International Journal of Computer Science Engineering & Technology
APPROVED BY
NATIONAL SCIENCE LIBRARY (NSL)
NATIONAL INSTITUTE OF SCIENCE COMMUNICATION AND INFORMATION RESOURCES (NISCAIR)
COUNCIL OF SCIENTIFIC AND INDUSTRIAL RESEARCH (CSIR)- NEW DELHI INDIA .
ISSN :2455-9091

Novel Clustering Method For The Categorical Data Using Mathematical Fuzzy Partitioning

[1] M. Porkizhi, [2] Dr. J. Thirumaran
[1] Ph.d Research Scholar, [2] Ph.D Research Supervisor,
[1, 2] Bharathiar University, India.

ABSTRACT: Record summarization gives an instrument to quicker understanding the gathering of text reports and has various genuine applications. Semantic comparability and clustering can be used proficiently to generate viable outline of extensive text accumulations. Condensing substantial volume of text is a testing and tedious issue especially while considering the semantic likeness calculation in summarization prepare. Summarization of text accumulation includes escalated text preparing and calculations to create the synopsis. In this paper, a novel framework in light of MapReduce innovation is proposed for condensing vast text gathering. The proposed method is planned utilizing Distributed Collaborative Document Clustering System and subject displaying utilizing Latent Dirichlet Allocation (LDA) for abridging the vast text gathering over MapReduce framework. The exhibited method is assessed as far as versatility and different text summarization parameters to be specific; pressure proportion, maintenance proportion, ROUGE and Pyramid score are additionally measured.

Keywords: [LDA, Text Clustering, Summarization, Map Reduce Framework]

1. INTRODUCTION

Clustering can be connected to many sorts of information, the concentrate of this theory is on clustering text archives, a field referred to in the writing as record clustering which is a subfield of text mining. Record clustering manages the unsupervised parceling of a report accumulation into significant gatherings in view of their textual substance, as a rule with the end goal of theme order; i.e. records in one group have a place with a specific point, while distinctive bunches speak to various subjects. Report clustering has numerous applications, for example, clustering of web search tool results to show sorted out and reasonable outcomes to the client (e.g. Vivisimo1), clustering archives in a gathering (e.g. computerized libraries), robotized (or semi-mechanized) formation of report scientific classifications (e.g. Yippee! also, Open Directory styles), and productive data recovery by concentrating on important subsets (groups) as opposed to entire accumulations.

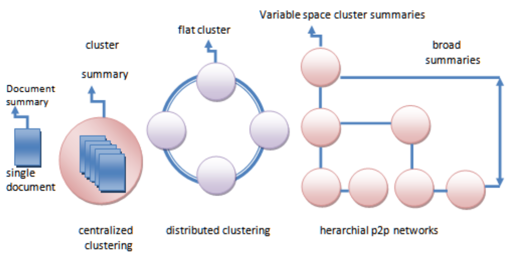

Figure 1: Levels of Clustering and Summarization

Text summarization is one of the critical and testing issues in text mining. It gives various advantages to clients and various productive genuine applications can be created utilizing text summarization. In text summarization a substantial accumulations of text archives are changed to a diminished and reduced text report, which speaks to the process of the first text accumulations. An outlined report helps in understanding the substance of the extensive text accumulations rapidly and furthermore spare a great deal of time by abstaining from perusing of every individual record in a vast text gathering. Text mining is utilized to portray diverse applications, for example, text classification, text clustering, observational computational semantic errands, and exploratory information investigation, discovering designs in text databases, finding successive examples in text and affiliation revelation.

Most text mining techniques utilize the Vector Space Model, acquainted by Salton in 1975 with speak to report objects. Each record is spoken to by a vector d, in the term space, d = {tf1, tf2... tfn}, where tfi, i = 1. . . n is the term recurrence in the report, or the quantity of events of the term ti in an archive. To speak to each report with a similar arrangement of terms, we need to concentrate every one of the terms found in the records and utilize them as our element vector. Once in a while another technique is utilized which clearly the dimensionality of the element vector is constantly high, in the scope of hundreds and some of the time thousands. Joins the term recurrence with the backwards record recurrence (TF-IDF). The archive recurrence dfi is the quantity of records in an accumulation of N reports in which the term ti happens. A run of the mill backwards archive recurrence (idf) component of this sort is given by log (N/dfi). The heaviness of a term ti in an archive is given by wi = tfi × log (N/dfi).

The calculation plays out the assignment of text summarization is called as text summarizer. The text summarizers are extensively arranged in two classes which are single-archive summarizer and multi-report summarizers. In single-archive summarizers, a solitary extensive text report is condensed to another single record outline, while in multi-report summarization, an arrangement of text records (multi reports) are abridged to a solitary archive rundown which speaks to the general look at the various reports. Multi-record summarization is a method used to outline various text archives and is utilized for seeing huge text report accumulations. Multi-report summarization produces a reduced rundown by removing the applicable sentences from a gathering of records on the premise of archive subjects. In the current years scientists have given much consideration towards creating archive summarization procedures.

Record Index Graph (DIG) show in which hubs speak to special words alongside term recurrence data, and edges speak to groupings of words. Since this model is utilized as the fundamental portrayal demonstrate in the key-expression extraction calculation. A concise meaning of the DIG model is given here.

The DIG is a coordinated diagram (digraph) G = (V,E) where V : is an arrangement of hubs {v1, v2, . . . , vn}, where every hub v speaks to a special word in the whole record set; and E : is an arrangement of edges {e1, e2, . . . , em}, with the end goal that each edge e is a requested combine of hubs (vi, vj). An edge from vi to vj demonstrates that the word vj seems progressive to the word vi in some archive.

Each record di is mapped to an archive sub-chart gi that speaks to the one of a kind words and their arrangements in that report (i.e. phrases). The DIG model is fabricated incrementally by combining each archive sub-chart into an aggregate diagram that speaks to records prepared up to di: Gi = Gi−1 ∪ gi. After combining an archive sub-diagram into the total chart, it is conceivable to remove the coordinating expressions between the new record and every past report. The rundown of coordinating expressions between record di and dj is figured by crossing the subgraphs of both archives, gi and gj , separately. Give pij a chance to signify such rundown, at that point: pij = gi ∩ gj A rundown of coordinating expressions between archive di and all already handled reports is registered by meeting the record sub-chart gi with the aggregate DIG Gi−1. Give pi a chance to mean such rundown, at that point: pi = gi ∩ Gi−1 This procedure produces finish state coordinating yield between each combine of reports in close direct time, with subjective length phrases.

2. CHALLENGES IN CLUSTERING

There are a number of problems associated with clustering, which are outlined here:

- Choice of a good (dis)similarity measure,
- Choice of the number of clusters,
- Ability to perform incremental update of clusters without re-clustering,
- Properly dealing with outliers,
- Interpretation of clustering results,
- Tackling distributed data,
- Scalability, both in terms of the number of objects and the no of dimensions,
- Evaluation of clustering quality.

Three of challenges are addressed interpretation of clustering results, scalability, and tackling distributed data.

Interpreting clustering results is addressed through document cluster summarization using a novel key-phrase extraction algorithm, while scalability and tackling distributed data are addressed through novel distributed clustering algorithms.

3. BACKGROUND AND LITERATURE REVIEW

MapReduce is a well known programming model for preparing expansive informational collections. It offers various advantages in taking care of huge informational indexes, for example, adaptability, adaptability, adaptation to internal failure and various different favorable circumstances. Lately various works are introduced by specialists in field of Big Data investigation and huge informational collections handling. The difficulties, openings, development and points of interest of MapReduce framework in dealing with the Big Data is displayed in various reviews. MapReduce framework is broadly utilized for handling and overseeing extensive informational indexes in a conveyed bunch, which has been utilized for various applications, for example, report clustering, get to log investigation, creating seek records and different other information scientific operations. A large group of writing is available lately to perform Big Data clustering utilizing MapReduce framework. An adjusted K-implies clustering calculation in light of MapReduce framework is proposed by Li et al. to perform clustering on huge informational collections. For breaking down huge information and mining Big Data MapReduce framework is utilized as a part of various works. A portion of the work displayed toward this path is web log investigation , coordinating for web-based social networking, outline and usage of Genetic Algorithms on Hadoop , social information examination , fluffy control based arrangement framework , log joining , online component determination , visit thing sets mining calculation and compacting semantic web articulations . Taking care of extensive text is an exceptionally troublesome assignment especially in learning revelation prepare

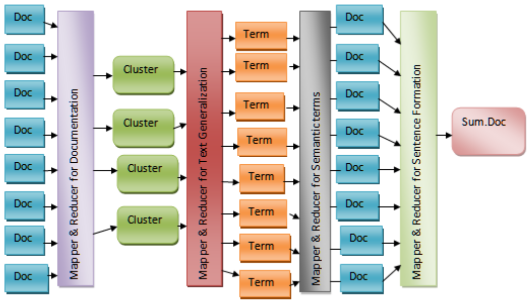

Figure 3.1 : Stages in MapReduce framework for multi document summarization

MapReduce framework is effectively used for a quantities of text preparing errands, for example, stemming, disperse the capacity and calculation stacks in a bunch, text clustering, data extraction, putting away and getting unstructured information, report likeness calculation characteristic dialect handling and pairwise archive closeness. Outlining expansive text gathering is an intriguing and testing issue in text examination. A quantities of methodologies are recommended for taking care of huge text for programmed text summarization. A strategy is proposed by Lai and Renals, for meeting summarization utilizing prosodic components and expand lexical elements. Highlights identified with exchange acts are found and used for meeting summarization. An unsupervised technique for the programmed summarization of source code text is proposed by Fowkes et al. The proposed system is used for code collapsing, which enables one to specifically conceal pieces of code. A multi-sentence pressure procedure is proposed by Tzouridis et al. A parametric most limited way calculation utilizing word charts is introduced for multisentence compressions. A parametric method for edge weights is utilized for producing the coveted synopsis. Parallel usage of Latent Dirichlet Allocation in particular, PLDA is proposed by Wang et al. The usage is conveyed utilizing MPI and MapReduce framework. It is exhibited that PLDA can be connected to extensive, genuine applications and furthermore accomplishes great versatility.

4. METHODOLOGY:

The shared archive clustering framework depends on three segments: an underlying clustering calculation utilizing comparability histogram-based clustering (SHC), a bunch summarization calculation (CorePhrase) and an appropriated report clustering calculation in light of trade of group outlines, suggestion and converging of associate records. Starting clustering is performed utilizing a Similarity Histogram-based Clustering (SHC).The coherency of a group is spoken to as a Cluster Similarity Histogram. Bunch Similarity Histogram: A brief factual portrayal of the arrangement of combine shrewd record similitudes circulation in the group. Various receptacles in the histogram compare to settled closeness esteem interims. Each receptacle contains the number of match savvy record similitudes in the relating interim. Likeness. With the end goal of this work, we characterize the likeness between two records as the proportion of their basic elements to the union of their elements

sim(di, dj) = di ∩ dj di ∪ dj

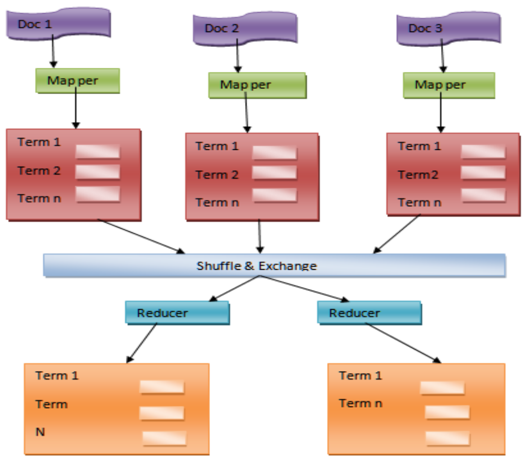

Figure 3.2 : Frequent terms counting from text collection using MapReduce framework

Dispersed Document Clustering and Cluster Summarization in P2P Environments. In the event that every document is spoken to as a vector of watchword weights, we can ascertain the comparability between a couple of records utilizing the generally utilized cosine coefficient: sim(di, dj) = cos(di, dj) =di .aThis cosine measure is utilized as a part of our trials to ascertain report to-record likeness. Notwithstanding which comparability work we pick, the similitude histogram idea stays nonpartisan to our decision. The main prerequisite is that the similitude measure constitutes a metric on the report vector space. A cognizant group ought to have high pairwise record similitudes. A run of the mill group has an ordinary circulation, while a perfect bunch would have a histogram where all likenesses are most extreme. We judge the nature of a likeness histogram (group cohesiveness) by computing the proportion of the number of similitudes over a specific closeness edge RT to the aggregate tally of likenesses. The higher this proportion, the more durable the group. Give NDc a chance to be the quantity of the reports in a group. The quantity of match astute similitudes in the bunch is NRc = NDc(NDc + 1)/2. Let R = {ri : i = 1, . . . ,NRc} be the arrangement of likenesses in the group. The histogram of the likenesses in the group is spoken to as:

$$Hc = \{hi : 1 \le i \le B\} \quad (4.1a)$$

hey = count(rk), $\delta \cdot (i - 1) \le rk < \delta \cdot i$

where B : is the quantity of histogram receptacles, howdy : is the include of likenesses container i, and δ : is the canister width of the histogram. The histogram proportion (HR) of a group, which shows bunch cohesiveness, is ascertained as:

HR(c) =PB

i=T hello there PB

j=1 hj

T = ⌊RT ・ B⌋

where HR(c) : the histogram proportion of group c, RT : the likeness limit, and T : the canister number relating to the comparability edge.

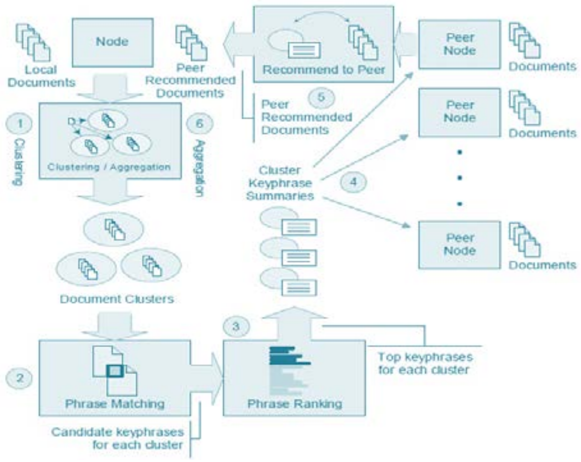

Figure 4.1: Distributed Collaborative Document Clustering System

Comparability Histogram-based Clustering calculation works by keeping up high HR for each group. New records are tried against each bunch, adding them to suitable groups on the off chance that they don't debase the HR of that bunch fundamentally. Arrangements are likewise made so as not to permit a chain response of "awful" reports being added to a similar group, in this way cutting its cohesiveness down essentially. The calculation works incrementally by emphasizing over the records at hub i, and for each bunch ascertains the group histogram proportion previously, then after the fact mimicking the expansion of the report to that group. On the off chance that the new proportion is more prominent than or equivalent to the old one, the report is added to the bunch. Generally on the off chance that it is not as much as the old proportion by close to ε and still above HRmin, it is included. Else it is not included. In the event that the report was not relegated to any bunch, another group is made to which the record is included.

A)LATENT DIRICHLET ALLOCATION:

Inert Dirichlet Allocation (LDA) is a well known subject displaying procedure which models text reports as blends of dormant points, which are enter ideas exhibited in the text. A subject model is a likelihood appropriation method over the accumulation of text reports, where each archive is demonstrated as a mix of points, which speaks to gatherings of words that have a tendency to happen together. Every point is displayed as a likelihood dispersion φk over lexical terms. Every point is exhibited as a vector of terms with the likelihood in the vicinity of 0 and 1. A record is demonstrated as a likelihood dispersion over themes In LDA, the subject blend is drawn from a conjugate Dirichlet earlier that is the same for all archives. The point displaying for text gathering utilizing LDA is performed in four stages. In the initial step a multinomial θt circulation for every point t is chosen from a Dirichlet appropriation with parameter β. In second step for each report d, a multinomial appropriation θb is chosen from a Dirichlet dissemination with parameter α. In third step for each word w in record s a subject t from θb is chosen.

B) K-MEANS CLUSTERING ALGORITHM

Clustering is a process of creating groups of similar objects. Clustering algorithms are categorized into five major categories namely, Partitioning techniques, Hierarchical techniques, Density Based techniques, Grid Based techniques and Model based techniques. Partitioning techniques are the simplest techniques which creates K number of disjoint partitions to create K number of clusters. These partitions are created using certain statistical measures like mean, median etc. K-means is a classical unsupervised learning algorithms used for clustering. It is a simple, low complexity and a very popular clustering algorithm. The k-means algorithm is a partitioning based clustering algorithm. It takes an input parameter, k i.e. the number of clusters to be formed, which partitions a set of n objects to generate the k clusters. The algorithm works in three steps. In the first step, k number of the objects is selected randomly, each of which represents the initial mean or center of the cluster. In the second step, the remaining objects are assigned to the cluster with minimum distance from cluster center or mean. In the third step, the new mean for each cluster is computed and the process iterates until the criterion function converges.

C) EXPERIMENTS AND RESULT ANALYSIS

Summarization procedures are ordered into two noteworthy classifications extractive or abstractive. Extractive summarization allots a channel and concentrates the sentences with most astounding coordinating criteria to shape the outlines. Abstractive summarization, then again, utilizes certain level of comprehension of the substance communicated in the first records and makes the synopses in light of data combination. Like most scientists in this field, the extractive summarization framework in utilized as a part of this work. Three noteworthy necessities for multi-report summarization are clustering, scope and hostile to repetition. Clustering is the capacity to bunch comparative archives and entries to discover related data, scope is the capacity to discover and remove the primary focuses crosswise over reports and hostile to repetition is the capacity to limit excess between sections in the outline. Clustering prerequisite is accomplished with the assistance of K-Means calculation to assemble the comparative records with the normal topics and furthermore is the piece of proposed strategy. Scope and hostile to repetition is accomplished with the assistance of sentence separating while at the same time creating the last synopsis.

D) SUMMARIZATION EVALUATION

Text summarization process is significantly assessed utilizing execution parameters to be specific, Compression Ratio (CR), Retention Ratio (RR), ROUGE score and Pyramid score.

E) COMPRESSION AND RETENTION RATIO

The Compression Ratio (CR) is the proportion of size of the condensed text report to the aggregate size of the first text archives. Maintenance Ratio (RR) is the proportion of the data accessible in the condensed record to the data accessible in the first text accumulations.

F) RESULT ANALYSIS

The versatility is figured utilizing diverse hubs and distinctive quantities of text record reports for producing the synopsis utilizing the proposed MapReducer based summarizer. Adaptability tends to increment in extent to the quantity of text archives with greatest quantities of hubs. Time to register the synopsis tends to diminish with increment in number of hubs. As the hubs builds the calculation time keeps an eye on direct and up to four hubs it turns out to be recently straight in proportionate to the quantity of text archives partaking in rundown. At the point when the quantity of hubs are changed from one to two the computational time defeat in exponential behavior and when the hubs comes to up to four the computational time ends up noticeably direct with proportionate to the quantity of text report gathering. The execution parameters of proposed summarizers i.e. pressure proportion, maintenance proportion, ROUGE and Pyramid scores are assessed for three unique situations. The summarizers are assessed for the accompanying three cases:

Case 1: Summarization without performing clustering and semantic similarity.

Case 2: Summarization with clustering but without considering semantic similarity.

Case 3: Summarization by considering both clustering and semantic similarity.

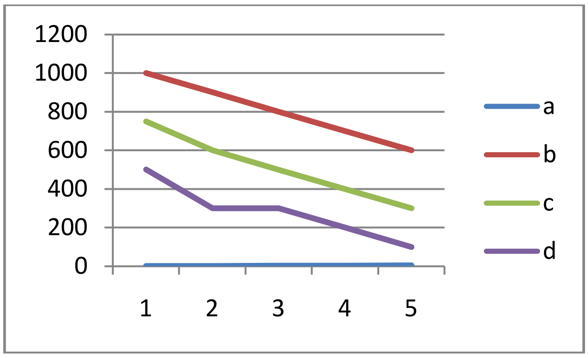

Figure 4.2: Scalability of MapReducer based summarizer

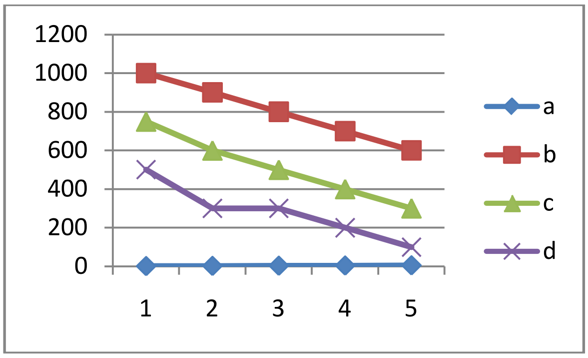

Figure 4.3: Time in ms for summarizing the text reports

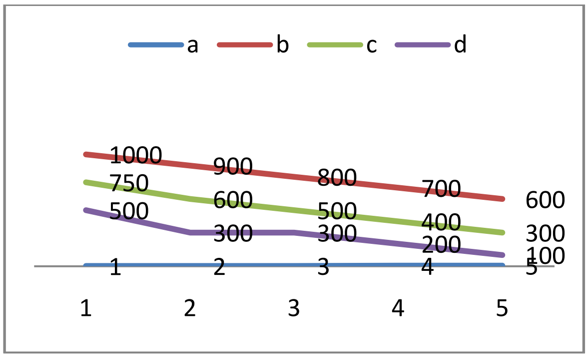

Figure 4.4 : Compression ratio for different cases

These outcomes plainly shows that semantic comparability alongside the clustering gives better summarization comes about when contrasted with the summarization without semantic likeness and clustering. Semantic closeness gives significant gathering of comparative text portions as summarization substance units for creating synopsis of the text accumulations. Semantic comparability guarantees better lumping of important text bunches when contrasted with the plain clustering of text records. Semantic closeness alongside clustering gives a system of support of the distinctive summarization content units from the diverse gatherings of text reports. Higher pyramid scores demonstrating that generally a greater amount of the substance is as exceptionally bweighted as could be allowed. High pyramid score mirrors the more noteworthy probability that more SCUs (Summarization Content Units) in the rundown show up in the pyramid. Much the same as the ROUGE score, greatest pyramid score is accomplished for the case III, where both semantic and textual closeness (clustering) is considered for compressing the text accumulations. It is likewise demonstrated that clustering (gathering the comparable text sections) gives better summarization in context to the summarization performed with non-grouped text accumulations. Clustering gives better summarization units (text fragments) for compressing the text accumulations. It is additionally certain that clustering alongside the semantic similitude gives better summarization content units to creating outline from the text accumulations. To better show the consequences of the distinctive cases, Fig. 15 outwardly show the examination. Figures exhibits bug outline demonstrating the correlations of the three distinct cases, it is plainly noticeable from the diagram that the estimations of execution parameters for case-III (considering both the clustering with semantic closeness) gives better outcomes when contrasted with whatever is left of the two cases.

5. CONCLUSIONS AND FUTURE ENHANCEMENTS

A multi-archive text summarizer in view of MapReduce framework is introduced in this work. Analyses are conveyed utilizing something like four hubs in MapReduce framework for a huge text accumulation and the summarization execution parameters pressure proportion, maintenance proportion and calculation timings are assessed for an expansive text gathering. It is likewise demonstrated tentatively that Map Reduce framework gives better adaptability and diminished time unpredictability while considering huge number of text records for summarization. Three conceivable instances of condensing the various archives are likewise examined similarly. It is demonstrated that viable summarization is performed when both clustering and semantic likeness are considered. Considering semantic comparability gives better maintenance proportion, ROUGE and pyramid scores for rundown. Future work toward this path can be giving the support to multi lingual text summarization over the MapReduce framework keeping in mind the end goal to encourage the synopsis era from the text archive accumulations accessible in various dialects.

REFERENCES:

[1] Shim K (2012) MapReduce Algorithms for Big Data Analysis, Framework. Proceedings of the VLDB Endowment 5(12):2016–2017

[2] Lee K-H, Lee Y-J, Choi H, Chung YD, Moon B (2011) Parallel Data Processing with MapReduce: A Survey. ACM SIGMOD Record 40(4):11–20

[3] Yang J, Li X (2013) MapReduce Based Method for Big Data Semantic Clustering. In Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference. Manchester, England, pp 2814–2819

[4] Ene A, Im S, Moseley B (2011) Fast Clustering using MapReduce. Proc. of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, New York, USA, pp 681–689

[5] Kolb L, Thor A, Rahm E (2013) Don't Match Twice: Redundancy-free Similarity Computation with MapReduce. Proc.of the Second Workshop on Data Analytics in the Cloud, ACM, New York, USA, pp 1–5

[6] Esteves RM, Rong C (2011) Using Mahout for clustering Wikipedia's latest articles: a comparison between K-means and fuzzy C-means in the cloud. In Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference. Athens, Greece, pp 565–569

[7] Li HG, Wu GQ, Hu XG, Zhang J, Li L, Wu X (2011) K-means clustering with bagging and mapreduce. Proc. 2011 44th Hawaii International Conference on IEEE System Sciences (HICSS). Kauai/Hawaii, US, pp 1–8

[8] Zhang G, Zhang M (2013) The Algorithm of Data Preprocessing in Web Log Mining Based on Cloud Computing. In 2012 International Conference on Information Technology and Management Science (ICITMS 2012) Proceedings Springer. Berlin, Heidelberg, Germany, pp 467–474

[9] Morales GDF, Gionis A, Sozio M (2011) Social content matching in mapreduce. Proceedings of the VLDB Endowment 4(7):460–469

[10] Verma A, Llora X, Goldberg DE, Campbell RH (2009) Scaling Genetic algorithms using MapReduce. Intelligent Systems Design and Application(ISDA). Ninth International Conference, Pisa, Italy, pp 13–18.

[11] D. Boley, M. Gini, R. Gross, S. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Document categorization and query generation on the World Wide Web using WebACE. AI Review, 13(5-6):365–391, 1999.

[12] D. Boley, M. Gini, R. Gross, S. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Partitioning-based clustering for web document categorization. Decision Support Systems, 27:329–341, 1999.

[13] Christopher Brooks, Scott Bateman, Wengang Liu, Gordon McCalla, Jim Greer, Dragan Gaevic, Timmy Eap, Griff Richards, Khaled Hammouda, Shady Shehata, Mohamed Kamel, Fakhri Karray, and Jelena Jovanovic. Issues and directions with 174 Distributed Document Clustering and Cluster Summarization in P2P Environments educational metadata. In 3rd Annual Scientific Conference of the LORNET Research Network (I2LOR 2006), Montreal, Canada, November 2006.

[14] Christopher Brooks, Mike Winters, Jim Greer, Gordon McCalla, James C. Lester, Rosa Maria Vicari, and Fabio Paraguau. The massive user modelling system (mums). In International Conference on Intelligent Tutoring Systems (ITS 2004), volume 3220 of LNCS, pages 635–645. Springer, 2004.

[15] Soumen Chakrabarti. Mining the Web: Discovering Knowledge from Hypertext Data. Morgan Kaufmann Publishers, 2003.

[16] K. Cios, W. Pedrycs, and R. Swiniarski. Data Mining Methods for Knowledge Discovery. Kluwer Academic Publishers, Boston, 1998.

[17] C. Clifton, R. Cooley, and J. Rennie. TopCat: data mining for topic identification in a text corpus. IEEE Transactions on Knowledge and Data Engineering, 16(8):949–964, August 2004.

[18] W. W. Cohen. Learning to classify English text with ILP methods. In Proceedings of the 5th International Workshop on Inductive Logic Programming, pages 3–24.Department of Computer Science, Katholieke Universiteit Leuven, 1995.

[19] R. Cooley, B. Mobasher, and J. Srivastava. Web mining: information and pattern discovery on the World Wide Web. In Proceedings of the Ninth International Conference on Tools with Artifical Intelligence, pages 558–567, 1997.

[20] Mark Craven, Dan DiPasquo, Dayne Freitag, Andrew K. McCallum, Tom M.Mitchell, Kamal Nigam, and Se´an Slattery. Learning to extract symbolic knowledge from the World Wide Web. In Proceedings of AAAI-98, 15th Conference of the American Association for Artificial Intelligence, pages 509–516, Madison, US, 1998.

[21] J.F. Superby, J-P. Vandamme, and N. Meskens. Determination of factors influencing the achievement of the first-year university students using data mining methods. In Proceedings of the Workshop on Educational Data Mining at the 8th International Conference on Intelligent Tutoring Systems (ITS 2006), 37–44.

[22] A.-H. Tan. Text mining: The state of the art and the challenges. In Pacific Asia Conference on Knowledge Discovery and Data Mining PAKDD'99 workshop on Knowledge Discovery from Advanced Databases, pages 65–70, 1999.

[23] Tom Payne Titus Winters, Christian Shelton and Guobiao Mei. Topic extraction from item-level grades. In Workshop on Educational Data Mining, 20th National Conference on Artificial Intelligence (AAAI 2005), pages 7–14, Pittsburgh, PA, 2005.

[24] Peter D. Turney. Learning algorithms for keyphrase extraction. Information Retrieval, 2(4):303–336, 2000.

[25] R. Weiss, B. Velez, M. A. Sheldon, C. Namprempre, P. Szilagyi, A. Duda,and D. K.

Gifford. Hypursuit: a hierarchical network search engine that exploits content-link hypertext clustering. In Hypertext'96: The 7th ACM Conference on Hypertext, pages 180–193, 1996.

[26] S. M. Weiss, C. Apt´e, F. Damerau, D. E. Johnson, F. J. Oles, T. Goetz, and T. Hampp. Maximizing text-mining performance. IEEE Intelligent Systems, 14(4):63–69, 1999.

[27] Ian H. Witten and Eibe Frank. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco, 2nd edition, 2005.