of MapReduce module for applying CHIR to BOP representation.

RESULTS AND DISCUSSIONS

This section depicts the results of experiments carried out for various document representations with and without distributional features. This section illustrates the outcomes of representing documents as bag-of-phrases and discusses its significance. Table-2 list some of the useful bigrams selected by CHIR algorithm.

Table-2. List of semantically rich bigrams chosen by CHIR.

alt.atheism	comp.graphics	comp.os.ms-windows.misc	comp.sys. ibm. pc. hardware	comp.sys.mac.hardware	comp.windows.x
moral_ object hypothesi_ moral bibl_ read human_ valu make_ sens object_ real definit_ object capit_ punish	comput_ graphic bit_displai bit_imag graphic_ program virtual_real graphic_ silicon board_run digit_equip color_imag acceler_ graphic map_textur analysi_ imag file_format file_imag imag_ process ascii_tabl color_ window board_ standard	file_manag font_truetyp manag_ program laser_printer server_ window binary_file driver_ printer process_ word version_ window devic_driver access_disk ftp_site memori_scsi	parallel_ port board_ mother control_ drive card_ control driver_ video card_scsi memori_ system chip_ control port_serial drive_ floppi cach_ memori	appl_ comput appl_ monitor driver_ softwar color_ displai appl_dealer desktop_ machin appl_mous driver_ softwar architectur_mem ori memori_ virtual appl_ system	system_ window manag_ window desktop_ virtual memori_ share foundat_ softwar commun_ protocol inform_ system

The occurrences of the bigrams in the training documents were analyzed. The bigram “imag_process” occurred in different forms in sentences such as “XV to process the images”, “all screen images are the processed images”, “the processed image will be read back”, “Practical Image Processing in C”, “writing programs for processing an image”, “the image was processed”, and “family of image processing Virtual Instruments”. The algorithm failed to identify phrases such as “image and signal processing.”

Table-3. Percentage of bigrams chosen by CHIR.

Category	Percentage of bigrams
alt. atheism	14.52
comp.graphics	41.51
comp.os.ms-windows.misc	31.25
comp.sys.ibm.pc.hardware	41.34
comp.sys.mac.hardware	39.51
comp.windows.x	28.51

It could be observed that the number of bigrams chosen decreases for the category alt. atheism and increases for the category comp. graphics. This shows that bigrams are more important for related categories.

Table-4. Comparison of precision and recall for bag of phrases representation with and without DF.

Categories	Precision (P)	Recall (R)	P with DF	R with DF
alt.atheism	0.9823	0.8647	0.9877	0.9051
comp.graphics	0.8178	0.5613	0.8456	0.5891
comp.os.ms-windows.misc	0.7612	0.6316	0.8021	0.6783
comp.sys.ibm.pc.hardware	0.7245	0.6947	0.7571	0.7256
comp.sys.mac.hardware	0.6432	0.7619	0.6719	0.7781
comp.windows.x	0.6926	0.7351	0.7498	0.7614

CONCLUSIONS

In this research work documents are represented as bag of phrases and a statistical feature selection algorithm with distributional feature is proposed for text classification. As POS tagging of every term is done by connecting to WordNet, the time taken by the algorithm was high. Hence the algorithm was rewritten using MapReduce programming model and experimented in a distributed environment using Hadoop framework. It was observed that the classifier was able to give better accuracy for technical documents with bag of phrases representation. Inclusion of distributional features for feature selection in bag of phrases representation improved the performance of the text classifier.

REFERENCES

[1] Scott C. Deerwester, Susan T. Dumais, Thomas K. Landauer, George W. Furnas and Richard A. Harshman. 1990. Indexing by latent semantic analysis. Journal of the American Society of Information Science. 41(6): 391-407.

[2] Sebastiani. 2002. Machine learning in automated text categorization. ACM Computing Surveys, Consiglio Nazionaledelle Ricerche, Italy. 34: 1-47.

[3] Yiming Yang and Jan O. Pedersen. 1997. A Comparative Study on Feature Selection in Text Categorization. Proceedings of 14th International Conference: Machine Learning, Nashville, TN, USA. pp. 412-420.

[4] Yanjun Li, Congnan Luo and Soon M. Chung. 2008. Text Clustering with Feature Selection by Using Statistical Data. IEEE Transactions on Knowledge and Data Engineering. 20(5): 641-652.

[5] M. F. Caropreso, S. Matwin and F. Sebastiani. 2001. A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization. Text Databases and Document Management: Theory and Practice, Idea Group Publishing, Hershey, US. pp. 78-102.

[6] M Janaki Meena, KR Chandran, A Karthik. 2012. AV Samuel, An enhanced ACO algorithm to select features for text categorization and its parallelization. Expert Systems with Applications. pp. 5861-5871.

[7] Chade-Meng Tan, Yuan-Fang Wang and Chan-Do Lee. 2002. The use of bigrams to enhance text

categorization. Information Processing and Management. 38(4): 529-546.

[8] M. Dash and H. Liu. 1997. Feature Selection for Classification. IEEE Transaction on Intelligent Data Analysis. 1: 131-156.

[9] M Janaki Meena, KR Chandran, J Brinda, P Sindhu. 2010. Enhancing feature selection using statistical data with unigrams and bigrams, International Journal of Computer Applications.

[10] David D. Lewis. 1992. Feature Selection and Feature Extraction for Text Categorization. Proceedings of Speech and Natural Language Workshop. pp. 212-217.

[11] Dumais S.T., Platt J., Heckerman D., Sahami M. 1998. Inductive learning algorithms and representations for text categorization. Proceedings of the 7th International Conference: Information and Knowledge Management, ACM, New York, NY, USA. pp. 148-155.