INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

ENHANCED SEMANTICS BASED DISTRIBUTED DOCUMENT CLUSTERING: EXPERIMENTAL EVALUATION

NEEPA SHAH, DR. (MRS.) SUNITA MAHAJAN

1. Assistant Professor, IT Department, DJSCE, Vile Parle (West), Mumbai, India.
2. Principal, Institute of Computer Science, M. E. T., Bandra (west), Mumbai, India.

Accepted Date: 15/03/2016; Published Date: 01/05/2016

Abstract: Document clustering is an unsupervised learning paradigm where clustering methods try to identify inherent groupings of the text documents. The importance of document clustering emerges from the massive volumes of textual documents created. Distributed document clustering is needed due to huge volume of text documents being generated and due to limitations of client-server computing. In our approach, we used Hadoop and MapReduce open source platforms for distributed document clustering. We tested our approach on 20-NewsGroup and Reuters-21578 document collection. This paper is the extended version "Efficient Pre-Processing for Enhanced Semantics Based Distributed Document Clustering" with more experimental evaluation. We performed more experiments to see the scalability of Hadoop cluster. Another set of experiment is to generate various clusters (5, 6, 7, and 8 on 20NG) with different count of iterations for refining the clusters and evaluating their performance improvement by using enhanced Stemmer algorithm.

Keywords: Document Clustering, Semantic Document Clustering, Distributed Document Clustering, Hadoop and MapReduce for document clustering

Corresponding Author: MS. NEEPA SHAH

Access Online On:

www.ijpret.com

How to Cite This Article:

Neepa Shah, IJPRET, 2016; Volume 4 (9): 1087-1097

PAPER-QR CODE

INTRODUCTION

There is huge volume of data in form of documents due to the technological advancement in computer hardware, powerful computers, storage media, and data collection equipment. So, it is becoming very hard to organize, analyse and present these documents efficiently [1]. These have brought challenges for the effective and efficient organization of text documents automatically [2]. This has led to the document clustering, an unsupervised machine learning approach; which organizes documents into different groups called as clusters. The documents in each cluster share some common properties according to some defined similarity measure. It plays an important role in helping users to effectively navigate, summarize, and organize the information [3].

The paper is organized as: Section II highlights related work in the area of document clustering, semantics based methods for document clustering, and distributed document clustering along with overview of our base paper. We give all results in section III. Here, we give experiments and results for K-Means for two datasets namely 20-NewsGroup and Reuter-21578. We cover experimental cases like adding more number of nodes, executing with different number of iterations and different number of clusters. We are using entropy and purity for evaluating clustering performance. We give results of basic dataset and compared the results with pre-processed datasets over parameters like time, entropy and purity. We conclude the paper in section IV.

RELATED WORK

In this section, related work in the area of document clustering like various basic algorithms and methods, parallel and distributed document clustering techniques, and Hadoop / MapReduce framework for document clustering is briefly discussed.

Various traditional methods are used for document clustering, including k-means (KM), hierarchical agglomerative clustering (HAC), their variations, clustering using concepts of frequent-itemset, fuzzy theory, neural network, genetic algorithm, self-organizing map, non-negative matrix factorization etc. These algorithms are studied and compared in [4].

Many semantics based approaches like latent semantic indexing, frequent word phrases, WordNet, ontology, part-of-speech tagging, sense disambiguation, machine learning have been studied and survey of these methods is given in [5].

As mentioned earlier, for distributed approach we are using Hadoop framework with MapReduce programming model. Various parallel and distributed algorithms and role of Hadoop and MapReduce in distributed clustering applications is covered in detail in [6]. The distributed document clustering using K-means document clustering algorithm is given in [7].

The base paper "Efficient Pre-Processing for Enhanced Semantics Based Distributed Document Clustering" gives complete highlight our approach and dataset pre-processing. We have discussed other modules related to incorporating semantics to our document clustering which includes POS tagging using Stanford Parser [8] and Word Sense Disambiguation using WordNet Dictionary [9]. The proposed pre-processing reduced the size of the original dataset by factor of 2 which overcame the main drawback of dimensionality curse encountered in traditional document clustering. We have compared results of our approach with traditional approach and given the improvements with relevant discussion.

Here, we are presenting extended experiments on same approach. We give the results of varying "n", which is adding number of nodes to the cluster. We also evaluated outcome of 20NG with 5, 6, 7, and 8 clusters as 5 clusters in base paper was giving better results. Similar to base paper, we then give improvement over traditional approach.

EXPERIMENTS AND RESULTS

As mentioned earlier, this paper presents extended experiments to paper "Efficient Pre-Processing for Enhanced Semantics Based Distributed Document Clustering". We further ran 3 iterations on complete fully pre-processed 20NewsGroup dataset with varying size of nodes i.e. gradually increasing the new data node in the cluster. The results found for time taken to cluster the documents is analysed using graph in Figure 1.

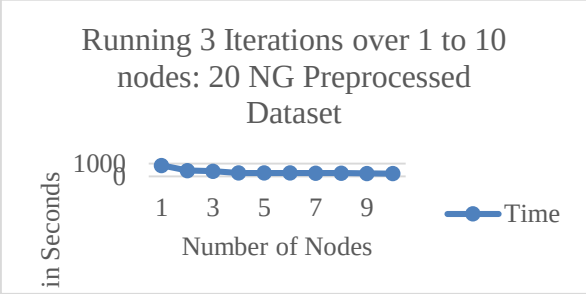


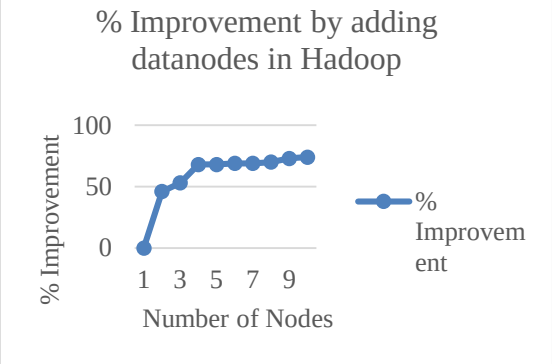
Figure 1: Graph showing reduction in time by increasing number of data nodes for 20NG dataset clustering

To find out the stability and scalability of increasing number of nodes of the cluster, we calculated the percentage improvement after adding each data node. The results of which are given below in Table 1. As it can be seen through Figure 2, after 4th node there is no significant improvement. So, for further experiments we considered cluster of 5 nodes.

Table 1: % improvement in time for clustering 20NG dataset on 1-10 nodes

Nodes	Time	% Improvement
1	858	0
2	460	46.38694639
3	402	53.14685315
4	276	67.83216783
5	270	68.53146853
6	266	68.997669
7	262	69.46386946
8	260	69.6969697
9	230	73.19347319
10	226	73.65967366

Figure 2: Graph showing % improvement in time by adding new data node for 20NG dataset clustering



We then performed experiments to see the stability of entropy and purity over number of iterations for which we ran the clustering algorithm on pre-processed 20NG dataset with 1 to 10 iterations on 5 nodes cluster. In this, we analysed the time taken to cluster the documents which obviously is linearly increasing with count of iteration. The linear growth of the time taken to cluster the 20NG dataset over 1 to 10 iterations is clearly seen in Figure 3.

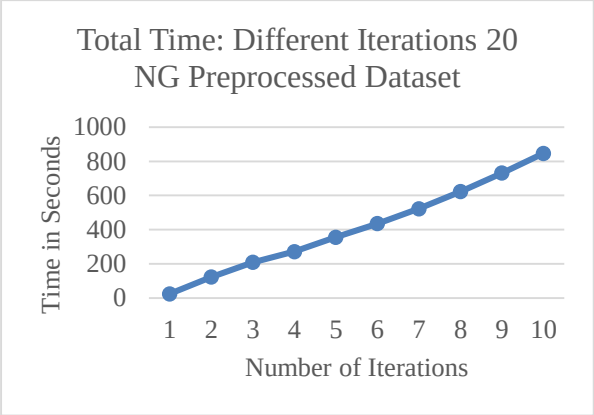


Figure 3: Graph showing increase in time by increasing number of iterations for 20NG dataset clustering

We also calculated entropy and purity values for this experiment setup and compared it. This was to finalize the best iterations for clustering 20NG dataset. The Table 2 given below shows Entropy and Purity values for clustering 20NG dataset for different iterations by varying it from 1 to 10. The stability of Entropy and Purity can be seen in Figure 4 and Figure 5 respectively. It is clear through these graphs that there is no much difference in both the values of Entropy and Purity by changing iterations above 3; rather it is worsening.

Table 2: Entropy and Purity values for clustering 20NG dataset for 1-10 iterations

Iterations	Entropy	Purity
1	0.484242258	0.319128659
2	0.484191186	0.319227106
3	0.484170207	0.319374575
4	0.484128931	0.319319728
5	0.484119751	0.319438769
6	0.483918761	0.320256971
7	0.483857071	0.320692498
8	0.483457759	0.320278384
9	0.483266383	0.320342795
10	0.483235172	0.320428766

\

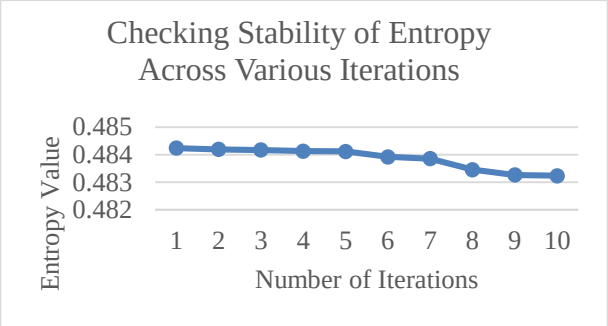


Figure 4: Graph showing Entropy values for 1 to 10 iterations for 20NG dataset clustering

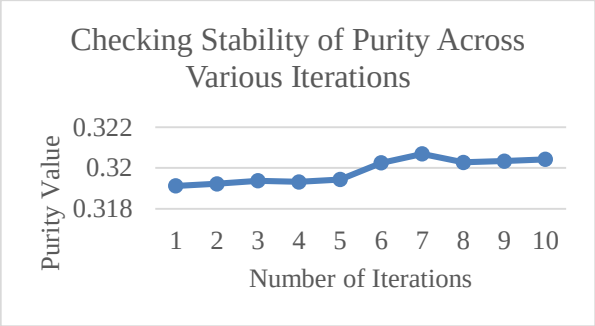


Figure 5: Graph showing Purity values for 1 to 10 iterations for 20NG dataset clustering

From above experiments, we concluded to use the cluster of 5 nodes and 3 iterations for the datasets. As in paper "Efficient Pre-Processing for Enhanced Semantics Based Distributed Document Clustering", for 20NG dataset 5 clusters gave better results compared to 10, 15 and 20; we further experimented to run the distributed K-Means algorithm for generating 5, 6, 7, and 8 clusters. As concluded from above experiments, the setup used is 5 nodes cluster and 3 iterations. Again, we calculated total time, entropy and purity values. The time taken for generating these clusters is given below in Figure 6.

Figure 6: Graph showing time for 20NG dataset clustering for 3 iterations and generating 5,6,7,8 clusters

For entropy and purity values, we ran the experiment of clustering over basic Stemmer algorithm and revised Stemmer algorithm. This was to identify the improvement used by our approach of using revised Stemmer over basic Stemmer algorithm. The outcome of these values for basic Stemmer algorithm is given in Table 3 and for revised Stemmer algorithm is given in Table 4.

Table 3: Entropy and Purity for 3 iterations and generating 5, 6, 7, 8 clusters: Basic Stemmer, 20NG dataset

Clusters	Entropy	Purity
5	0.518795066	0.276412134
6	0.534641021	0.278842881
7	0.62767953	0.282214326
8	0.642606769	0.222824124

Table 4: Entropy and Purity for 3 iterations and generating 5, 6, 7, 8 clusters: Revised Stemmer, 20NG dataset

Clusters	Entropy	Purity
5	0.526843474	0.29184857
6	0.550064927	0.32563716
7	0.640519671	0.31268436
8	0.634138214	0.24893071

The respective graph analysis for Entropy and Purity for these experiments for basic Stemmer is given in Figure 7 and Figure 8 respectively. It is clear that it is giving better results for 5, 6, and 7 clusters.

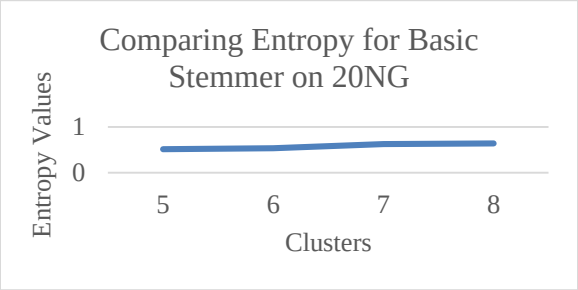


Figure 7: Entropy values for 3 iterations and generating 5, 6, 7, 8 clusters: Basic Stemmer, 20NG dataset

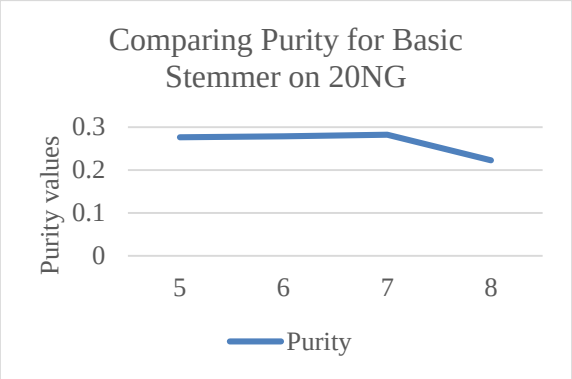


Figure 8: Purity values for 3 iterations and generating 5, 6, 7, 8 clusters: Basic Stemmer, 20NG dataset

Same way, the graphs for Entropy and Purity values for revised Stemmer is given in Figure 9 and Figure 10 respectively.

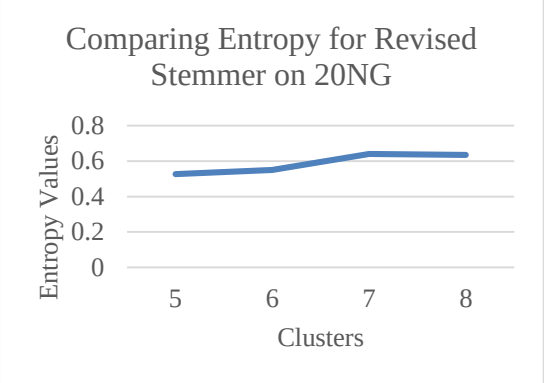


Figure 9: Entropy values for 3 iterations and generating 5, 6, 7, 8 clusters: Revised Stemmer, 20NG dataset

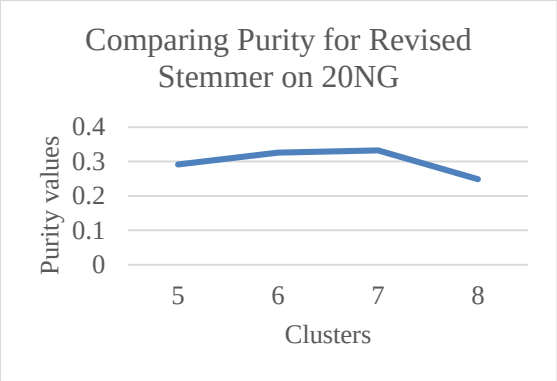


Figure 10: Purity values for 3 iterations and generating 5, 6, 7, 8 clusters: Revised Stemmer, 20NG dataset

From these experiments, we then calculated percentage improvement achieved by applying revised Stemmer algorithm. The details of this improvement is given in

Table 5 and graphical representation of the same is given in Figure 11.

Table 5: % Improvement in Entropy and Purity: Basic and Revised Stemmer Algorithm

Clusters	5	6	7	8
Entropy	1.55136	2.88490	2.04565	-1.31784
Purity	5.584572	16.7815	10.7967	11.71622961

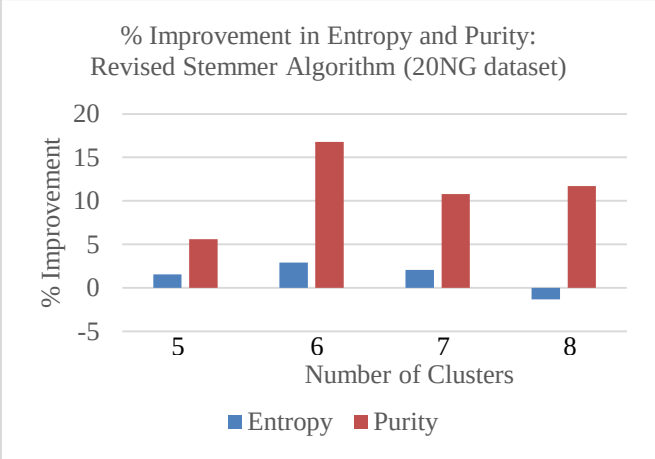


Figure 11: Entropy and Purity: % Improvement through Revised Stemmer, 20NG dataset

It can be clearly seen that both the Entropy and Purity gives best improvement for 6 clusters. It is also seen that it gives reverse performance for 8 clusters.

CONCLUSION AND FUTURE SCOPE

As the volume of information continues to increase, there is growing interest in helping people better find, filter and manage these resources. Text clustering, which is the process of grouping documents having similar properties based on semantic and statistical content, is an important component of document organization and management.

Here, we presented extended experiments to our earlier paper "Efficient Pre-Processing for Enhanced Semantics Based Distributed Document Clustering". In that paper, we had discussed distributed document clustering using Hadoop framework and K-means algorithm. For semantics, Part-Of-Speech (POS) and WordNet was used.

For stemming, we proposed to use modified Porter's stemmer algorithm and found that modified stemming gives better results in terms of purity and entropy. In this paper, we compared basic Porter Stemmer algorithm with the revised one and found significant improvement in both Entropy and Purity values.

Further, we experimented to apply clustering algorithm to generate 5, 6, 7, and 8 clusters instead of 5, 10, 15, and 20 clusters for 20NG dataset. This is because, 5 clusters had given better outcome compared to 10, 15, and 20. Through this experiment, we derived that 5, 6, and 7 clusters give better results whereas for 8 clusters the performance in terms of entropy and purity is deteriorating.

We also demonstrated reduction in time for execution of the algorithm over a cluster of 1 to 10 nodes. Through this experiment, we conclude that after 4-5 nodes of cluster there is no significant improvement. This could be due to overhead of interaction in the cluster.

For further work, we propose to test distributed document clustering on still large datasets like RCV1. We also propose to use bisecting k-means for clustering as it would give better results than k-means algorithm

REFERENCES

1. Rekha Baghel and Dr. Renu Dhir, "A Frequent Concepts Based Document Clustering Algorithm," Int'l Journal of Computer Applications, vol. 4, No.5, pp. 0975 – 8887, Jul. 2010

2. Huang, “Similarity measures for text document clustering,” Proc. of the 6th New Zealand Computer Science Research Student Conference NZCSRSC, pp. 49-56, 2008.
3. Nicholas O. Andrews and Edward A. Fox, “Recent developments in document clustering,” Technical report published by citeseer, pp. 1-25, Oct. 2007.
4. Neepa Shah and Sunita Mahajan, “Document Clustering: A Detailed Review,” Int’l Journal of Applied Information Systems, Vol. 4, No. 5, pp.30-38, Oct. 2012.
5. Neepa Shah and Sunita Mahajan. “Semantic based ocument Clustering: A Detailed Review,” Int’l Journal of Computer Applications Vol. 52, No. 5, pp.42-52, Aug. 2012.
6. Neepa Shah and Sunita Mahajan, “Semantics Based Distributed Document Clustering: Proposal,” Int’l Journal of Computer Science Engineering and Information Technology Research, Vol. 3, Issue 2, pp. 379-388, Jun. 2013.
7. Neepa Shah and Sunita Mahajan, “Distributed Document Clustering Using K-Means”, Int’l Journal of Advanced Research in Computer Science and Software Engineering, Vol. 4, Issue 11, pp. 24-29, Nov. 2014.
8. Stanford CoreNLP Parser, available at http://nlp.stanford.edu/software/corenlp.shtml visited on 6th Mar. 2014.
9. WordNet, A lexical Database for English, available at http://WordNet.princeton.edu/