IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

Effective Phrase-Based Document Indexing for Web Document Clustering

Mr. Patil A.A[1], Prof. Naidu G.J[2]

Student, Dept of Computer Science &Engineering, ST.MARY's Group of Institution, Hyderabad, India[1]

Prof., Dept of Computer Science &Engineering, ST.MARY's Group of Institution, Hyderabad, India[2]

Abstract: Record grouping procedures generally depend on single term investigation of the report information set, for example, the Vector Space Model. To accomplish more precise record bunching, more educational components including phrases and their weights are especially imperative in such situations. Archive grouping is especially valuable in numerous applications, for example, programmed arrangement of archives, gathering web search tool results, constructing a scientific classification of reports, and others. This paper presents two key parts of effective record grouping. The initial segment is a novel expression based record file demonstrate, the Archive File Chart, which takes into consideration incremental development of an expression based file of the archive set with an accentuation on productivity, as opposed to depending on single-term lists as it were. It gives proficient expression coordinating that is utilized to judge the closeness between archives. The model is adaptable in that it could return to a minimal representation of the vector space model on the off chance that we pick not to record phrases. The second part is an incremental record grouping calculation in light of amplifying the snugness of bunches via precisely watching the pair-wise record comparability appropriation inside bunches. The mix of these two segments makes a hidden model for strong and precise report likeness figuring that prompts quite enhanced results in Web record grouping over customary techniques.

Keywords: Web mining, record similitude, phrase-based indexing, report grouping, archive structure, record list chart, phrase coordinating.

1. INTRODUCTION

With an end goal to stay aware of the enormous development of the Internet, numerous exploration ventures were focused on on the most proficient method to sort out such data in a way that will make it less demanding for the end clients to discover the data they need productively and precisely. Data on the Web is present as content archives (arranged in HTML), what's more, that is the reason numerous Web record preparing frameworks are established in content information mining strategies.

Content mining offers numerous ideas with conventional information mining strategies. Information mining incorporates numerous systems that can uncover innate structure in the hidden information. One of these methods is bunching. At the point when connected to printed information, grouping strategies attempt to recognize inalienable groupings of the content records so that an arrangement of bunches is created in which groups display high intra cluster comparability and low intercluster closeness [1]. For the most part talking, content record grouping strategies endeavor to isolate the records into gatherings where every gathering speaks to some point that is not the same as those themes spoken to by alternate gatherings [2]. By applying content mining in the Web area, the procedure gets to be what is known as Web mining. There are three sorts of Web mining when all is said in done, by and Blockeel [3]: 1) Web structure mining, 2) Web utilization mining, and 3) Web content mining. We are chiefly keen on the last sort.

Uses of record bunching include: grouping of recovered records to introduce sorted out and reasonable results to the client (e.g., [4]), grouping archives in a gathering (e.g., advanced libraries), mechanized (or semi automated) making of record scientific classifications (e.g., Yahoo and Open Directory styles), and proficient data recovery by concentrating on applicable subsets (groups) instead of entire accumulations.

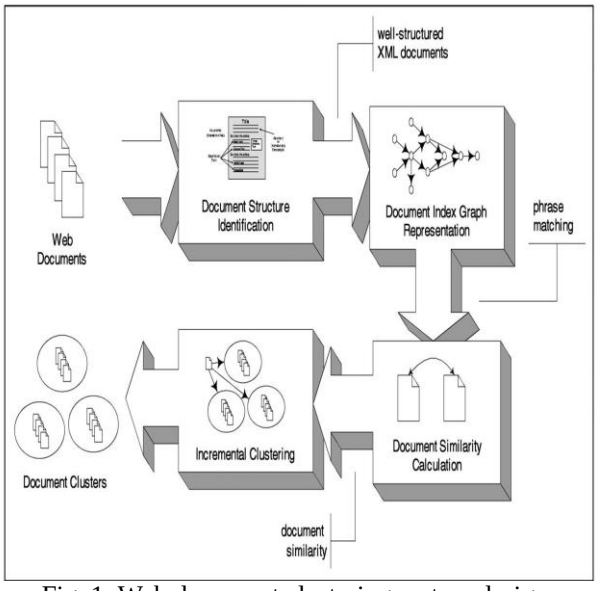

Fig. 1. Web document clustering system design

DOI 10.17148/IJARCCE.2016.55184

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

2. WEB DOCUMENT STRUCTURE ANALYSIS

Web records are known not semi structured. HTML tags are utilized to assign distinctive parts of the report. Be that as it may, following the HTML dialect is implied for indicating the format of the report, it is utilized to show the archive to the client in an agreeable way, as opposed to indicate the structure of the information in the record, subsequently they are semi structured. Be that as it may, it is still conceivable to distinguish key parts of the archive in view of this structure. The thought is that some parts of the archive are more enlightening than other parts, accordingly having diverse levels of importance in view of where they show up in the archive and the labels that encompass them. It is less educating to treat the title of the archive, for instance, and the body message similarly.
The proposed framework investigates the HTML record and rebuilds the report as per a foreordained structure that relegates distinctive levels of noteworthiness to diverse record parts. The outcome is an all around organized XML report that compares to the first HTML report, yet with the noteworthiness levels doled out to the diverse parts of the first record.

Right now, we allocate one of three levels of importance to the diverse parts; HIGH, MEDIUM, and LOW. Case of HIGH criticalness parts are the title, Meta catchphrases, Meta depiction, what's more, segment headings. Case of MEDIUM criticalness parts are content that show up in striking, italics, hued, hyper-connected content, picture interchange content, and table inscriptions. LOW importance parts are generally involved the record body message that was not allocated any of alternate levels.
A formal model is exhibited here that speaks to record highlights as sentences instead of individual words. The model accepts that the constituents of an archive are a set of sentences, which thus are made out of an arrangement of terms.
A record is spoken to as a vector of sentences:

di ¼ fsij : j ¼ 1; . . . ; pig;
sij ¼ ftijk : k ¼ 1; . . . ; lij; wijg;
Where
di: is archive i,
sij: is sentence j in archive i,
pi: is the quantity of sentences in archive i,
tijk: is term k of sentence sij,
lij: is the length of sentence sij, and
wij: is the weight connected with sentence sij.

The above definition is an immediate mapping of the genuine archive to a formal representation that breaks a record into an arrangement of sentences. Sentence weights are allocated as indicated by their noteworthiness, as talked about in Section 2. This definition does not consider the recurrence of sentences (or a portion of sentences) as a sentence weight. Sentence recurrence will be mulled over when coordinating expressions between archives. The justification for conceding sentence recurrence estimation was to perform a sluggish calculation after coordinating part of a sentence with different archives, instead of registering all conceivable frequencies forthright that won't not be utilized as a part of likeness figuring later.

3. DOCUMENT INDEX GRAPH

To accomplish better grouping comes about, the information demonstrate that underlies the bunching technique should precisely catch the remarkable elements of the information. As per the Vector Space Model, the archive information is spoken to as an element vector of terms with various weights allotted to the terms agreeing to their recurrence of appearance in the archive. It doesn't speak to any connection between the words, so sentences are separated into their individual segments with no representation of the sentence structure.
The proposed Document Index Graph (DIG for short) files the reports while keeping up the sentence structure in the first reports. This permits us to make utilization of more educational expression coordinating as opposed to singular words coordinating. Besides, DIG likewise catches the diverse levels of essentialness of the first sentences, hence permitting us to make utilization of sentence essentialness. Postfix trees are the nearest structure to the proposed model, however they experience the ill effects of enormous excess. Apostolic gives more than 40 references on addition trees, and manber also, Myers include later ones. Be that as it may, the proposed DIG model is not only an expansion or an improvement of postfix trees; it takes an alternate point of view of how to match states proficiently, without the requirement for putting away excess data.

3.1 DIG Detailed Structure

This area gives points of interest of the expression indexing structure to serve as a kind of perspective for execution purposes. Phrase indexing data is put away in the diagram hubs themselves as report tables. Fig. 2 outlines

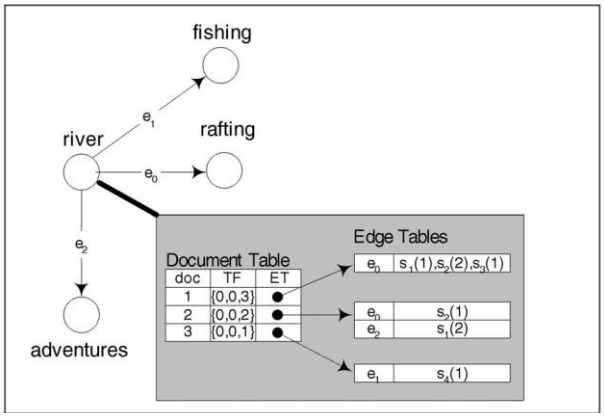

Fig. 2 DIG structure detail

the data put away in one of the hubs given in the past illustration. Essentially, the structure kept up in every hub is a table of archives. Every archive section in the archive table records the term recurrence (TF) of the word in that record. Since words can show up in various parts of a record with various levels of noteworthiness, the recorded

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

term recurrence is really broken into those levels of noteworthiness, with a recurrence check for each level (these are the three numbers under the TF section.) This structure helps in accomplishing a more precise comparability measure based on the level of criticalness.
Since the diagram is coordinated, every hub keeps up a rundown of active edges per archive section. This rundown of edges tells us which sentence proceeds with which edge. The assignment of making a sentence way in the chart is in this way to record the essential data in this edge table to mirror the structure of the sentences.

3.2 DIG Construction

The DIG is constructed incrementally by preparing one archive at a period. At the point when another report is presented, it is examined in consecutive form, and the diagram is upgraded with the new sentence data as essential. New words are added to the chart as fundamental and associated with different hubs to mirror the sentence structure. The chart building process turns out to be less memory requesting when no new words are presented by another record (or not very many new words are presented). Now, the chart turns out to be more steady, furthermore, the main operation required is to upgrade the sentence structure in the diagram to suit the new sentences presented. It is extremely basic to note that presenting another report will just require the review (or expansion) of those words that show up in that report and not each hub in the diagram. This is the place the proficiency of the model comes from. Alongside indexing the sentence structure, the level of centrality of every sentence is likewise recorded in the chart. This permits us to review such data when we measure the similitude with different records.

3.3 DIG Construction and Phrase Matching

Calculation after presenting another archive, finding coordinating phrases from beforehand seen archives turns into a simple errand utilizing DIG. Calculation 1 depicts the procedure of both incremental diagram building and expression coordinating. Rather than building archive sub graphs and meeting them with the total DIG, the calculation incrementally joins new archives into DIG while gathering coordinating expressions from past reports in the meantime.
The system begins with another report to prepare (line 1). Coordinating expressions from past archives are done by keeping a rundown M that holds a section for each past archive that imparts an expression to the current archive di. For every sentence (for circle at line 3), we process the words in the sentence consecutively, including new words (as new hubs) to the chart and developing a way in the diagram (by including new edges if fundamental) to speak to the sentence we are preparing. As we proceed with the sentence way, we redesign M by including new coordinating expressions and their individual record identifiers, and developing expression matches from the past cycle (lines 14 to 16). We first counsel the report table of vk1 for records that have sentences that proceed with the edge ek. Those records offer at minimum two terms with the present sentence under thought. We inspect the rundown M for any past coordinating phrases (from past cycles) to broaden the current two-term phrase match (anxious ek). This permits the augmentation of past matches, and can proceed for any length phrase match. On the off chance that there are no coordinating expressions at some point, we simply upgrade the separate hubs of the chart to mirror the new sentence way (line 19). After the entire record is prepared, M will contain all the coordinating expressions between the present record and any past record that mutual no less than one expression with the new record. At long last, we upgrade Gi to be the current combined DIG and yield M as the rundown of reports with all the fundamental data about the coordinating expressions, which will be utilized as a part of similitude computation later.

3.4 DIG Complexity

The case displayed here is a basic one. Genuine Web archives will contain many words. With an extremely substantial report set, the diagram could turn out to be more perplexing as far as memory use. By definition, the quantity of diagram hubs will be precisely the same as the quantity of novel words in the information set. In most pessimistic scenario, the quantity of edges will be m2 (m is the quantity of novel words) assuming each word is trailed by each other word in the corpus. Nonetheless, commonly, the quantity of edges is around one request of greatness bigger than the quantity of hubs. In
terms of memory utilization contrasted with the vector space model, on the off chance that we accept that we don't look after expression indexing structures, the model will utilize memory as expansive as
the quantity of nonempty sections in a term-by-report vector space model grid (since it speaks to a transformed rundown of term-to-record term frequencies.) If we keep up phrase indexing structures, we require additional memory as huge as the quantity of records times the normal terms per record.

4. A PHRASE-BASED SIMILARITY MEASURE

As said before, expressions pass on neighborhood setting data, which is key in deciding a precise closeness between archives. Toward this end, we formulated a likeness measure in light of coordinating expressions as opposed to individual terms. This measure misuses the data removed from the past expression coordinating calculation to better judge the closeness between the archives. This is identified with the work of Isaacs and Aslam who utilized a couple shrewd probabilistic archive closeness measure in view of Information Theory.
In spite of the fact that, they demonstrated it could enhance conventional comparability measures, yet it is still in a general sense in view of the vector space model representation. The expression likeness between two reports is computed taking into account the rundown of coordinating expressions between the two archives. From a data theoretic purpose of view, the likeness between two articles is viewed as how much they partake in like

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

manner. The cosine and the Jaccard measures are in fact of such nature, however they are basically utilized as single-term based likeness measures. Lin [29] gave a formal definition for any data theoretic similitude measure as:

sim (X,Y)= $X \cap Y / X \cup Y$

The essential suspicion here is that the comparability between two records depends on the proportion of the amount they cover to their union, all regarding phrases. This definition still concurs with the significant supposition of the cosine and the Jaccard measures, and to Lin's definition as well. This expression based closeness measure is an element of four components: the quantity of coordinating expressions P, the lengths of the coordinating expressions.

4.1 Combining Single-Term and Phrase Similarities

On the off chance that the likeness between records is construct exclusively in light of coordinating expressions, and not single-terms in the meantime, related records could be judged as non similar in the event that they do not share enough expressions (an ordinary case.) Shared expressions give vital nearby connection coordinating, however here and there similitude in light of expressions just is not adequate. To lighten this issue, and to deliver top notch groups, we joined single-term comparability measure with our phrase based similitude measure. Exploratory results to legitimize this case are given in Section 6.2. We utilized the cosine connection closeness measure, with TF-IDF (Term Frequency– Converse Document Frequency) term weights, as the single term closeness measure. The cosine measure was picked due to its wide use in the report bunching writing, and since it is depicted as having the capacity to catch human order conduct well. The TF-IDF weighting is likewise a generally utilized term weighting plan .

5. INCREMENTAL DOCUMENT CLUSTERING

In this segment, we exhibit a brief diagram of incremental grouping calculations and present the proposed calculation, in view of pair-wise record closeness, and utilize it as a major aspect of the entire Web record bunching framework. The part of an archive likeness measure is to give judgment on the closeness of archives to each other. Nonetheless, it is up to the bunching technique how to make use of such similitude count. Steinbach et al. [32] give a great examination of report bunching procedures.

An expansive exhibit of information bunching techniques can be additionally found in [33], [34]. Charikar et al. talked about an incremental progressive grouping [35] too. Beil et al. [36] proposed a grouping calculation taking into account regular terms that address the high dimensionality issue of content information sets. Pantel also, Lin [37] proposed the CBC report grouping calculation that discovers group delegates as an approach to choose the participation of groups later. The thought here is to utilize an incremental grouping strategy that will misuse our closeness measure to create groups of top notch (evaluating nature of bunching is portrayed in Section 6).

Incremental grouping is a crucial system for online applications, where time is a basic variable for ease of use. Incremental grouping calculations work by handling information objects each one in turn, incrementally doling out information items to their separate groups while they advance. The procedure is sufficiently straightforward, however confronts a few difficulties. The most effective method to decide to which group the following article ought to be allotted? How to manage the issue of insertion request?

Once an item has been appointed to a bunch, ought to its task to the group be solidified or is it permitted to be reassigned to different bunches later on? As a rule, a heuristic technique is utilized to manage the above difficulties. A "great" incremental bunching calculation needs to discover the individual group for each recently presented object without fundamentally giving up the precision of grouping because of insertion arrange or settled article to-group task. We will quickly examine four incremental bunching techniques in the light of the above difficulties, before we present our proposed technique.

5.1 SHC Complexity Analysis

By definition, the time intricacy of the comparability histogram-based bunching calculation is Oðn2þ, since for each new report we should register its closeness to all already seen reports. This is a property of all calculations that work in view of a report likeness network. Notwithstanding, the likeness histogram representation gives us favorable position with regular archive corpora. Commonly, a report likeness vector (containing its likeness to each other report) will be adequately meager. This is generally in light of the fact that a significant huge rate of archives don't share any words (after expulsion of stopwords), particularly archives from various classes, so their closeness is zero. We can exploit this scanty vector by compacting zero components into the primary container of the similitude histogram of every bunch, and just process nonzero components which really influence the calculation time of the calculation. This technique spares calculation and makes the calculation subquadratic in run of the mill circumstances.

5.2 Dealing with Insertion Order Problems

Our system for the insertion request issue is to execute a report reassignment methodology. This methodology does not totally take out the issue, but rather it diminishes its impact, i.e., the procedure is nondeterministic and distinctive insertion requesting will bring about various apportioning of the records.

More established records that were included before new groups were made ought to have the opportunity to be reassigned to recently made groups. Just records that appear to be "terrible" for a specific group are labeled and considered for reassignment to different groups. The reports that are contender to leave a bunch are the records that in the event that they were expelled from the group, the bunch likeness histogram proportion will increment, i.e., the group is in an ideal situation without them. We keep with every report a quality demonstrating the

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

histogram proportion if the report was not in the bunch. On the off chance that this quality is more prominent than the present histogram proportion, then the archive is a possibility for leaving the bunch. This "labeling" of awful reports permits bunches to be reassessed occasionally in order to evacuate those reports. A terrible archive will be expelled from a bunch if, and just on the off chance that, we can discover one or more different bunches that can acknowledge the archive. By "acknowledge," we mean the report will either build their histogram proportion or diminishing by close to ". Along these lines, profiting the underlying and the beneficiary bunches. This system makes a dynamic transaction plan between bunches for report task. It likewise permits for covering bunches, and element incremental report grouping.

6. EXPERIMENTAL RESULTS

With a specific end goal to test the viability of the Web bunching framework, we led an arrangement of examinations utilizing our proposed information model, phrase coordinating, likeness measure, also, incremental grouping strategy. The analyses led were separated into two sets. We initially tried the adequacy of the DIG model, introduced in Section 3, and the going with expression coordinating calculation for computing the closeness between reports in view of expressions versus singular words as it were. The second arrangement of analyses was to assess the exactness of the incremental archive bunching calculation, exhibited in Section 5, in view of the bunch cohesiveness measure utilizing likeness histograms.

6.1 Experimental Setup

The accessibility of Web record information sets reasonable for grouping is constrained. Notwithstanding, we utilized three information sets, two of which are Web archive information sets, and the third is a gathering of articles posted on different USENET newsgroups. Table 2 portrays the information sets. The principal information set (DS1) is a gathering of 314 Web reports physically gathered and named from different University of Waterloo also, Canadian Web sites.2 this information set was utilized as a part of [42]. It is ordered physically taking into account point portrayal. This information set has a moderate level of cover between the diverse classes. The second information set (DS2) is an accumulation of 2,340 Reuters news articles posted on Yahoo! news, and was utilized by Boley as a part of [43], [44], [45]. The classifications of the information set originate from the Yahoo classifications of Reuter's news nourish. The cover between classes is entirely low in this information set. The third information set is a subset of the full 20-newsgroups accumulation of USENET news bunch articles. This information set is accessible from the UCI KDD Archive.3 Each news bunch constitutes an alternate classification, with fluctuating cover between them; some news gatherings are extremely related (e.g., talk.politics.mideast and talk.politics.misc) and others are most certainly not related by any means (e.g., comp. Graphics and talk.religion.misc.).

6.2 Effect of Phrase-Based Similarity on Clustering Quality

The similitude ascertained by our calculation was utilized to develop a similitude framework between the archives. We chose to utilize three standard archive bunching procedures for testing the impact of expression likeness on bunching [33]:

1. Various leveled Agglomerative Clustering (HAC)
2. Single Pass Clustering, and
3. K-Nearest Neighbor Clustering (K-NN).

For each of the calculations, we built the similitude network and let the calculation bunch the records taking into account the displayed closeness lattice.

The outcomes recorded in Table 3 demonstrate the change in the grouping quality on the primary information set utilizing the joined closeness measure. The changes demonstrated were accomplished at a closeness mix element somewhere around 70 and 80 percent (phrase comparability weight). The parameters decided for the diverse calculations were the ones that delivered best results. The rate of change ranges from 19.5 to 60.6 percent expansion in the F-measure quality, and 9.1 to 46.2 percent drop in Entropy (lower is better for Entropy). Clearly the expression based comparability assumes an imperative part in precisely judging the connection between records. It is realized that Single Pass grouping is exceptionally touchy to commotion; that is the reason it has the most noticeably bad execution. Be that as it may, when the expression likeness was presented, the nature of groups created was pushed near that created by HAC and K-NN.

7. CONCLUSIONS AND FUTURE RESEARCH

We introduced a framework made out of four segments in an endeavor to enhance the record bunching issue in the Web space. Data in Web archives does not lie in the substance just, however in their innate semi structure. We displayed a Web report investigation segment that is equipped for distinguishing the weights of different Web reports expressions and separating the report into its sentence constituents for further preparing.

The second part, and maybe the most critical one that has the majority of the effect on execution, is the new record model presented in this paper, the Document Index Graph. This model depends on indexing Web records utilizing phrases and their levels of criticalness. Such a model empowers us to perform phrase coordinating and similitude count between records in an extremely strong, proficient, and exact way. The nature of grouping accomplished utilizing this model altogether surpasses the customary vector space model based methodologies.

The third segment is the expression based comparability measure. Via precisely looking at the components influencing the level of cover between reports, we contrived a phrase-based similitude measure that is fit for precise count of pair-wise archive likeness.

The fourth part is an incremental report bunching strategy in view of keeping up high group cohesiveness by

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 5, Issue 5, May 2016

enhancing the pair-wise report likeness dispersion inside every group.

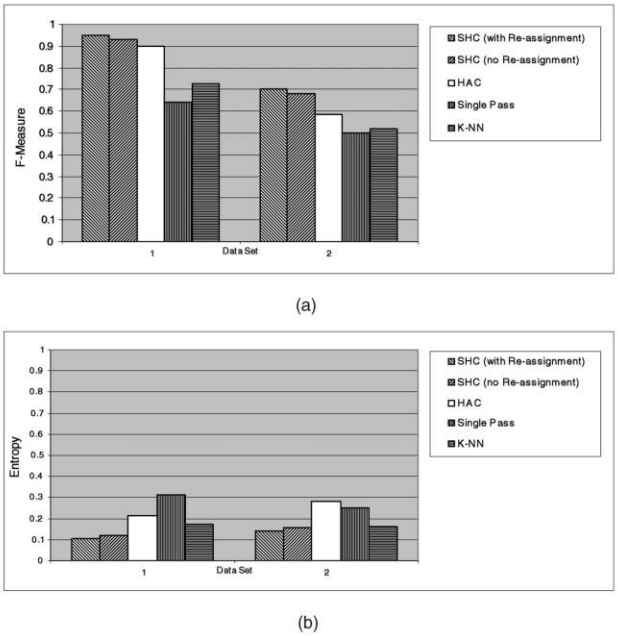

Fig. 3. Quality of clustering comparison. (a) Clustering quality—F-measure. (b) Clustering quality entropy.

There are various future exploration bearings to expand and enhance this work. One bearing that this work might proceed on is to enhance the precision of comparability figuring between reports by utilizing distinctive comparability count techniques. In spite of the fact that the current plan demonstrated more precise than conventional techniques, there is still opportunity to get better. In spite of the fact that the work displayed here is gone for Web archive bunching, it could be effortlessly adjusted to any archive sort too. Be that as it may, it won't profit by the semi structure found in Web archives. We will probably explore the utilization of such model on standard corpora and see its impact on bunching contrasted with conventional strategies.

ACKNOWLEDGEMENT

This work has been mostly bolstered by a key stipend from the Natural Sciences and Engineering Research Committee of Canada (NSERC).

REFERENCES

[1] K. Cios, W. Pedrycs, and R. Swiniarski, Data Mining Methods for Knowledge Discovery. Boston: Kluwer Academic Publishers, 1998.
[2] W.B. Frakes and R. Baeza-Yates, Information Retrieval: Data Structures and Algorithms. Englewood Cliffs, N.J.: Prentice Hall,1992.
[3] R. Kosala and H. Blockeel, "Web Mining Research: A Survey,"ACM SIGKDD Explorations Newsletter, vol. 2, no. 1, pp. 1-15, 2000.
[4] O. Zamir and O. Etzioni, "Grouper: A Dynamic Clustering Interface to Web Search Results," Computer Networks, vol. 31,nos. 11-16, pp. 1361-1374, 1999.
[5] S. Dumais, J. Platt, D. Heckerman, and M. Sahami, "Inductive Learning Algorithms and Representations for Text Categorization," Proc. Seventh Int'l Conf. Information and Knowledge Management,pp. 148-15, Nov. 1998.
[6] H. Kargupta, I. Hamzaoglu, and B. Stafford, "Distributed Data Mining Using an Agent Based Architecture," Proc. Knowledge Discovery and Data Mining, pp. 211-214, 1997.
[7] U.Y. Nahm and R.J. Mooney, "A Mutually Beneficial Integration of Data Mining and Information Extraction," Proc. 17th Nat'l Conf. Artificial Intelligence (AAAI-00), pp. 627-632, 2000.
[8] Y. Yang, J. Carbonell, R. Brown, T. Pierce, B. Archibald, and X. Liu, "Learning Approaches for Detecting and Tracking News Events," IEEE Intelligent Systems, vol. 14, no. 4, pp. 32-43, 1999.
[9] D. Freitag and A. McCallum, "Information Extraction with HMMs and Shrinkage," Proc. AAAI-99 Workshop Machine Learning for Information Extraction, pp. 31-36, 1999.
[10] T. Hofmann, "The Cluster-Abstraction Model: Unsupervised Learning of Topic Hierarchies from Text Data," Proc. 16th Int'l Joint Conf. Artificial Intelligence (IJCAI-99), pp. 682-687, 1999.
[11] T. Honkela, S. Kaski, K. Lagus, and T. Kohonen, "WEBSOM—Self- Organizing Maps of Document Collections," Proc. WSOM '97, Workshop Self-Organizing Maps, pp. 310-315, June 1997.
[12] W.W. Cohen, "Learning to Classify English Text with ILP Methods," Proc. Fifth Int'l Workshop Inductive Logic Programming, pp. 3-24, 1995.
[13] M. Junker, M. Sintek, and M. Rinck, "Learning for Text Categorization and Information Extraction with ILP," Proc. First Workshop Learning Language in Logic, J. Cussens, ed., pp. 84-93, 1999.
[14] S. Scott and S. Matwin, "Feature Engineering for Text Classification," Proc. 16th Int'l Conf. Machine Learning (ICML-99), pp. 379- 388, 1999.
[15] S. Soderland, "Learning Information Extraction Rules for Semi-Structured and Free Text," Machine Learning, vol. 34, nos. 1-3, pp. 233-272, 1999.