Journal of Computer Sciences

Original Research Paper

A Novel Approach for Discovering the Patterns by using PDBD Model in Big Data

Kamatchi Sundravadivelu and Muthuraman Thangaraj

Department of Computer Science, School of Information Technology, Madurai Kamaraj University, Madurai, India

Article history
Received: 14-11-2021
Revised: 01-04-2022
Accepted: 14-04-2022

Corresponding Author:
Kamatchi Sundravadivelu
Department of Computer Science, School of Information Technology, Madurai Kamaraj University, Madurai, India
Email: svadiveluk2021@gmail.com

Abstract: Big data is a term that refers to information collected from a wide range of sources, including transaction processing systems, sensors, digital photographs, internet click stream logs, movies, and social media. The goal of text mining is to reveal new patterns and relationships that could lead to the discovery of previously unknown sources. It provides ample scope for interpretation of genuine user intent or contextual meanings as the case be. This research focuses on the creation of a novel approach for discovering patterns in big data using the Pattern Discovery in Big Data (PDBD) model in text mining. From textual sources, text-mining techniques extract a range of patterns and important facts. Sequential pattern mining helps to extract informative elements from a set of sequences based on the frequency of occurrences. The Pattern Discovery in Big Data (PDBD) model study has four primary contributions to uncovering patterns uniquely and effectively: Semantic information extraction, pattern improvements, LDA Topic Modeling, and cluster assignments. The proposed work compared the Pattern Taxonomy Model (Inner Pattern Evolving) pattern inner evolving model with other existing similar benchmark models to evaluate the model performance. The empirical results prove that the proposed model outperformed the compared models with significant improvement in accuracy Improvement in this Pattern Discovery in Big Data (PDBD) model yields better results in discovering the patterns.

Keywords: Pattern Mining, Semantic Information and Extraction, Pattern Implementation, Support Value, LDA Topic Modeling

Introduction

Text mining can be defined as the process of extracting interpretable information from natural language texts. The raw text comes in many forms such as messages, emails, tweets, documents, articles, blogs, log entries, etc. The information extraction process is carried out with techniques from other disciplines of data science, such as Natural language Processing/Understanding, text classification, clustering, pattern mining, information retrieval, etc. (Li *et al*., 2015). The focus of the proposed study is based on the notion that pattern mining techniques when applied with raw text data help to uncover useful information, which further supports intelligent decision making (Sze-To and Wong, 2018). The role of pattern mining is largely gaining momentum with the advent of big data. Big data condition occurs when the nature of data is beyond the compute capacity of the existing frameworks in an environment. In other words, the traditional processing methods fail to handle the huge volume, variety, and veracity of data. There are many ways to model such large data to extract useful information as said above. Pattern mining is one such field that has techniques to identify repeating patterns in raw text such that they collectively convey the underlying context. Text patterns are also known as informative subsequences of large sequential data (Thangaraj and Sundravadivelu, 2020). The work focuses on developing an effective algorithm for discovering patterns from large documents (Wan *et al*., 2019; Zhong *et al*., 2010). The considerable issues related to information extraction, mining, and analytics must be researched because the growth rate of unstructured data is relatively large and has been increasing in recent times. The key hurdles of extracting valuable information appear to be scalability, dimensionality, and heterogeneity of unstructured data. The key questions are how to convert unstructured data into a structured format for better representation. To improve analytics and decision-making, efficient and accurate transformation of unstructured data to structured data is

required. Automatic pattern mining from unstructured data is a field that looks for innovative ways to extract semantics (Palaniammal, 2021) and contextual information from data by analyzing and recognizing patterns (Thangaraj and Vijayalakshmi, 2013).

Pattern mining is a data mining task that focuses on analyzing data correlations and identifying useful patterns from vast datasets. The objective of pattern mining is to devise a strategy for removing significant patterns from a data set that is both proficient also, successful. Ontology coordinating, process mining, direction, and requirement programming are just a few of the applications and areas where it's applied. When working with huge and very large datasets, solving pattern mining problems takes a long time. Many optimizations and high-throughput computing techniques have been presented to increase the runtime performance of pattern mining systems. When dealing with natural language documents, however, these solutions are ineffective since only several important patterns are used and shown to the end client.

When dealing with such data, it suffers from runtime and performance accuracy issues. The inclusion of co-referenced words, synonyms, polysemy terms, and other domain-specific terminology contributes to this. The overall performance of such an approach is harmed as a result of these concerns. As a result, a generic intelligent frequent pattern mining model is proposed for dealing with pattern mining challenges in natural language documents in this research paper.

Item sets, bases that show up in an informational index with a recurrence more noteworthy than a client determined edge are known as frequent patterns (Thangaraj and Sundravadivelu, 2022). A frequent itemset, for instance, is a gathering of items that show up oftentimes together in an exchange informational collection, like milk and bread. A frequent sequential pattern is an aftereffect, like purchasing a PC first, then a computerized camera, and lastly a memory card, if it happens habitually in a shopping history information base (Thangaraj and Karthikeyan, 2014). Subgraphs, subtrees, and sublattices are examples of structural forms that can be joined with item sets or subsequences to construct a substructure. Finding common patterns is critical for mining associations, correlations, and a variety of other data interactions.

It also assists with data indexing, grouping, clustering, and different data mining activities. Continuous pattern mining is a significant data mining task and a research point in data mining. When these algorithms are used in a hybrid way to combine these algorithms into machine learning data models, the models achieve excellent accuracy and efficiency. Other issues of pattern mining from large-scale unstructured data sets include conflict resolution, establishing a balance among in formativeness, representativeness, unbalanced data, and structuring the data.

Related Work

Support Vector Machine (SVM) is a set of guided learning methods (supervised learning) for classification and regression analysis that examines data and recognizes patterns. Support Vector Machine is a machine learning approach based on the Structural Risk Minimization (SRM) principle, intending to reduce the loss function to determine the optimum hyperplane that divides two classes in the input space. This is done by searching for a hyperplane or decision boundary that divides one class from another. The Support Vector Machine seeks to determine the optimum hyperplane from an infinite number of functions (Thangaraj and Sivakami, 2018).

The infinite function in hyperplane search in the Support Vector Machine method is a benefit since it allows processing to be done at any time, regardless of the data. The data-gathering stage is the initial step in the sentiment analysis process. Data can be gathered from a variety of sources, including news portals, online forums, other social media, and personal blog sites, in addition to Twitter (Thangaraj and Gayathri, 2013). The next stage is labeling when the data has been effectively collected into a dataset. The purpose of the labeling, in this case, is to divide the data into multiple sentiment classifications that will be used in the study. Training data is used to train the system to recognize the pattern being sought, whilst testing data is used to verify the outcomes of the training. The findings of each annotator's categorization are then compared and checked to confirm that the sentiment rating for the comment is appropriate.

Pattern Taxonomy Model (Inner Pattern Evolving) the terms in conventional d-patterns were focused on the papers in the training set. Because of the low-frequency problem, the approach will be effective in reducing the unfavorable impacts of loud patterns. Since it just adjusts a pattern's term supports inside the pattern, this method is called inner pattern evolution. To classify documents into appropriate or inappropriate groups, a threshold is frequently utilized. A total struggle offender, which is a subset of A, and a fractional clash offender party, who holds part of A's terms, are the two types of offenders. The following is a description of the underlying concept of bringing up-to-date patterns: Complete conflict offenders are first separated from their d-patterns.

The term supports for partial conflict offenders is modified to reduce the impact of noisy documents. The algorithm Inner Pattern Evolving performs the main procedure of inner pattern evolution. This technique takes a set of patterns as input Zhong *et al*. (2010). The result is a partial set of d-patterns.

Many valuable aspects are given by a knowledge-based system, such as pattern support and confidence, pattern relationships, pattern taxonomies distribution, and taxonomies dimension. Some components of the PTM system, such as pattern relationships and pattern support,

have been investigated. The majority of data mining algorithms, such as PTM, are computationally expensive, especially during the Pattern Deploying phase. Reducing the dimensionality of the component space in the information base is one way to increase the efficiency of a pattern taxonomy-based model. Applying length-decreasing support constraints to frequent pattern mining is an alternative method.

The ontology-aware neural network is a general framework that may be applied to a wide range of microbiome data mining problems. On the one hand, biological data is frequently organized in a hierarchical or ontological fashion; as a result, ONN is well-suited to these types of data. In gene mining, species mining, and dynamic pattern finding, on the other hand, neural network methods could yield models that always outperform traditional methods. As a result, neural network models may make pattern mining from microbiome data much easier. The ONN's strength in knowledge discovery has been demonstrated in a variety of scenarios, including novel genes, new species, and novel dynamic patterns of communities, among others. Thus, ONN might be a framework for pattern mining from microbiome data, representing a paradigm shift from standard machine learning to an ontology-aware and model-based method, which has vast application scenarios in microbiome data mining. This recursive technique considers the criterion of the highest proportion of information gain, i.e., chooses the attribute that best ranks the data (Viloria *et al*., 2019).

Pattern Algorithm - Directed Aligned Pattern Clustering (PD-APCn) The model employs a systematic approach to adaptively estimate the representation model width from data without doing an exhaustive search, as well as to discover mutational uncommon patterns including small substitutions and frame shift insertion and deletion. As a result, the final APCs created by PD-APCn are more stable and robust because they adhere to the conditions dictated by the data's more natural sequence structures and functionality (Lucchese *et al*., 2013). The results of the discovery reveal such phenomena. It, therefore, addresses the challenging challenge of defining the size of a conserved zone while avoiding an extensive search for such a size parameter to find an alternative solution. Synthetic datasets with a priori known mutant protein sequence regions were used to evaluate the performance of PD-APCn. It was proved through parameter research that PD-APCn consistently performed well across these datasets, suggesting its resilience.

Amado *et al*. (2018) while topic detection algorithms are relatively mature, domain-oriented topic detection offers a wide range of applications. It's a first to develop a strategy for detecting open-source cyber threat themes in real-time. Building on existing general topic detection technologies and security domain knowledge, Domain-oriented Topic Discovery based on Features Extraction and Topic Clustering (DTD-FETC) examines threat data from open source security news sites. The approach aims to detect both emergent danger issues and historical event continuations in real-time. This solves the problem of sparse datasets in security intelligence and word vector models can't employ good semantic information to train feature word vectors (Indra and Thangaraj, 2019).

The approach for extracting keyword features ITFIDF-LP, the subject word feature extraction method LDA-SLP, and a named entity feature extraction method was used to identify topics. Based on the Hierarchical Agglomerative Clustering algorithm of the centroid linkage method, this study proposed a centroid linkage method based on vector product similarity Zheng *et al*. (2018). It was also discovered that multi-layer topic clustering structures may be utilized to identify subjects and associated events, allowing for more accurate trend detection.

The authors (Anujna and Ushadevi, 2017), has given that, the unstructured data cannot be converted into database directly. By applying all the text mining rules and procedures and coding the unstructured data can be converted into database and thus, it will be helpful for the users to search the relevant or proper words from the text files.

An algorithm for pattern discovery which includes the process of pattern evolving and pattern deploying, to improve the effectiveness of using and updating discovered patterns for finding relevant and interesting information (Aswini and Lavanya, 2014).

PANDA+, an algorithmic framework was used to optimize different cost functions generalized into a unifying formulation (Lucchese *et al*., 2020). The goodness of the algorithm was measured by measuring the quality of the extracted patterns. An evaluation was conducted on synthetic data, where patterns were artificially embedded, and on real-world text collection, where each document is labeled with a topic. In order to qualitatively evaluate the usefulness of the discovered patterns, PANDA+ is evaluated to detect overlapping communities in a bipartite network.

This article (Palaniammal and Vijayalakshmi, 2013), implemented a system, which suggests the user all the effective details to know about an educational domain. It is reliable because though it is being inputted with synonymous words and misspelled words, it retrieves the similar result.

The system is limited to work on, inner pattern evolution and unable to identify the negative document in dataset (Pawar and Karande, 2014). The operation only considers *xml* file documents for obtaining the patterns. But the inner pattern evolution in negative document term set for identifying false positive documents is not considered.

An effective pattern discovery technique has been proposed to overcome the low-frequency and misinterpretation problems for text mining.

In many real-world situations, frequent item sets mining are advantageous. FIM, on the other hand, has several restrictions, such as the fact that purchase quantities are not taken into account and things of varying utility and importance are thought to be comparable. The high utility item sets mining approach has been designed to find high utility itemsets in datasets to overcome this restriction on FIM (Sujatha and Poonguzhali, 2018). The goal of utility mining is to uncover patterns with a high utility, where the utilities of a pattern are defined in a utility function based on the domain knowledge employed. In our daily lives, a sequence is created all over the place in the shape of an ordered list of items.

For example, DNA sequences and Web access logs, among other things. From event sequences or time-series items, Sequential Pattern Mining (SPM) discovers and analyses statistically significant sub-sequences. Market basket analysis, Web mining, bioinformatics, and other applications are all possible with SPM. Pattern mining is typically done on datasets where the existence or absence of item sets is known. However, in many real-world systems, data can be a source of uncertainty. Sampling mistakes, intrinsic imprecise measurement, outdated sources, network bandwidth constraint and latency, and limited device capacity all contribute to uncertainty.

Existential probability can be used to describe uncertainty. Data extraction from ambiguous data has become a popular research topic in recent years. There has been an increase in demand for utility-oriented pattern mining. UPM is a critical activity with numerous high-potential applications, including e-commerce, banking, and biomedicine. The majority of the algorithms, however, do not scale or manage massive amounts of data. The processing of a huge amount of data in a reasonable time with the use of huge computing resources will increase the energy consumption and consequently, this will lead to an increase in greenhouse gas emissions and environmental impacts. Authors deal with this problem and analyze the relevance of green measures and big data (Cai *et al.*, 2019). In the same context, authors (Wu *et al.*, 2018) discussed the pattern mining approaches for extracting inter-relationships among terms and in turn among concepts.

Based on the existing trends concerning pattern mining approaches, the techniques when combined with Machine learning models improve efficiency and also accommodate text entity recognition procedures. These abilities make a hybrid model combining pattern mining algorithms and ML models a better alternative for handling natural language data.

Proposed Work

The research work consists of three main data layers, the pattern layer, and the topic model layer. Each layer is made up of sub-process stages such as Pre-processing text data, Set of Documents, Pattern Discovery, Semantic Information and Extraction, Pattern Improvements, Topic modeling, and Cluster assignments.

- In pre-process stage; the data is cleaned for eliminating noisy data. Data cleaning deals with common text pre-processing techniques such as case removal, whitespace, special characters, stop word removal, stemming, and lemmatization. Besides, the words that are shorter than three letters and those that are longer than 25 characters are removed. As the presence of these words harms the accuracy of the model. In the next stage, using a sequential pattern mining algorithm, the document is converted into a paragraph and further into individual terms for identifying the word sense
- The pattern discovery stage involves frequent and closed pattern Techniques, to understand the underlying patterns in the consecutive words as sometimes the system falsely identifies the negative documents
- The semantic information and extraction stage uses d- a pattern algorithm. Patterns are organized based on the weights of terms. It then evaluates the term weights and discovers specific patterns in the set of documents
- Pattern improvements are applied to reduce false-positive assignments due to the presence of synonym words. It eliminates both noise documents and noise patterns to get positive assignments

LDA Algorithm is a topic modeling algorithm used to understand the inherent topics in the document.

- The proposed model aims to classify the documents according to the subject they describe. It is achieved using the similar other textual content, the vocabularies derived from them, and then the relationship between its terms
- The consecutive terms are grouped based on similarity and underlying relationships into multiple clusters using the Hclust algorithm.

Data Collection and Pre-Processing

The data is a group of research articles published in online databases, spanning multiple topics, such as science, language, arts, etc. The total number of records taken for the review is 36,431 articles from Google Scholar, PubMed, NCBI, Elsevier, and IEEE. The metadata about the data collection consists of Author, Date of publication, Subject, Accession number, and Publisher details. The Data is then processed for the removal of unwanted or noisy text content. Before data cleaning, the documents are reduced to

paragraphs and then to words without losing word order. This is done to retain the semantic component of the text data. Text pre-processing approaches such as removal of punctuation, duplicate, and whitespace removal are carried out. Unlike common pre-processing pipelines, in this study, the frequent words are retained except for words related to Parts of Speech, by tagging the tokens. Term weighting is used for assigning higher weights to rare words, prepositions, conjunctions, and interjection words as they assist in differentiating word sequence patterns. Apart from these, the words mentioning core topic-based words are removed as they will induce unwanted Bias in one article and ambiguity in other articles (Wan *et al*., 2019).

Pattern Discovery

Sequential Pattern Mining

This is the crucial step in the proposed work. It uses text mining techniques to undertake initial pattern mining procedures such as identifying entities that constitute a pattern, the number of patterns that represent a concept, relationship between terms and among concepts. It helps to discover the word sense, sentence relationships, semantic meanings, and ambiguity reduction [new1].

Algorithm 1: Sequential Pattern Mining
F1=the set of frequent 1-sequence
a=2,
do while Fa-1!=Null;
Generate candidate sets Ca (set of candidate sequences);
For all input sequences s in the database D
do
  Increment count of all a in Ca if s supports b
End do
Fa={b Ɛ Ca such that its frequency exceeds the threshold}
A=a+1;
End do
Result=Set of all frequent sequences is the union of all Fa's

Algorithm 1 sequential patterns denote terms that appear frequently consecutive all over the document search space. It uses both frequent and closed frequent mining routines to preserve word order and reduce the generation of sub-sequences, as it may result in redundant processing times. The enumeration strategy is based on patterns generated based on the common lexicographic order which further assists in extending the traversal for the entity of interest and full sequence attainment. This is our dataset will result in higher matching costs which could be reduced by using frequent consecutive words. This is similar to defining a-priori for various topics the algorithm will consider in the next stages. The domain knowledge to facilitate the same will be provided by the data corpus obtained from the data pre-processing stage. The text mining in this phase is primarily composed of extracting semantically similar word collections that also occur together frequently. This reduces the search space for the extraction of patterns representing various topics. The challenge of sequential pattern mining entails extracting frequent non-contiguous or contiguous sub-sequences. Because algorithms must construct separate containers for a subset of terms that appear in combination with the same kind of words and those with unique word combinations. This adds complexity to the model. Additionally, most of the retrieved sequences in sequential pattern mining are quite similar to each other and excessively redundant. However, this could be converted into semantic patterns with minimum effort in terms of identification and also solves the problem of redundant features. As redundant features indicate the semantic richness of concepts.

After this stage, only the frequent terms are considered for further processing. Every single item gets counted into the model in the first pass. Frequent items are generated from the candidate two sequence sets and another pass is required for finding their frequency. From the frequent two sequences, we can generate candidate three sequences. The steps are repeated until any more frequent sequences are not generated.

Pattern discovery extracts the patterns from the corpus where every pattern consists of a set of terms that occur together or according to some predefined criteria. Such patterns exhibit a taxonomy-oriented data structure that can be traversed in a bottom-up direction to retrieve the data itself. This taxonomical structure also signifies the semantic relationships between various terms. It provides sub-set relations through sequential patterns or words in the corpus. Which $T = \{t_1, t_2, t_3, t_4 ... t_z\}$ be a bunch of terms that are extricated from the algorithm. Let $T = \{tt_1, tt_2, tt_m\}$ be a lot of terms (or watchwords) that can be eliminated from the course of action of positive records, D+. The negative documents are represented as D- (Xuan *et al*., 2015).

Algorithm 2: Frequent Pattern Extraction:

Input: Set of document listed item

Output: Set of Frequent Patterns.

1: start

2: Documents (D) → split into paragraphs (PS1)

3: for each document

4: Calculate the absolute support (sup_a) and relative support (sup_a)

5: assign the min_support valued

6: if (sup_a, sup_a ≥min_support)

7: compute the Frequent Patterns

8: eliminate the limit patterns for relevant to patterns

10: end if

11: end if

Algorithm 2 the base support of various terms with respect to the confidence scores of each is used to filter frequent patterns. The confidence score is calculated through both direct and indirect support values denoted by (supa1) and (supr1). The value of support for terms is measured through terms that occur together in certain intervals. The patterns are identified using the formula given below, where, X denotes the terms and |x| denotes the magnitude of support for multiple terms to occur together and PS1 (D) denotes the pattern score of confidence obtained for a specific term to be part of the pattern being assessed:

$$sup_{r1}(X) = |X| / |PS1\ (D)|$$

An ordered list of terms is called a sequential pattern. $Saa = < aa_1, aa_2, aa_3 \ldots. aa_i >$ is a subsequence of another sequence $Sbb = <bb_1, bb_2, bb_3 \ldots. bb_i>$ is called Saa is a subset of Sbb if and only if it belongs to $jj_1, jj_2 \ldots jj_b$ such that $1 \leq jj_1 \leq jj_2 \leq jj_b \leq j$ and $a_1 = b_{j1} \ldots a_i; = b_{jb..}$ Any recurring patterns will have two subtypes such as super and sub-patterns. For example, *S* is the super pattern of *sbb* which has another sub pattern set called *saa*. The overall support of patterns obtained from at least two patterns is more significant as consequent patterns form regular patterns.

A frequent sequential pattern X is known as shut while perhaps no super pattern x_1 of X with the end goal that $sup_{a1}(Xl) = sup_{a1}(X)$.

Algorithm 3:

Frequent & Closed pattern mining (Db, min_sup, M)
Input: A database DB, min_sup.
Output: The complete frequent and closed pattern set M.

1. Start
2. Remove infrequent items and empty sequences, and
3. Sort each item set of a sequence in DS.
4. SS^{1} ⟶ all frequent and closed one item sequence;
5. SS ⟶ SS^{1};
6. for each frequent and sequence s Є SS^{1} do
7. Clospan(s, DS, Min_sup, M);
8. Eliminate frequent and closed sequences from M.
9. Stop

Algorithm 3 presents a frequent pattern explained as the item set, as consecutive terms that appear together with a frequency equivalent to or surpassing the user-defined threshold. The frequent pattern is converted into a covering set of documents in Fig. 2 shows the best five most successive words. It is also important to verify whether the words are semantically relevant. It has to be validated to ascertain the degree of semantic abundance. The following module is designed to achieve such patterns and eliminate other patterns that are not semantically sound.

Semantic Information and Extraction

In this stage, the aim is to obtain semantic information from the previous individual patterns. It strives to establish relationships among patterns. It creates semantic pattern subsets that contain information that in turn acts as metadata for various entities in the data. When semantic patterns are identified, it helps to label the topics appropriately. The metadata information can be further tuned to map concepts represented by the patterns. This feature helps to convert unstructured data into structured data in a form easier to interpret and extend for other tasks (Xuan *et al*., 2015).

D-Pattern Mining Algorithm

The D-pattern mining algorithm helps to identify the set of D-patterns from the available patterns. It denotes document-specific patterns. AS the domain changes the patterns change according to the document-specific terms. The patterns are archived D-patterns for the corresponding domain. The deploying process plays a significant role in finding the d-patterns and support assessment terms. Example for Sub-sequence is a sequence $Saa = Saa_1, Saa_2, Saa_3 \ldots. Saa_n$ is a subsequence of other sequence terms such as $Sbb = Sbb_1, Sbb_2, Sbb_3 \ldots. Sbb_m$, which was denoted by α, β. A set of integers such as $1 \leq ii_1 < ii_2 < \ldots < ii_n \leq m$ such that $Sa_1 = Sb_1, Sa_2 = Sb_2 \ldots\ldots Sa_n = Sb_m$.

Algorithm 4: D-Pattern Mining Algorithm

The patterns thus obtained in the positive category are converted into the required D-patterns. It is evolved in steps 6 to 9 until the optimal pattern is achieved. The optimal patterns are categorized based on the support scores achieved. The support values are summarized when there are common terms in more than two patterns.

input : positive documents D^+; minimum support, *min_sup*.
output: d-patterns DP, and supports of terms.

$DP = \emptyset$;
foreach *document* $d \in D^+$ **do**
  let $PS(d)$ be the set of paragraphs in d;
  SP = SPMining($PS(d)$, *min_sup*);
  $\widehat{d} = \emptyset$;
  foreach *pattern* $p_i \in SP$ **do**
    $p = \{(t, 1) | t \in p_i\}$;
    $\widehat{d} = \widehat{d} \oplus p$;
  end
  $DP = DP \cup \{\widehat{d}\}$;
end
$T = \{t | (t, f) \in p, p \in DP\}$;
foreach *term* $t \in T$ **do**
  $support(t) = 0$;
end
foreach *d-pattern* $p \in DP$ **do**
  foreach $(t, w) \in \beta(p)$ **do**
    $support(t) = support(t) + w$;
  end
end

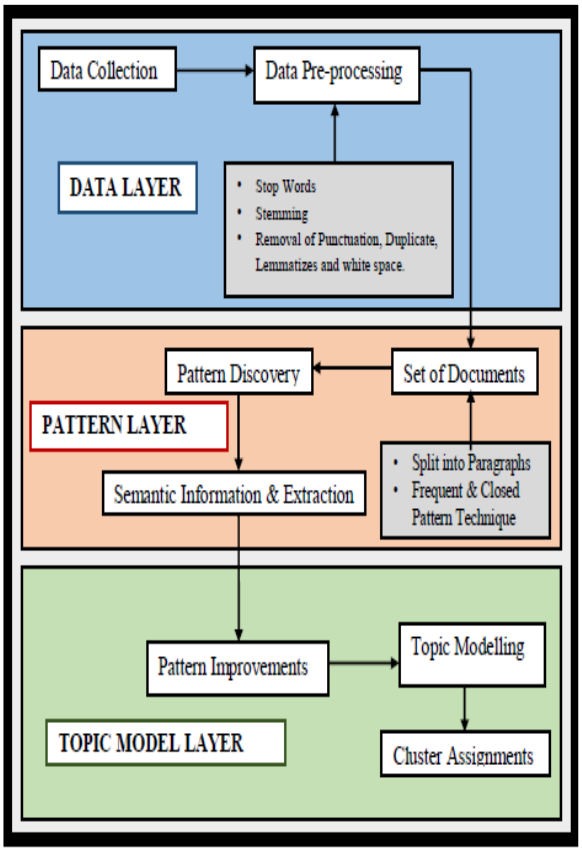

Fig. 1: Framework of Pattern Discovery in Big Data model (PDBD)

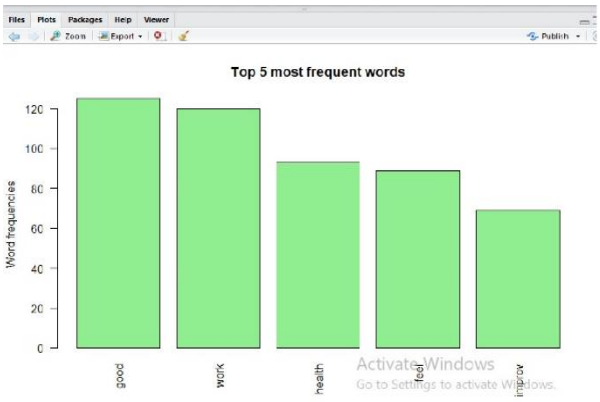

Fig. 2: Top 5 most successive words

The PDBD framework in Fig. 1 is proposed to discover the patterns from big data applications. It is a three layer architecture, which has data layer, pattern layer and topic model layer. The data layer is used for data acquisition and preprocessing of data. The pattern layer find semantic information and discover the pattern. Third layer, Topic model layer deals with topic modeling and cluster assignment.

Algorithm 4, (D- Pattern mining) reveals describes the training process which is used to find the set of d-patterns. The fundamental objective of the deploying system is to figure out the d-patterns which are available in text documents and also for finding word support assessment. In these words, supports are determined based on all the words present in the d-patterns. A set of words or keywords represented as $td = \{td_1, td_2 \ldots td_n\}$ and a sequence is represented as $ss = ss_1, ss_2 \ldots ss_n$ where ssi belongs to the text documents as td. The d-pattern was effectively discovered and term support evaluations are the main task for finding deploying process. In Algorithm 4, a positive document consists of d-patterns that discovered patterns in a positive report. These are represented as a bunch of d-designs. These d-designs are registered for all terms in d-designs which are dependent on ordinary structures. Haq *et al*. (2019) the deployed pattern is discovered patterns effectively and they are grouped up. The d-pattern algorithms are used for finding all patterns in certain documents that are gathered. Patterns are organized in a specific format it evaluates term weights and discovered specific patterns. The d-pattern terms are processed by upholds. The evaluation of Term support is determined by the weight of the term. The semantically rich patterns are extracted in this manner.

Pattern Improvements

Inner Pattern Evolving: The IPE procedure is utilized to decrease the results of noise patterns. PTM Algorithm is used for finding d-patterns in the positive documents (D+) which are based on minimum support and d-patterns are deployed to find out the term support Haq *et al*. (2019). In the test phase, it calculates the weights for all incoming documents it can be sorted based on weights. To discover the patterns are much more specific than all documents. Now and then, the framework falsely lessened negative documents as positive documents and diminish the noise documents.

The core idea of the Frequent Pattern Growth algorithm (Min *et al*., 2020) is denoted by item sets that appear together frequently. And Apriori Property is denoted by a subset of frequent items. Join Operation: To observe Lk, a bunch of up-and-comers k- item sets is created by joining Lk-1 with itself. The main benefit of the Frequent Pattern tree is that the algorithm checks the tree just two times. Aside from this significant benefit, the others incorporate completeness. The length of the frequent pattern tree is determined by the number of the events of frequent items it contains. The three levels are determined by the highest number of items in each pass of the algorithm. The tree is traversed from beginning to end based on the header table. The pattern depends on the recurring terms in the tree. For each pattern base, a frequent pattern tree is traversed using the conditions specified by the user or by the learning functions. This stage provides continuous patterns and a single pattern if the tree contains solitary patterns for the given conditions (Yao *et al*., 2016).

The Improved IPE will be applicable to the low-frequency patterns of text mining for the clients. The consideration of further developed internal pattern evolving has been demonstrated to expand accuracy that is the undesirable impacts of regular item-set mining methods have been disposed of significantly by the proposed framework.

Algorithm 5: FP Growth (*D', D', DP, μ*)
Input: The transaction database DBase, Minimum Support min_sup
Output: Frequent itemsets to construct corresponding Frequent Pattern tree, Frequent Pattern-array; Ahead-mining (Frequent Pattern-tree) to transaction database DBase.

(1) For process the item I (i=n~1) of header table L of Frequent Pattern-tree from the back forward
(2) {
(3) Generate frequent 1-itemsets: M= {I|M.support=I.support}
(4) If (there are items behind I) {
(5) If all the items behind I are proceeded{
(6) Map_back(Frequent Pattern-tree,Ii);
(7) Increase_back(Frequent Pattern-tree,M,i);
(8) }
(9) Else
(10) Call algorithm 4
(11) }
(12) } Increase_back (Frequent Pattern-tree,M,i)
(13) For check the item Ij(j=n~i+1)of header table L of Frequent Pattern-tree from the back forward
(14) If (Ij.support>=minsup) {
(15) Generate frequent itemsets:M=(M{Ij}|M.support=Ij.support)
(16) If (there are frequent items behind Ij){
(17) Map_back (Frequent Pattern-tree,Ii);
(18) Increase_back (Frequent Pattern-tree, M,i);
(19) }
(20) }Map_back (Frequent Pattern-tree, Ii)
(21) For the item Ij (j=i+1~n) behind Ii in header table L
(22) According to the corresponding support number of item in Frequent Pattern-array update support number of item of the header table item.

Topic Modeling

The obtained patterns are incorporated into a topic modeling algorithm to extract topics in an unstructured manner. The topic modeling part uses Latent Dirichlet Allocation (LDA) model which is extended by sampling weights from a Dirichlet distribution, the conjugate before the multinomial distribution. Depending on the frequency of terms in each cluster the topics are classified using LDA. The aim is to extract topics from research papers and use them for the classification of research articles. The topic terms are obtained using cluster analysis, D-pattern, and Inner pattern evolving algorithms.

The usage of pattern mining algorithms with LDA will help to extract semantic information from texts as it represents term contextual information. This was lacking in LDA in Fig. 3. Configuration of the subsequent item terms and points. The LDA provides the data model with the ability to assign probabilities for each term related to a specific class or category. It assigns zero probabilities for entities outside the class boundaries or search space thereby reducing over fitting.

LDA accepts the accompanying generative process for each record in a corpus:

1. Pick B ~ Poisson (ξ)
2. Pick θ ~ Dir (α)
3. for every one of the *B* words *bn*:

(a) Choose a subject *Yk* Multinomial (θ)
(b) Choose a word *bn* from $p(wn|Yk, \beta)$

A multinomial probability adjusted on the theme *Yk*. First, the dimensionality k of the Dirichlet appropriation (and consequently the dimensionality of the point variable y) is expected to be known and fixed. Second, the word probabilities are defined by a $k \times V$ framework β where $\beta_{ij} = p\ (wj = 1|yi = 1)$, which for the present we treat as a proper amount that will be assessed. At long last, the Poisson supposition that isn't basic to whatever follows and more reasonable report length appropriations can be utilized on a case by case basis. The topics obtained from the LDA topic model are depicted in the table along with the number of articles. The segment named "T" lists every subject to work with referring to in the text in Fig. 5. Count how frequently a topic shows up as an essential theme inside a paragraph.

Next to "Nr." the three most significant terms are displayed close by with the relating ˇ dissemination values, which are switched over completely to up-sides since they are utilized distinctly for correlation purposes in Fig. 6. Visualize the topic distributions within the given articles/documents and a telescopic view of topics obtained from them.

According to Zheng *et al*. (2018), there is a document formation process by assuming D as document corpus with document class item archives (d) as follows.

In Algorithm 6 LDA first start with assuming 'T' topics in the document. The circle through 'D' the document and distribute each word in the report of any of the 'T' topics. In steps 7 ad 8 for each document loop through each word w and compute $p(wj|tk)$ and $p(tk|di)$. After completing the calculation then follow the step 9 for update the $p(wj|tkid)$ such as $p(wj|tkidi) = p(tk|di) \times p(wj|tk)$, until the loop through each word in each document. In step 10 reassign the topic for the currently selected word based on $p(wj|tkidi)$. Other repeat for all processes in the entire document, and finally collected the model.

The model is evaluated on known text samples and validated. The model sometimes falsely assigns a positive pattern for a topic as a negative. To solve the issue, a pattern improvement module is used.

Algorithm 6: Latent Dirichlet Allocation

1. start
2. Randomly assign "T" topics to all the words in 'D' documents
3. create document wise topic count.
4. Create topic wise assignment of word count from all documents.
5. Resample a word. (Eliminate topic assignment).
6. Decrement the count for the respective topic allocated from the document – topic matrix.
7. Calculate p $(t_k |D_i) = n_{ik}+\alpha/N_i-1+T_\alpha$ (where nik is the complete number of words in ith report in kth theme, Ni is the quantity of words the ith record, T is the quantity of topics considered. α is a hyper parameter.
8. Calculate $p(w_j|t) = mj_k+\beta / \sum_{j\epsilon v} d_{jk}+V\beta$ (where m_{jk} is the corpus wide task of document wj to kth theme. V is the jargon of the corpus. β is a hyper parameter.
9. calculate for word w_j $p(w_j |t_{ki} d_i) = p(t_k |d_i) \times p(w_j |t_k)$
10. Resampling / Reassign for a given word w_j in a document d_i, find the topic 'T' for which p $(w_j |t_{ki} d_i)$ is maximum and reassign the word to the 'T^{th}' topic.
11. Repeat for all (Repeat steps 4 to 10 for all the words in the entire document.)
12. Repeat process (Repeat steps 2 to 11 for a predefined number of iterations.)
13. stop

Cluster Assignment

Cluster analysis or clustering: The term cluster refers to the grouping of similar objects and these objects must have a low inter-term distance for those inside a cluster and a higher inter-term distance from terms in other clusters. In Fig. 7 k-means clusters are shown for two groups.

Hclust algorithm: The Hierarchical Agglomerative Clustering works according to the bottom-up manner. Initially, it treats every document as a cluster or a leaf node in a solitary tree. As the algorithms iterates, two subgroups are formed as clusters one supporting the document other against the document. The documents are iteratively compared to one another until optimal clusters are obtained. The hierarchy becomes longer as a new cluster node gets added in each iteration.

Algorithm 7: Hclust (type of Agglomerative Hierarchical clustering)

Step 1: Start

Step 2: Turn each input data set d into all documents, i.e., as single cluster

Step 2: For each pair of clusters p1, p2.

Step 3: Combine terms that are at lesser distance from the others

Step 4: Register distance between the new cluster and every one of the old clusters.

Step 5: Stop

In above algorithm 7, step 1 for preparing the input data in all documents, in step 2 loops the pair of clusters p_1 and p_2 then merges the clusters. In step 4 computing similarity information between every pair of clusters in the data set. Repeat step 2 and step 3, until a single cluster contains the group of all documents. Step 6 determines the most commonly used threshold value. The patterns that make up each topic will be clustered according to their semantic similarity scores. It forms the data dictionary for future classification tasks. The dictionary is again fed into the Topic model for validation. The steps are repeated until optimal accuracy is obtained.

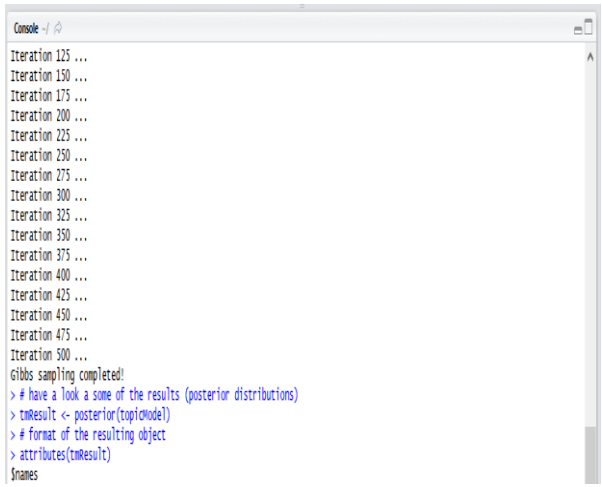

Fig. 3: Format of the resulting object terms and topics

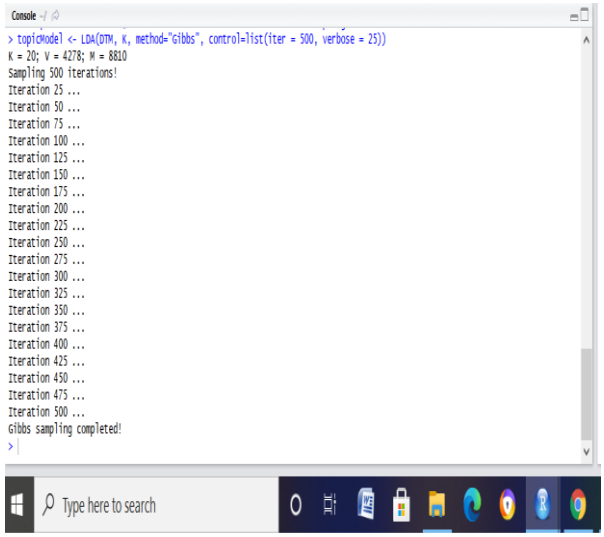

Fig. 4: LDA Model, Inference via 500 Iterations

```
Console
+   primaryTopic <- order(topicsPerDoc, decreasing = TRUE)[1]
+   countsOfPrimaryTopics[primaryTopic] <- countsOfPrimaryTopics[primaryTopic] + 1
+ }
> sort(countsOfPrimaryTopics, decreasing = TRUE)
        govern claim state unit treati          govern state relat unit island
                                623                                     594
      nation countri peopl peac great       public govern object great system
                                576                                     525
 year amount treasuri public expenditur          state congress unit act senat
                                524                                     521
        state war govern mexico peac congress attent report recommend subject
                                476                                     461
       bank gold govern currenc money       offic depart servic public appoint
                                428                                     420
    state constitut power govern peopl          law state unit court citizen
                                414                                     383
       state unit trade vessel nation  product countri duti import manufactur
                                373                                     373
     navi vessel ship naval construct     peopl govern condit nation interest
                                370                                     369
     land indian state tribe territori        year increas number mail servic
                                368                                     357
        armi war forc militari offic         state line territori great unit
                                336                                     319
>
```

Fig. 5: Count how often a topic appears as a primary topic within a paragraph

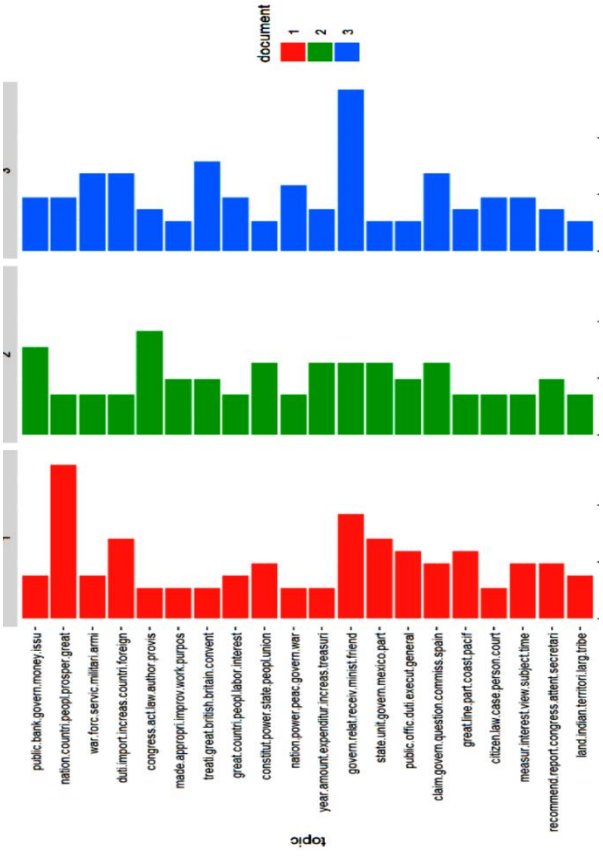

Fig. 6: Visualize the topic distributions within the documents and a distance view of the topics in the data over time

The Fig. 4 shows the number of iterations undertaken by Gibbs sampling technique. It evolves the initial patterns discovered through iterations. The algorithm converged after 500 iterations.

Experimental Results

The system is implemented using Intel Core (TM) i5 with big data analytics using in Tool - R and its corresponding libraries in Windows 10 OS having a RAM capacity of 8 GB. The empirical analysis intends to optimize the parameters of the model using our proposed methodology. It also evaluates the suitability of pattern discovery algorithms in big data. The data set is collected from an online research article database. The collection of documents taken for the review is 36,431 articles from Google Scholar, PubMed, NCBI, Elsevier, and IEEE.

Comparative Model

The low frequency and misinterpretation issues are controlled by the procedure of the pattern discovery. Two processes are used for Pattern Taxonomy Model (Inner Pattern Evolving) technique such as pattern deploying and pattern evolving for refining the text documents discovered patterns. The following experiments are carried out to assess the performance of PDBD against that of the Pattern Taxonomy Model (PTM) to prove the applicability of the proposed model in unstructured data. This study of pattern discovery in big data technique involves three processes, semantic information, and extraction by using Inner pattern improvement, topic modeling, and cluster assignment, to reduce the over fitting of discovered patterns in text using big data.

Metrics

An accuracy, review, and F-Measure are a couple of standard evaluations are used The term precision refers to retrieving all the documents which are relevant to a query by a user that was retrieved and the F-Measure (F1 score or F score) is a proportion of a test's exactness and is characterized as the weighted consonant mean of the accuracy and review of the test. The measures such as Precision, Recall, and F-Measure evaluate model effectiveness from the topic modeling perspective:

$$Weight(\mathrm{E}) = \sum e_{CTP} support(e)(e,p)$$

$$Precision = True\ Positive / True\ Positive + False\ Positive$$

$$Recall = True\ Positive / True\ Positive + False\ Negative$$

$$\mathrm{F-measure} = 2 * Precision * Recall / (Precision + Recall)$$

$$Threshold(PD) = min\left(\sum support(e)\right)$$

$$p \epsilon Dl(e,w) \epsilon \beta(p)$$

- Accuracy

Accuracy measure calculates the proportion of correctly predicted items to the total number of

predictions made for the entire corpus. It assesses the topic assignments for documents and their terms:

$$Accuracy = TP + TN / TP + FP + FN + TN$$

- The precision

It measures all positive predictions that belong to the positive class among all predictions. It gives the true positive measure of the model. The PDBD model obtained a precision score of 92% whereas the PTM model obtained a precision score of 68% as seen in Fig. 8. This proves that the proposed model outperformed the baseline model.

To find precisions, the data with different sizes starting from 1 TB to 5 GB was taken. In Fig. 8 X-axis denotes Document size, Y-axis denotes precision value. The graph was drawn in comparison to Pattern Taxonomy Model. The output of the Pattern Discovery in the big data model shows better performance than the achievement of the result % is 92%:

- The recall

The recall is the small number of recovered records among every pertinent document.TP (True Positive) represents positive documents, TP (True positive) represents false documents from the existing system and FN (False Negative) represents failure documents from the proposed system. In Fig. 9 X-axis denotes Document size, Y-axis denotes recall value. The existing Pattern Taxonomy Model (Inner Pattern Evolving) model has obtained a Recall score is 69%. The proposed (pattern discovery in big data) model has obtained a Recall score which is 91% the improvement is successful:

- The f-measure

The weighted harmonic mean of the system's precision and recall is defined as the F-measure:

$$F - Measure\ Score\ is\ 2*(Recall*Precision)/(Recall+Precision)$$

The existing Pattern Taxonomy Model (Inner Pattern Evolving) model has obtained an F-Measure score is 68%. The proposed (pattern discovery in big data) model has obtained an F-Measure score which is 92%. The improvement is effective. To find the F-Measure value, the data with different sizes starting from 1 TB to 5 TB was taken. In Fig. 10 X-axis denotes Document size, Y-axis denotes F-Measure. The following graph results were compared with pattern discovery in big data with Pattern Taxonomy Model (Inner Pattern Evolving). The output of the pattern discovery in big data model shows better performance than Pattern Taxonomy Model (Inner Pattern

Comparative Study of Result Analysis

In this part, we assess our proposed work. The Pattern Discovery Big Data Model (PDBD) over the data against Interactive Latent Dirichl*et Al*location (ILDA) and LDA with Dirichlet Prior Modified (LDA-DP) approach against metrics as Exactness, Precision, Recall, and F1 Measure-Score.

It is the weighted average or measure of optimizing Precision as well as Review. It relies upon both deceiving positives and false negatives. F1 score is genuine in circumstances where the class dispersal is exceptionally lopsided. The F1 score obtained in our case is 91%.

From the above Fig. 11 and Table 2, it is inferred that the proposed work excels in the comparative method with 91% accuracy, 92% precision, 91% Recall, and 91% F1 Score.

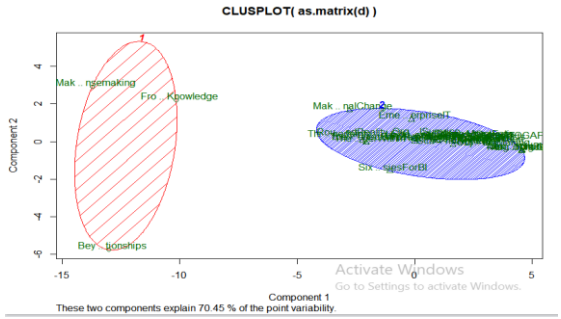

Fig. 7: K Means with two clusters

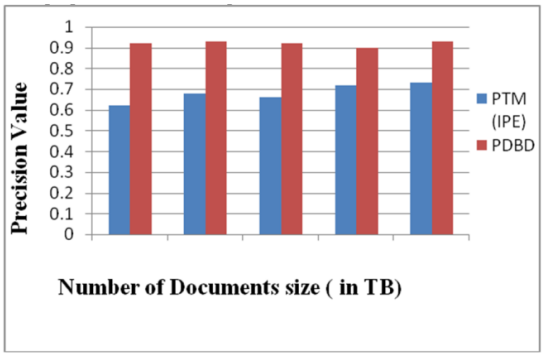

Fig. 8: Precision: Comparison of Pattern Taxonomy Model (Inner Pattern Evolving) and Pattern Discovery in big data model

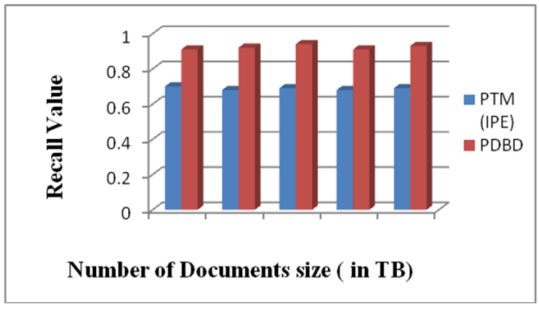

Fig. 9: Recall: Comparison of Pattern Taxonomy Model (Inner Pattern Evolving) and pattern discovery in big data model

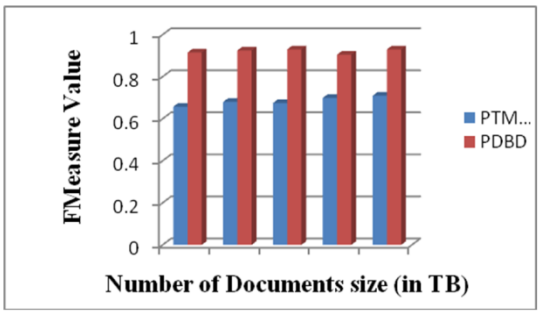

Fig. 10: F-measure: Comparison of pattern taxonomy model (inner pattern evolving) and pattern discovery in big data model

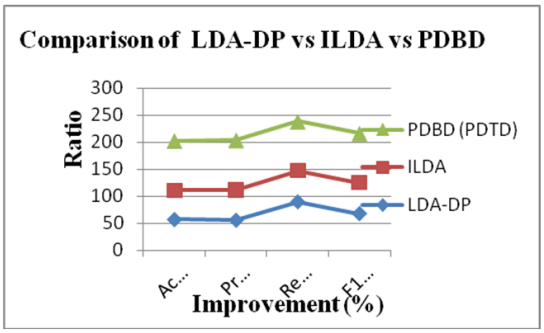

Fig. 11: Comparative graph result of LDA-DP, ILDA, and PDBD (pattern discover big data)

Table 1: Comparative Study of pattern taxonomy model (inner pattern evolving) and pattern discovery in big data

Model	Accuracy	Precision	Recall	F1-Score
PTM (IPE)	68.33	68	69	68
PDBD	91.66	92	91	92

Table 2: Comparative Study of LDA-DP, ILDA, and PDBD (PDTD-Pattern Discover Twitter Data)

Model	Accuracy	Precision	Recall	F1-Score
LDA-DP	57.20	55.90	89.98	68.20
ILDA	54.00	56.01	57.60	57.25
PDBD (PDTD)	91.66	92.00	91.00	91.00

The Table 1 states the comparison among performance metrics. The proposed model achieved approximately 92% accuracy in extracting the valid patterns from the documents. The performance is again complemented by its score achieved in precision, recall and F1-scores.

The Table 2 states the comparison among performance metrics on Twitter dataset. The proposed model achieved approximately 92% accuracy which is well above the compared models. The performance is again emphasized by its score achieved in precision, recall and F1-scores.

Discussion

From the above experiments and the results, it is shown that the model with patterns discovery in the big data model performs the best. The identification of semantic terms shows that the terms that are "important" for that specific topic but not necessarily "important" for the whole corpus. The frequent patterns help achieve the objective of finding the most dominant topic in each article which covers most of the article part. It also differentiates what was the number of documents that came across different topics and how similar those documents were to be lying on the same topic. The results were really impressive from the various performance metrics experiments as the topics are unique and separable. The datasets are taken for the study also contained a similar set of articles across each unique topic. Also, it is to be noticed that Pattern Discovery in Big Data (PDBD) model provides the best scholarly graphic topics contrasted with different techniques, aside from some methods that failed to create topics that aggregate related words. The PDBD (Pattern Discovery in Big Data) model produces higher-quality topics and more coherent topics than the other methods in our evaluated datasets. Removing less quantity of keywords prompted a high coherence score in Pattern Discovery in Big Data (PDBD).

Conclusion

By combining, synthesizing, and displaying data from enormous amounts of research, we were able to uncover important themes in the literature. The relationships between text content and its subjects are mapped in this study. Because of the rapid accumulation of research articles, text mining technologies are critical for boosting the possibility that publications will be discovered by individuals who are most interested in applying knowledge to improve classification performance. If the data could be transformed into a learnable structure with proper labels for supervised learning, numerous valuable machine learning applications may be developed. Based on research article datasets, we conducted a comparative analysis of several pattern mining methodologies and tactics.

The discovery of sequential patterns with both frequent and closed patterns was studied. In the atopic modeler, the produced patterns were validated and topic patterns were optimized. Using semantic patterns, the model was compared to other existing algorithms. The exactness of the proposed model's outcomes was higher than the baselines. Our tests suggest that the proposed model of pattern discovery in big data has the potential to reduce the cost of manual labeling for future classification tasks by a significant amount. In the future, the proposed methodology could be enhanced to extract semantic knowledge from additional huge databases. Other types of data besides those discussed in this article could benefit from the pre-processing approaches.

Acknowledgment

I thank our research supervisor from MKU, Madurai. Who provided insight and greatly assisted the research.

Author's Contributions

Kamatchi Sundravadivelu: Writing-original draft, writing-editing, software, visualization, and Investigation.

Muthuraman Thangaraj: Methodology, Formal analysis.

Ethics

The paper reflects the authors' own research and analysis in a truthful and complete manner.

References

Amado, A., Cortez, P., Rita, P., & Moro, S. (2018). Research trends on big data in Marketing: A text mining and topic modeling based literature analysis. European Research on Management and Business Economics,24(1),1-7. doi.org/10.1016/j.iedeen.2017.06.002

Anujna, M., Ushadevi, A. (2017). Converting and deploying an unstructured data using pattern matching. American Journal of Intelligent Systems. 7(3), 54-9.

Aswini, V., & Lavanya, S. K. (2014, April). Pattern discovery for text mining. In 2014 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC) (pp. 412-416). IEEE. https://ieeexplore.ieee.org/abstract/document/6915399

Cai, L., Qi, Y., Wei, W., Wu, J., & Li, J. (2019). mrMoulder: A recommendation-based adaptive parameter tuning approach for big data processing platform. Future Generation Computer Systems, 93, 570-582. doi.org/10.1016/j.future.2018.05.080

Haq, M. I. U., Li, Q., & Hassan, S. (2019). Text mining techniques to capture facts for cloud computing adoption and big data processing. IEEE Access, 7, 162254-162267. doi.org/10.1109/ACCESS.2019.2950045

Indra, R., & Thangaraj, M. (2019). An integrated recommender system using semantic web with social tagging system. International Journal on Semantic Web and Information Systems (IJSWIS), 15(2), 47-67. doi.org/10.4018/IJSWIS.2019040103

Li, S., Tian, X., Zuo, L., & Li, X. (2015, October). A synonym mapping method of operators in mathematical formula retrieval. In 2015 8th International Conference on Biomedical Engineering and Informatics (BMEI) (pp. 629-633). IEEE. doi.org/10.1109/BMEI.2015.7401580

Lucchese, C., Orlando, S., & Perego, R. (2013). A Unifying Framework for Mining Approximate Top-$ k $ Binary Patterns. IEEE Transactions on Knowledge and Data Engineering, 26(12), 2900-2913. doi.org/10.1109/TKDE.2013.181

Lucchese, M., & Sundaravadivelu, K. (2020). Mining effective patterns from text data-a survey International Journal of Scientific & Technology Research. ISSN-10: 2277-86161930 IJSTR©2020

Thangaraj, M., Sundravadivelu, K. (2022), Survey on Pattern Discovery in Text Mining, International Journal of Computer engineering and Applications, 2321-3469. https://www.ijcea.com/

Min, F., Zhang, Z. H., Zhai, W. J., & Shen, R. P. (2020). Frequent pattern discovery with tri-partition alphabets. Information Sciences, 507, 715-732. doi.org/10.1016/j.ins.2018.04.013

Palaniammal, K. (2021). Refining Search Performance through Semantic based CBR Model and QoS Ranking Methodology. Indian Journal of Science and Technology. 2021, 14(21), 1775-85.

Palaniammal, K., & Vijayalakshmi, S. (2013). Ontology Based Meaningful Search Using Semantic Web and Natural Language Processing Techniques. ICTACT Journal on Soft Computing, 4(01), 662-666.

Pawar, T. A., & Karande, N. D. (2014). Effective Pattern Discovery for Text Mining Using Pattern Based Approach. International Journal of Advance Research in Computer Science and Management Studies, 2(9).

Sujatha, G. S., & Poonguzhali E, (2018). Text Mining Using Pattern Taxonomy Model for Effective Pattern Discovery, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181.

Sze-To, A., & Wong, A. K. (2018). Discovering patterns from sequences using pattern-directed aligned pattern clustering. IEEE Transactions on NanoBioscience, 17(3), 209-218. doi.org/10.1109/TNB.2018.2845741

Thangaraj, M., & Gayathri, V. (2013). A context-based technique using tag-tree for an effective retrieval from a digital literature collection. Journal of Computer Science, 9(11), 1602. doi.org/10.3844/jcssp.2013.1602.1617

Thangaraj, M., & Karthikeyan, V. K. T. (2014). KT-grand: An algorithm for web content filtering. J AdvaResea Comp Sci Mana Stud, 2(9), 371-376.

Thangaraj, M., & Sivakami, M. (2018). Text classification techniques: A literature review. Interdisciplinary Journal of Information, Knowledge and Management, 13, 117.doi.org/10.28945/4066

Thangaraj, M., & Vijayalakshmi, C. R. (2013). Performance study on rule-based classification techniques across multiple database relations. International Journal of Applied Information Systems, 5(4), 1-7.

Viloria, A., López, J. R., Leyva, D. M. G., Vargas-Mercado, C., Hernández-Palma, H., Llinas, N. O., & Rodriguez, J. V. (2019). Data Mining Techniques and Multivariate Analysis to Discover Patterns in University Final Researches. Procedia Computer Science, 155, 581-586. doi.org/10.1016/j.procs.2019.08.081

Wan, J., Zheng, P., Si, H., Xiong, N. N., Zhang, W., & Vasilakos, A. V. (2019). An artificial intelligence driven multi-feature extraction scheme for big data detection. IEEE Access, 7, 80122-80132. doi.org/10.1109/ACCESS.2019.2923583

Wu, J., Guo, S., Huang, H., Liu, W., & Xiang, Y. (2018). Information and communications technologies for sustainable development goals: State-of-the-art, needs and perspectives. IEEE Communications Surveys & Tutorials, 20(3), 2389-2406. doi.org/10.1109/COMST.2018.2812301

Xuan, J., Lu, J., Zhang, G., & Luo, X. (2015). Topic model for graph mining. IEEE transactions on cybernetics. 2015, 45(12), 2792-803.

Yao, Q., Gao, X., Lei, X., & Zhang, T. (2016, December). The Research and Improvement Based on FP-Growth Data Mining Algorithm. In 2016 International Conference on Modeling, Simulation and Optimization Technologies and Applications (MSOTA2016) (pp. 287-293). Atlantis Press.

Zheng, L., Caiming, Z., & Caixian, C. (2018). MMDF-LDA: An improved Multi-Modal Latent Dirichlet Allocation model for social image annotation. Expert Systems with Applications, 104, 168-184. doi.org/10.1016/j.eswa.2018.03.014

Zhong, N., Li, Y., & Wu, S. T. (2010). Effective pattern discovery for text mining. IEEE transactions on knowledge and data engineering, 24(1), 30-44. doi.org/10.1109/TKDE.2010.211