Journal of
Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7572605

CONTEXT AWARE EXTRACTION OF CONCEPTS FROM UNSTRUCTURED DATA USING MACHINE LEARNING ALGORITHMS

[1]Shankarayya Shastri, [2]Dr.Veeragangadharaswamy T.M
[1]Research Scholar, Dept. of CSE, RYMCE, Ballari -583104
[2]Professor, Dept. of CSE, RYMCE, Ballari -583104
shan.shas@gmail.com

ABSTRACT: Data analysis is a key procedure in the part of data science that obtains needed information from any statistics. The ease of access and maintenance makes structured data the most popular choice among many organizations even today. On the other hand, with the fast development of technology, large amounts of more unstructured data like text and image are being generated. Unstructured data is data that didn't have any pre-defined system associated with it. Because of the accessibility of a high number of electronic text records from various sources explaining unstructured and semi-structured data, document categorization work becomes an interesting part to control data nature. Text classification is an efficient activity that can be achieved using the originality of categorization algorithms. Recently, Machine Learning (ML) approaches offer a novel chance to emerge unstructured data into existing knowledge bases without the requirement to manually organize the data into topic-based content enriched with semantic metadata. Hence in this work, Context aware extraction of concepts from unstructured data using Machine Learning algorithms is presented. Textured Context Pattern (TCP) method with Lexical Subgroup (LS) model is used to explain the relevancy between the feature of query document and from whole dataset. the Unsupervised Cross-Correlated Neural Network (CCNN) is used to find the matching feature. The performance of presented model is described regarding Precision, Recall and F1-score.

KEYWORDS: Structured data, Unstructured data, Machine Learning, Cross-Correlated Neural Network (CCNN).

I. INTRODUCTION

With increasing proliferation and technological advancement, a large amount of text data is generated every day in the form of social media platform, websites, company information, healthcare information and news. Indeed, extracting interesting patterns like opinions, summaries and facts of different lengths from text data is a difficult task [6]. The exponential enhancement of text data and ongoing growth of data age, forms managing like huge amounts of information even more difficult. Online text is semi-structured or unstructured; Examples contains academic papers, online journals, news sources, and books. Before growth of technology, people only process this lot of information, which takes much period [1].

Data analysis and its complexity vary according to the type of data. The complexity of analysis is associated with several aspects such as data resources, the accuracy of analysis, and domain dependence. Structured data is similar to machine-language and forms the operation and management of data very straightforward; unstructured data, however, is natural language text that does not have a severe semantic structure or database format. Obviously; If unstructured data can be readily transformed into structured data, it becomes easier to derive intelligence from unstructured data.

Unstructured data is asymmetrical data without a predefined information system. A continuous stream of data over time is unstructured and grouping this information is a laborious work as it lacks class labels and accumulates above the period. As information continues to grow, this makes challenging to train and generate a system from scratches every duration. Since there are many types of records on the Internet that are used on a large scale in various applications, document type identification for classification models to facilitate further operations is a complex task. Text semi-structured and unstructured files have several varience in their behaviour, including the structure of text presentation, level of ambivalence, degree of useful and utilization of idioms, punctuation and metaphors [9].

Key phrases are critical for searching and sorting scholarly papers. A list of key-phrases is an important feature of a scientific text. Key-phrases consist of concise representation of the contents of the text. They assist to search engines notice and sort papers [4]. Qualitative selection of key phrases definitely affects the visibility of the paper and its number of citations. Text classifiers will arrange, organize, and classify nearly any type of text, including documents, medical research, files, and content noticed in the web. Unstructured data accounts for over 80% of all information, with text being one the regular classify. Analyzing, absorbing, arranging and shifting by text information are hard and time utilization because of its disordered behaviour, many businesses didn't utilize its complete possibility [2].

Text classification is a basic work in the part of Natural Language Processing (NLP) and is often used in information retrieval, untrustworthiness analysis and detection, sentiment analysis, detection of spam emails, etc. Text mining is a word used to explain the system of obtaining forms or knowledge from unstructured data. Text classification is a model in which they need to obtain needed data from text. This is where ML and text classification come into play. Text classifiers are utilized to fast and low expensive classify all relevant text types, including emails, legal documents, social media, surveys etc. In the data learning and the prediction process, there are several models of classification techniques are to validate the data based on the similarity metrics. For scholars and other researchers needs to search for references that are related to the work and for the documentation process. The reason for file categorization, available algorithms examine the distribution of content terms in a file.

Artificial intelligence (AI), especially, machine learning (ML) has developed fast in present time in the context of data analysis and computing, which generally permits functions to act in an intelligent approach [8]. ML generally gives systems the capability to study and improve automatically from understanding without being specially programmed, and normally represents the latest technologies in the Fourth Industrial Revolution. There are different types of ML algorithms in this area such as supervised, unsupervised, semi-supervised and reinforcement learning. ML helps to search for the related documents that are matched with the given reference.

Hence in this work, Context aware extraction of concepts from unstructured data using Machine Learning algorithms is described. The remaining work is organized as follows: The section II explains the literature survey. The section III describes Context aware extraction of concepts from unstructured data using Machine Learning algorithms. The section IV describes the result analysis of presented approach. Finally, the work is concluded in section V.

II. LITERATURE SURVEY

Dan Zhang et. al., [10] explains Text Complexity Classification Data Mining Model Based on Dynamic Quantitative Relationship between Modality and English Context. *is article starts with a conceptual investigation of text complexity analysis and describes the five parameters of dynamic, complexity, hiding, sentiment, and ambiguity, including the origin of text complexity and the expression of user needs in a networked environment. Second, depended on the particular method of text mining, data collection, data processing and data visualization, it is described to group user request analyzes into three phases of text difficulty accession, identification and statement. Examination depended on text mining technology. The experimental output shows gathered quantitative relevant data is noticed and described to understand the conversion of quantitative relevant data into product characteristics.

Dongdong Zhang, Changchang Yin, Jucheng Zeng, Xiaohui Yuan and Ping Zhang et. Al., [11] describes combining structured and unstructured data for predictive models: a deep learning model. In this analysis, the authors present 2 common-reason multi-modal neural network models by embedding sequential structured files with structured information to enhance patient representation learning. The described combination systems support files embeddings for the presentation of extensive clinical notes files and convolutional neural networks to system of sequential clinical notes and temporal signals or LSTM networks as well as one-hot encoding for statistical data.

Fouad Zablith, Ibrahim H. Osman et. al., [14] presents Review Modus: Text Classification and Sentiment Prediction of Unstructured Reviews using a Hybrid Combination of ML and Performance Models. Review Modes, a text mining and processing framework (1) depends on model structure and its comparable prediction questions to instruct an ML models to detect the categorization of feedbacks around algorithm measurements; (2) An external review detects sentiment in feedbacks are depended on training datasets; and (3) Transforms scales obtained from feedback for later examination. The method is calculated in the factors of 11 e-government services, at which evaluation of the model differs from the physical operation of structured feedback verified by three individual examiners.

Sathya Madhusudhanan, Suresh Jaganathan and Jayashree L S et. Al., [15] presents Incremental Learning for Classification of Unstructured Data Using ELM. A framework for clustering metadata describes CUIL (Classification of Unstructured Data), that clusters metadata, gives a name to every cluster, and later makes a system utilizing an Extreme Learning Machine (ELM), an incremental feed-forward neural network, information for every batch. Based on the tabular results, this work shows high accuracy and efficiency. However, this work has some limitations: (1) Complexity for resolving random weights by trial and error model till the intended accuracy is attained for the training dataset as well as (2) Complexity in selecting total of invisible neurons, i.e., the total of invisible neurons enhances as high accuracy is attained.

Lipika Dey, Hardik Meisheri and Ishan Verma et. al., [17] presents Predictive Analytics with Structured and Unstructured data - Deep Learning based approach. A general deep learning framework for predictive analytics using both unstructured and unstructured data is presented. That describe a case-study to check the operation and applicability of the described model where it use LSTM to predict the movement direction of structured data. utilizes events obtained from news documents.

Shucheng Gong and Hongyan Liu et. al., [19] Constructs Decision Trees for Unstructured Data. A decision tree construction framework is known as CUST is described and that will solve structured text. CUST explains the uses of partitioning importance made by unstructured feature values and decreases the total scans in datasets by creating accurate data forms. Observations on real-world datasets shows that CUST enhances ability to build categorizes for structured data.

III. CONTEXT AWARE EXTRACTION OF CONCEPTS FROM UNSTRUCTURED DATA

Context aware extraction of concepts from unstructured data using Machine Learning algorithms is presented in this section. The flow diagram of presented approach is shown in Fig. 1.

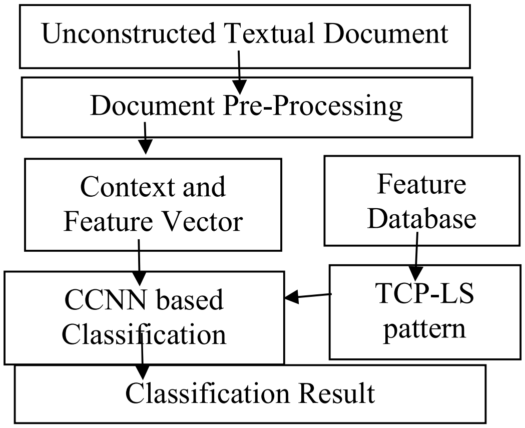

Unconstructed Textual Document
Document Pre-Processing
Context and Feature Vector
Feature Database
CCNN based Classification
TCP-LS pattern
Classification Result

Fig. 1: Flow Diagram of Context aware extraction of concepts from unstructured data

The main aim is to satisfy the objectives that are listed as follows: To present the domain specific representation techniques for unstructured text data and to develop a technique for discovering relationship between overlapping or similar meaning of words in large clusters.

Unstructured data, usually text, is data that does not have a predefined format (eg, e-mail, word processing documents, or presentations). Unstructured data is created and gathered for variety types, that is Word documents, email, PowerPoint presentations, survey responses, transcripts of call center interactions, posts from blogs, and social media sites. Different type of unstructured information consists of images, audio and video documents. Unstructured documents are free from just documents and don't have a set structure but can still be scanned, captured and imported.

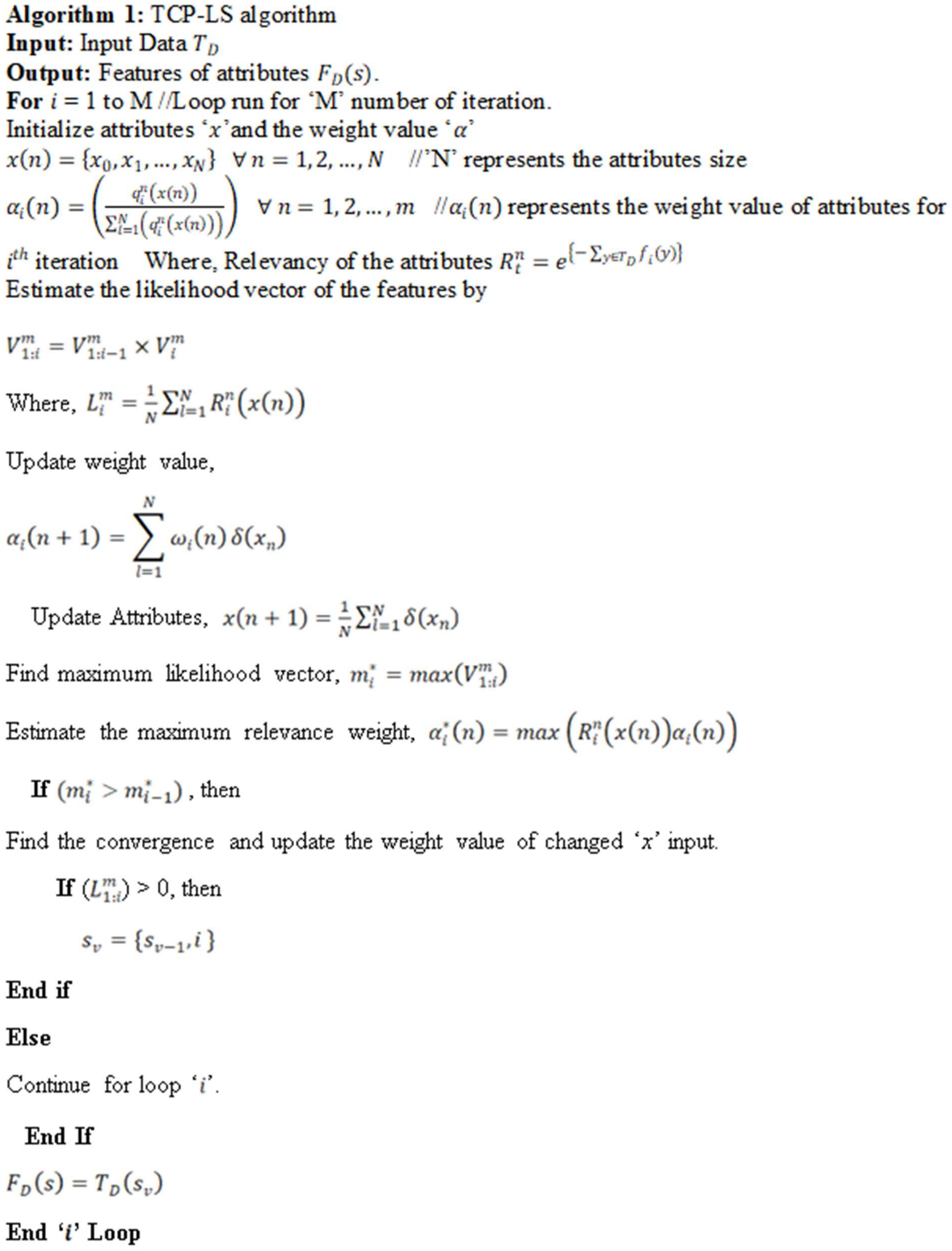
In this analysis, the unstructural text documents are taken from computer folders where files are related to database, networking, database, image processing, etc. The file might be csv file or pdf file. The unstructured documents data is need to pre-processed because computer is a machine that does not have the capability to know explicitly how to process text or worse. The pre-processing improves the reliability and accuracy. The pre-processing stage removes missing or inconsistent data values resulting from human or computer error which can enhance the accuracy and standard of dataset forms it more reliable. It makes information consistent.

Data pre-processing includes different methods like stemming, segmentation, lemmatization, stop word removal, etc. Stemming is the procedure of reducing words to their root by removing inflections by dropping unrequited letters, usually suffixes. Stemming and lemmatization are techniques utilized by search engines and chatbots to describe meaning behind a word. Stemming utilizes root of word, during lemmatization utilizes text where term is utilized. Data segmentation involves taking information and grouped it and grouping equal data based on selected parameters so that it can be used highly beneficially. Stop word cutting is the most regularly used pre-processing steps in various NLP applications. The idea is to remove words that appear in common across all documents in the corpus. In general, articles and pronouns are usually classified as stop words. Stop words are abundantly available in any human language. By removing these words, low-level information is removed from the text to focus more on significant data.

Data pre-processing stage perform certain operation like null values handling, redundancies, and unrequited information. Data can be misattributed and contain noise. Cleaning and fixing false or inaccurate data from records and datasets requires finding and correcting (or removing) incorrect, inaccurate, or meaningless information, such as replacing, updating, or deleting confidential material. This step removes semicolons and commas. Dataset consists of many problems like punctuation marks, pronunciation marks and remaining particular characters, numbers, white spaces, abbreviations, capital letters, etc. For this reason, they should be standardized.

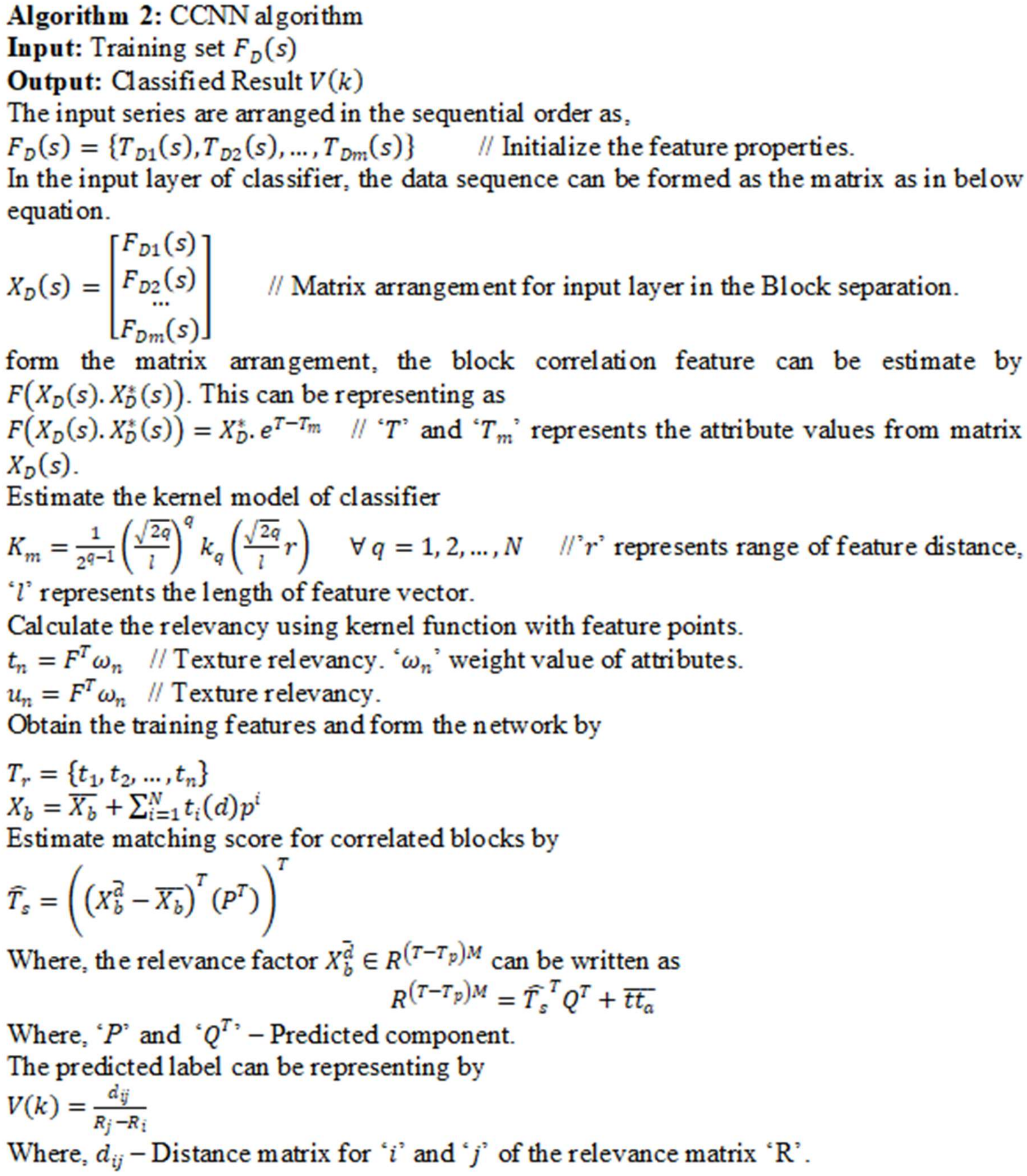
The total of invisible states of the input series weighted by the alignment scores refers to a context vector. Every term in input series is referred by combination of both (ie, forward and backward). A feature vector is an ordered list of numerical features of an observed phenomenon. It represents the input features to the predictive machine learning model. Humans can analyze qualitative data to make decisions. There are two types of context: physical context and linguistic context.Context and feature vectors are used to identify document belongs to which type.

To achieve the objectives key points and to enhance the classification performance and to reduce the time complexity for data analysis, optimization method can be implemented to select best matching of the keywords and feature attributes that are relevant to the query document. This can be achieved by using the Textured Context Pattern (TCP) model with Lexical Subgroup (LS) method to find relevancy among feature of query document and from entire dataset. A lexicon is referred to as a part of an NLP model which consists of data (semantic, grammatical) related to the independent terms or word strings. Lexical representations correspond to whole word forms, while lexical subgroup representations correspond to sequences of word parts or segments. Description of TCP-LS Algorithm is as follows:



```
**Algorithm 1:** TCP-LS algorithm
**Input:** Input Data $T_D$
**Output:** Features of attributes $F_D(s)$.
**For** $i$ = 1 to M //Loop run for 'M' number of iteration.
Initialize attributes '$x$'and the weight value '$\alpha$'
$x(n) = \{x_0, x_1, \dots, x_N\}$  $\forall n = 1, 2, \dots, N$   //'N' represents the attributes size
$\alpha_i(n) = \left(\frac{q_i^n(x(n))}{\sum_{l=1}^{N}\left(q_i^n(x(n))\right)}\right)$  $\forall n = 1, 2, \dots, m$  //$\alpha_i(n)$ represents the weight value of attributes for
$i^{th}$ iteration   Where, Relevancy of the attributes $R_t^n = e^{\{-\sum_{y \in T_D} f_i(y)\}}$
Estimate the likelihood vector of the features by

$V_{1:i}^m = V_{1:i-1}^m \times V_i^m$

Where, $L_i^m = \frac{1}{N}\sum_{l=1}^{N} R_i^n\left(x(n)\right)$

Update weight value,

$\alpha_i(n+1) = \sum_{l=1}^{N} \omega_i(n)\,\delta(x_n)$

  Update Attributes, $x(n+1) = \frac{1}{N}\sum_{l=1}^{N} \delta(x_n)$

Find maximum likelihood vector, $m_i^* = max(V_{1:i}^m)$

Estimate the maximum relevance weight, $\alpha_i^*(n) = max\left(R_i^n\left(x(n)\right)\alpha_i(n)\right)$

  **If** $(m_i^* > m_{i-1}^*)$ , then

Find the convergence and update the weight value of changed '$x$' input.

    **If** $(L_{1:i}^m) > 0$, then

      $s_v = \{s_{v-1}, i\}$

**End if**

**Else**

Continue for loop '$i$'.

 **End If**

$F_D(s) = T_D(s_v)$

**End '$i$' Loop**
```

To find the matching feature, the Unsupervised Cross-Correlated Neural Network (CCNN). Correlation describes how one or more variables are related to each other, these variables can be features of the input data used to predict the target variable. With this system, first the pre-processed feature is matched with the pattern by using CCNN to find the type of data without directly passed into the whole dataset. The algorithm description of CCNN classifier is as follows:



```
Algorithm 2: CCNN algorithm
Input: Training set $F_D(s)$
Output: Classified Result $V(k)$
The input series are arranged in the sequential order as,
$F_D(s) = \{T_{D1}(s), T_{D2}(s), \dots, T_{Dm}(s)\}$        // Initialize the feature properties.
In the input layer of classifier, the data sequence can be formed as the matrix as in below equation.
$X_D(s) = \begin{bmatrix} F_{D1}(s) \\ F_{D2}(s) \\ \dots \\ F_{Dm}(s) \end{bmatrix}$        // Matrix arrangement for input layer in the Block separation.
form the matrix arrangement, the block correlation feature can be estimate by $F(X_D(s).X_D^*(s))$. This can be representing as
$F(X_D(s).X_D^*(s)) = X_D^*.e^{T-T_m}$    // 'T' and '$T_m$' represents the attribute values from matrix $X_D(s)$.
Estimate the kernel model of classifier
$K_m = \frac{1}{2^{q-1}}\left(\frac{\sqrt{2q}}{l}\right)^q k_q\left(\frac{\sqrt{2q}}{l}r\right) \quad \forall q = 1, 2, \dots, N$    //'r' represents range of feature distance, 'l' represents the length of feature vector.
Calculate the relevancy using kernel function with feature points.
$t_n = F^T\omega_n$    // Texture relevancy. '$\omega_n$' weight value of attributes.
$u_n = F^T\omega_n$    // Texture relevancy.
Obtain the training features and form the network by

$T_r = \{t_1, t_2, \dots, t_n\}$
$X_b = \overline{X_b} + \sum_{i=1}^{N} t_i(d)p^i$
Estimate matching score for correlated blocks by
$\widehat{T_s} = \left(\left(X_b^{\bar{d}} - \overline{X_b}\right)^T (P^T)\right)^T$
Where, the relevance factor $X_b^{\bar{d}} \in R^{(T-T_p)M}$ can be written as
                                        $R^{(T-T_p)M} = \widehat{T_s}^T Q^T + \overline{tt_a}$
Where, 'P' and '$Q^T$' – Predicted component.
The predicted label can be representing by
$V(k) = \frac{d_{ij}}{R_j - R_i}$
Where, $d_{ij}$ – Distance matrix for 'i' and 'j' of the relevance matrix 'R'.
```

To identify the matching document, the keywords from the statements are gathered to form the attributes for searching the queries. Then with the similarity metrics, the best matching documents are short listed to recommend the relevant documents. Finally, presented context aware text pattern prediction process for the dataset can be compare with 70%, 80% and 90%

training data. Finally, the CCNN classifies the document that belongs to which type. Feature database is used to create the subfolder which is used to store the classification result.

IV. RESULT ANALYSIS

In this section, Context aware extraction of concepts from unstructured data using Machine Learning algorithms is implemented through python. In this approach three different datasets namely SemEval2010, Wiki20 and Pubmed datasets are used. The performance of presented approach is investigated with different datasets.

It is the significant standardized dataset, Gathered 244 complete scientific papers of SemEval-2010 from ACM Library. From these six to eight page essays enclose four various parameters of computer science: distributed Artificial Intelligence Information Search and Retrieval, Social and behavioral Sciences and Distributed Systems. The reporter and experts selects the keywords for every document. Its processing duration is 1.078 sec. The current keyphrases of 3129 and 656 are missing.

PubMed Central full papers are generated from PubMed corpora of life science journals nearly 0.026 billion ebooks of MIDLINE cites. This is garthering of 500 stories collected from single sources. Medical Subject Headings (MeSH), is a keyword dictionary is utilized to describe papers, PubMed of gold keyword contains 14.24 keyphrases for each document. It contains 2513 presented and 4607 non- presented keyphrases with a processing time of 0.266 seconds.

Wiki-20 gathering consists of twenty English-language scientific analysis documents of computer science contains on different contents. Every report was assigned keywords with fifteen teams, every team consisting of two senior candidates in computer science, using the names of Wikipedia articles as student vocabulary. Every article should have about 5 keywords assigned from teams. By an average, 5.7 keywords are accepted from every team. The presented keyphrase of 315 and a non- presented keyphrase of 395 with an processing time of 0.016 seconds. The result analysis of presented approach is calculate using features like, true positive rate (sensitivity), precision, F1-score and accuracy with the reference of Ground truth of database. Precision, recall and F1-score are three significant and relevant metrics which are utilized in this model to differentiate the evaluation with remaining models.

Precision: The precision is worked to evaluate the better forms, which are detected exactly from the number of detected forms in a true class.

$$\text{Precision}=\text{Key}_{\text{Corrected}}/\text{Key}_{\text{Predicted}} \quad (1)$$

where KeyCorrected is the total exactly detected key-phrases matched against standard key-phrases and KeyPredicted is the number of detected key-phrases from the file.

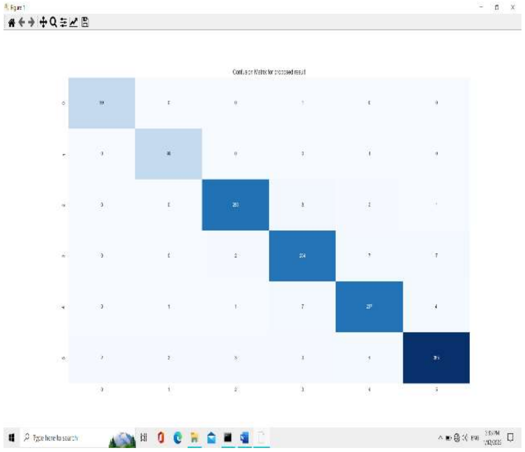
Recall: It is also called as sensitivity. It is the ratio of positive values exactly predicted relative to exact positive values; and can be calculated as

$$\text{Recall}=\text{Key}_{\text{Corrected}}/\text{Key}_{\text{Predicted}} \quad (2)$$

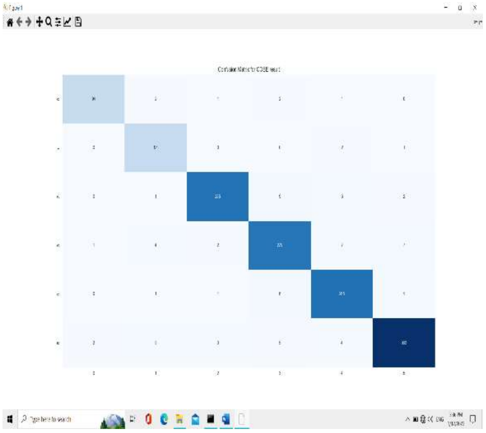
F1-Score: F1-score is the significant performance in ML. It nicely summarizes the model's detection evaluation by integrating the two evaluations precision and recall.

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Specificity: It is algorithm/model's ability to classify a true negative of each category available. The Fig. 2 (a) shows the confusion matrix for presented approach and 2(b) shows the confusion matrix for COBEC approach.

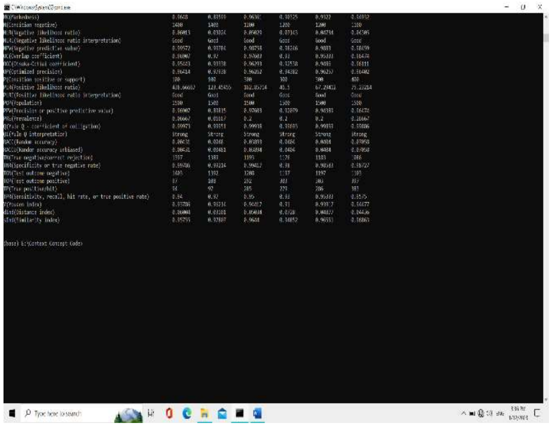


(a)



(b)
Fig. 2: Confusion Matrix for (a) Presented Matrix and (b) COBEC Approach

The Fig. 3 shows the output screen of implemented Context aware extraction of concepts from unstructured data.



```
MK(Markedness)                                           0.9648      0.83559     0.96367     0.90525     0.9322      0.86932
N(Condition negative)                                    1400        1400        1200        1200        1200        1100
NLR(Negative likelihood ratio)                           0.06013     0.03024     0.05029     0.07143     0.04734     0.04305
NLRI(Negative likelihood ratio interpretation)           Good        Good        Good        Good        Good        Good
NPV(Negative predictive value)                           0.99572     0.99784     0.98758     0.98246     0.9883      0.98659
OC(Overlap coefficient)                                  0.96907     0.97        0.97603     0.93        0.95333     0.96474
OOC(Otsuka-Ochiai coefficient)                           0.95223     0.93338     0.96293     0.92538     0.9485      0.96111
OP(Optimized precision)                                  0.96414     0.97918     0.96262     0.94382     0.96257     0.96402
P(Condition positive or support)                         100         100         300         300         300         400
PLR(Positive likelihood ratio)                           438.66667   123.45455   162.85714   46.5        67.29412    75.23214
PLRI(Positive likelihood ratio interpretation)           Good        Good        Good        Good        Good        Good
POP(Population)                                          1500        1500        1500        1500        1500        1500
PPV(Precision or positive predictive value)              0.96907     0.89815     0.97603     0.92079     0.94333     0.96478
PRE(Prevalence)                                          0.06667     0.06667     0.2         0.2         0.2         0.26667
Q(Yule Q - coefficient of colligation)                   0.99973     0.99851     0.99938     0.99695     0.99853     0.99886
QI(Yule Q interpretation)                                Strong      Strong      Strong      Strong      Strong      Strong
RACC(Random accuracy)                                    0.00431     0.0048      0.03893     0.0404      0.0404      0.07058
RACCU(Random accuracy unbiased)                          0.00431     0.00481     0.03894     0.0404      0.0404      0.07058
TN(True negative/correct rejection)                      1397        1389        1193        1176        1183        1086
TNR(Specificity or true negative rate)                   0.99786     0.99214     0.99417     0.98        0.98583     0.98727
TON(Test outcome negative)                               1403        1392        1208        1197        1197        1103
TOP(Test outcome positive)                               97          108         292         303         303         397
TP(True positive/hit)                                    94          97          285         279         286         383
TPR(Sensitivity, recall, hit rate, or true positive rate) 0.94       0.97        0.95        0.93        0.95333     0.9575
Y(Youden index)                                          0.93786     0.96214     0.94417     0.91        0.93917     0.94477
dInd(Distance index)                                     0.06004     0.03101     0.05034     0.0728      0.04827     0.04436
sInd(Similarity index)                                   0.95755     0.97807     0.9644      0.94852     0.96553     0.96863


(base) E:\Context Concept Code>
```

Fig. 3: Output Screen

The table 1 shows the performance evaluation of SemEval2010 dataset.

Table 1: Performance Evaluation on SemEval2010 dataset

Methods	F1-score	Precision	Recall
KCFA	0.9051	1	0.8627
Presented Context aware extraction of concepts from unstructured data using ML algorithms	0.9248	1	0.8863

Compared to KCFA, presented approach has better results in terms of precision, recall and F1-score. The Fig. 4 shows the performance metrics comparison.

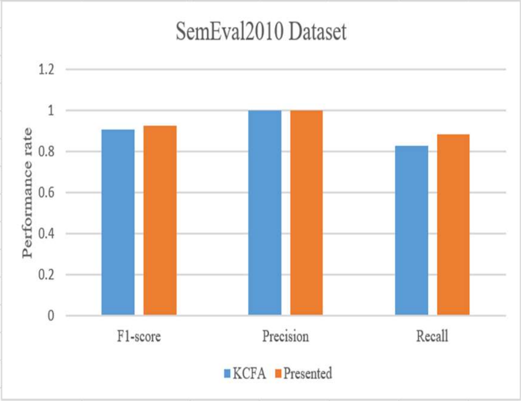


Fig. 4: SemEval2010 dataset performance metrics comparison

In Fig. 2 the x-axis as well as y-axis indicates different methods and performance rate respectively. Presented approach using SemEval 2010 dataset has high performance than KCFA approach. The table 2 represents the performance metrics evaluation on Wiki20 dataset.

Table 2: Performance Evaluation on Wiki20 dataset

Methods	Precision	Recall	F1-score
KCFA	1	0.4437	0.6146
Presented approach using Wiki20 dataset	1	0.8652	0.8347

Presented approach using Wiki20 dataset has better results in terms of precision, recall and F1-score than KCFA. The Fig. 5 shows the performance metrics comparison using Wiki20 dataset.

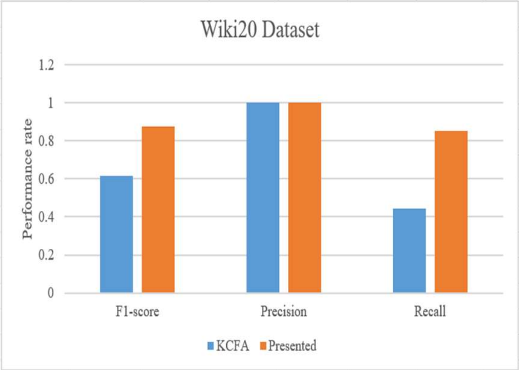


Fig. 5: Wiki20 dataset performance metrics comparison

Compared to KCFA, presented approach using Wiki20 dataset has better recall, better F1-score and equal precision. The Table 3 represents the performance evaluation on PubMed dataset.

Table 3: Performance Evaluation on PubMed dataset

Methods	Precision	Recall	F1-score
KCFA	1	0.3529	0.5217
Presented Context			

aware extraction of concepts from unstructured data using ML algorithms	1	0.8652	0.8347

The performance evaluation of presented approach on PubMed dataset has high recall, high f1-score than KCFA. The Fig. 6 shows the graphical representation.

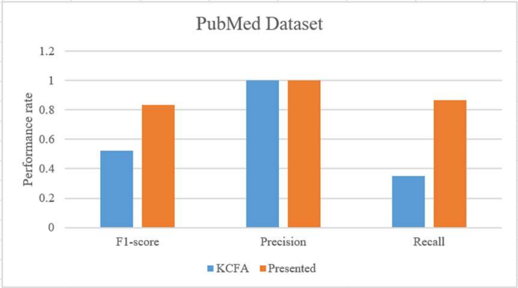


Fig. 6: PubMed dataset performance metrics comparison

The table 4 indicates the performance of KCFA and Presented approach using different datasets like Nguyen2007, Theses100 and Krapivin2009.

Table 4: F1-Score comparison

Methods	**Nguyen2007**	**Theses100**	**Krapivin 2009**
KCFA	0.8895	0.6233	0.8937
Presented	0.9247	0.8546	0.9262

Presented approach using Krapivin2009 dataset has high F1-score. The performance comparison of three different datasets is shown in below Fig. 7.

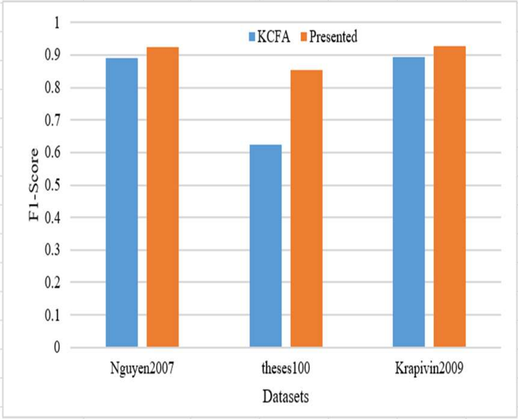


Fig. 7: Comparison Graph for F1-score

The table 5 represents the recall performance of KCFA and presented approaches using Nguyen2007, Theses100 and Krapivin2009 datasets.

Table 5: Recall Performance Comparison

Methods	**Nguyen2007**	**Theses100**	**Krapivin2009**
KCFA	0.801	0.4528	0.8078
Presented	0.9126	0.8612	0.8463

Presented approach has high recall using Nguyen2007 dataset while the KCFA approach has high recall using krapivin2009. The fig. 8 shows the graphical representation of recall performance comparison.

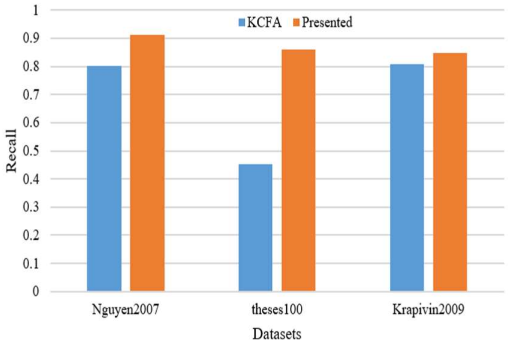


Fig. 8: Comparison Graph for Recall

Presented approach with Nguyen2007 dataset has high recall than KCFA with Nguyen2007 dataset. The theses100 and karipivin2009 datasets also has better recall for presented approach. The table 6 shows the Macro-averaged recall comparison between different datasets using different approaches.

Table 6: Comparison of Macro-averaged Recall

Datasets	**COBEC (1)**	**COBEC (2)**	**COBEC -T(1)**	**COBEC -T(2)**	**Presented**
SemEval	0.164	0.127	0.14	0.098	0.6964
DM	0.232	0.249	0.198	0.21	0.6722
Wiki20	0.125	0.138	0.122	0.135	0.706
OS	0.302	0.348	0.209	0.255	0.6444
DB	0.272	0.342	0.253	0.313	0.68888
Theses100	0.097	0.192	0.082	0.166	0.7325
Nguyen2007	0.204	0.257	0.191	0.238	0.6718

The theses100 dataset has high macro-averaged recall among other datasets (SemEval, DM, Wike20, OS, DB and Nguyen2007) for presented approach. The Fig. 9 shows the recall performance comparison of different approaches using different datasets.

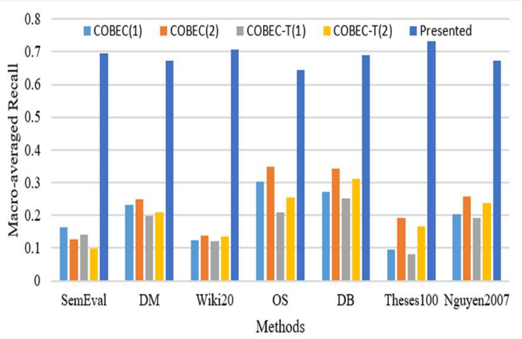


Fig. 9: Comparison graph for Macro-averaged Recall

The table 7 represents the macro-averaged precision performance evaluation.

Table 7: Comparison of Macro-averaged Precision

Datasets	COBEC (1)	COBEC (2)	COBEC -T(1)	COBEC -T(2)	Presented
SemEval	0.094	0.164	0.102	0.078	0.673
DM	0.373	0.393	0.32	0.333	0.645
Wiki20	0.273	0.306	0.266	0.3	0.772
OS	0.866	0.943	0.6	0.733	0.976
DB	0.265	0.314	0.248	0.285	0.667
Theses100	0.087	0.133	0.083	0.12	0.692
Nguyen2007	0.0264	0.234	0.243	0.213	0.647

Compared to different datasets, The OS dataset has high precision for presented approach. The Fig. 10 shows the graphical representation of Macro-averaged precision.

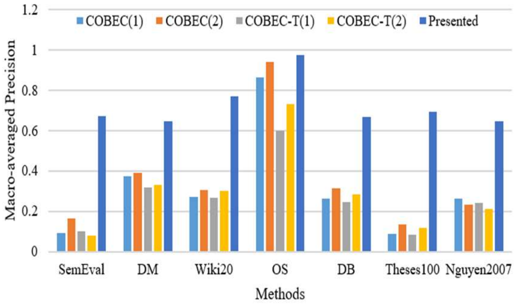


Fig. 10: Comparison graph for Macro-averaged Precision

The OS dataset has better precision for all approaches and in addition it has high precision for presented approach. The Table 8 compares the Macro-averaged F1-score of different datasets for different approaches.

Table 8: Comparison of Macro-averaged F1-score

Datasets	COBEC (1)	COBEC (2)	COBEC-T (1)	COBEC-T (2)	Presented
SemEval	0.124	0.132	0.094	0.1	0.7786
DM	0.288	0.307	0.242	0.254	0.734
Wiki20	0.174	0.197	0.166	0.184	0.6578
OS	0.448	0.51	0.31	0.38	0.7534
DB	0.263	0.321	0.244	0.293	0.7243
Theses100	0.0906	0.123	0.078	0.093	0.695
Nguyen2007	0.209	0.227	0.196	0.212	0.693

Among these datasets, the SemEval dataset has high macro-averaged F1-score for presented approach. The Fig. 11 shows the comparison graph for macro-averaged F1-score.

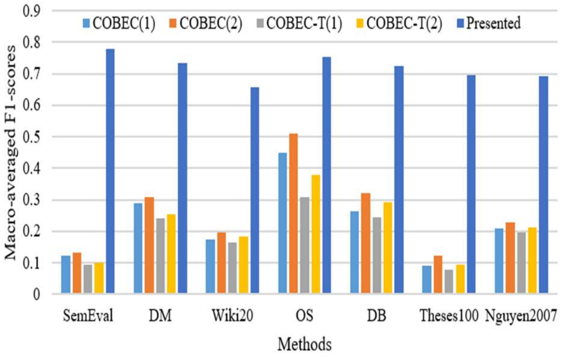


Fig. 11: Comparison graph for Macro-averaged F1-Score

The SemEval dataset has high Macro-averaged F1-score than other datasets. The Fig. 12 shows the ROC (Receiver Operating Characteristics) curve comparison between COBEC and Presented Context aware extraction of concepts from unstructured data using Machine Learning algorithms.

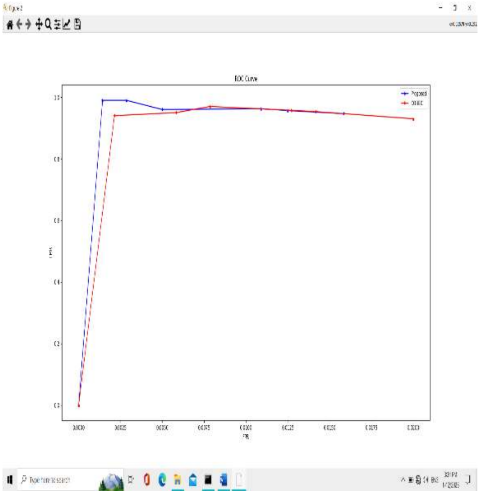


Fig. 12: ROC curve

In fig. 12, the red color curve line indicates the ROC of COBEC approach whereas Blue colour line indicates the ROC of presented approach. Compared to COBEC approach, presented Context aware extraction of concepts from unstructured data using Machine Learning algorithms has better ROC. The Fig. 3 shows the performance comparison regarding Accuracy, Kappa Coefficient, Sensitivity, Specificity and Macro F1-score.

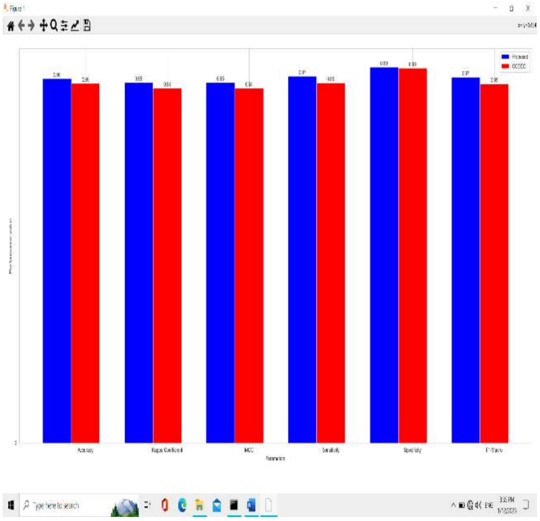


Fig. 13: Performance Comparison between COBEC and Presented Approach

In fig. 13 the blue colour indicates presented Context aware extraction of concepts from unstructured data using Machine Learning algorithms approach and red colour indicates COBEC Approach. The Context aware extraction of concepts from unstructured data using Machine Learning algorithms has high accuracy, Kappa Coefficient, Sensitivity, Specificity and Macro F1-score than COBEC Approach.

V. CONCLUSION

In this work, Context aware extraction of concepts from unstructured data using Machine Learning algorithms is presented. In this analysis, different datasets namely SemEval2010, Wiki20 and Pubmed datasets are used. In this approach, .phd files are taken as the input. The unstructured text documents are pre-processed to remove the unnecessary data and to clean the data. Textured Context Pattern (TCP) system with Lexical Subgroup (LS) method is used for finding the relevancy between feature of query document and from complete dataset. CCNN is used for finding matching feature for the classification of documents. Feature database creates the subfolder to store the classification results. This approach has classified different PHD documents and their domains where they belong to. The evaluation of presented method is measured in terms of Precision, Recall and F1-score. Different datasets are used to investigate the performance of presented approach. The performance of presented approach is compared with three datasets namely SemEval2010, Wiki20 and Pubmed datasets, however better results are achieved through SemEval 2010 dataset. In addition Macro-averaged of F1-Score, recall and precision are investigated with different methods. Nguyen2007 dataset has high recall, theses100 dataset has high macro-averaged recall and OS dataset has high macro-averaged precision.

VI. REFERENCES

[1] Mohammad Badrul Alam Miah, Suryanti Awang, Md Mustafizur Rahman, A. S. M. Sanwar Hosen and In-Ho Ra, “A New Unsupervised Technique to Analyze the Centroid and Frequency of Key phrases from Academic Articles”, Electronics 2022, 11, 2773, doi:10.3390/electronics11172773

[2] Abdullah Alqahtani, Habib Ullah Khan2, Shtwai Alsubai, Mohemmed Sha, Ahmad Almadhor, Tayyab Iqbal4 and Sidra Abbas, “An efficient approach for textual data classification using deep learning”, Frontiers in Computational Science, 2022, DOI 10.3389/fncom.2022.992296

[3] Dmitriy Dligach, Timothy Miller, Steven Bethard and Guergana Savova, “Exploring Text Representations for Generative Temporal Relation Extraction”, Proceedings of the 4th Clinical Natural Language Processing Workshop, pages 109 - 113 July 14, 2022, 2022 Association for Computational Linguistics

[4] Anna Glazkova and Dmitry Morozov, “Applying Transformer-based Text Summarization for Keyphrase Generation”, 2022, DOI:10.48550/arXiv.2209.03791

[5] Miah, M.B.A.; Awang, S.; Azad, M.S.; Rahman, M.M, “Keyphrases Concentrated Area Identification from Academic Articles as Feature of Keyphrase Extraction: A New Unsupervised Approach”, Int. J. Adv. Comput. Sci. Appl. 2022, 13, pp. 788-796.

[6] Menghan Zhang, “Applications of Deep Learning in News Text Classification”, Hindawi Scientific Programming Volume 2021, Article ID 6095354, 9 pages, doi:10.1155/2021/6095354

[7] Shubham Jain, Amy de Buitléir, Enda Fallon, “A Framework for Adaptive Deep Reinforcement Semantic Parsing of Unstructured Data”, 2021 International Conference on Information and Communication Technology Convergence (ICTC),DOI: 10.1109/ICTC52510.2021.9620904

[8] Anja Wilhelm and Wolfgang Ziegler, “Extending semantic context analysis using machine learning services to process unstructured data”, SHS Web of Conferences 102, 02001 (2021), doi:10.1051/shsconf/202110202001ELTC2021

[9] Nany Katamesh, Osama Abu-Elnasr and Samir Elmougy, “Deep Learning Multimodal for Unstructured and Semi-Structured Textual Documents Classification”, Computers, Materials & Continua,DOI:10.32604/cmc.2021.015761

[10] Dan Zhang, “Text Complexity Classification Data Mining Model Based on Dynamic Quantitative Relationship between Modality and English Context”, Hindawi Mathematical Problems in Engineering Volume 2021, Article ID 4805537, 10 pages, doi:10.1155/2021/4805537

[11] Dongdong Zhang, Changchang Yin, Jucheng Zeng, Xiaohui Yuan and Ping Zhang, “Combining structured and unstructured data for predictive models: a deep learning approach”, BMC Medical Informatics and Decision Making”, (2020) 20:280, doi:10.1186/s12911-020-01297-6

[12] Ahmed Ghozia, Gamal Attiya, Emad Adly and Nawal El-Fishawy, “Intelligence Is beyond Learning: A Context-Aware Artificial Intelligent System for Video Understanding”, Hindawi Computational Intelligence and Neuroscience Volume 2020, Article ID 8813089, 15 pages, doi:10.1155/2020/8813089

[13] Tushar Ghorpade, Bhavika Tuteja, Vaibhav Dholam, Gauri Patil, Ashutosh Bhujbal, “Learning of Unstructured Data Using Machine Learning Algorithm”, International Journal Of Information And Computing Science, 2019, ISSN NO: 0972-1347, Volume 6, Issue 4, April 2019

[14] Fouad Zablith, Ibrahim H. Osman, “Review Modus: Text Classification and Sentiment Prediction of Unstructured Reviews using a Hybrid Combination of Machine Learning and Evaluation Models”, Applied Mathematical Modelling (2019), doi:10.1016/j.apm.2019.02.032

[15] Sathya Madhusudhanan, Suresh Jaganathan and Jayashree L S, “Incremental Learning for Classification of Unstructured Data Using Extreme Learning Machine”, MDPI Journal Algorithms 2018, 11, 158; doi:10.3390/a11100158

[16] Mona Mowafy, A. Rezk, H. M. El-bakry, “Building Unstructured Crime Data Prediction Model”, International Journal of Computer Application (2250-1797) Issue 8 Volume 4, July-August 2018, doi:10.26808/rs.ca.i8v4.01

[17] Lipika Dey, Hardik Meisheri and Ishan Verma, “Predictive Analytics with Structured and Unstructured data - A Deep Learning based Approach”, IEEE Intelligent Informatics Bulletin December 2017 Vol.18 No.2

[18] Li Guo a, Feng Shi, Jun Tu, “Textual analysis and machine leaning: Crack unstructured data in finance and accounting”, The Journal of Finance and Data Science 2 (2016) 153e170, doi.org/10.1016/j.jfds.2017.02.001

[19] Shucheng Gong and Hongyan Liu, “Constructing Decision Trees for Unstructured Data”, International Conference on Advanced Data Mining and Applications, ADMA 2014: Advanced Data Mining and Applications pp 475–487

[20] Hassanin M. Al-Barhamtoshy and Fathy E. Eassa, “A data Analytics for unstructured Text”, Life Science Journal 2014;11(10).