Deciphering Complex Meanings from Unstructured Data: A Hybrid Approach

Dr. N S Patil[1], Dr Vinutha H P[2], Preethi B[3]

[1]Associate Professor, Department of Information Science and Engineering, Bapuji Institute of Engineering and Technology, Davangere,India.

[1]e-mail:patilbathi@gmail.com

[2]Professor, Department of Information Science and Engineering, Bapuji Institute of Engineering and Technology, Davangere, India.

[2]e-mail:vinuprasad.hp@gmail.com

[3]Department of Computer Science and Engineering, Bapuji Institute of Engineering and Technology Davanagere, India.

[3]e-mail:preethib027@gmail.com

ABSTRACT

In the modern digital era, unstructured text comprises a significant portion of computer-generated data. Extracting meaningful insights and deciphering complex semantics from such data is both a challenging task and a valuable opportunity due to the inherent ambiguity of natural language. This paper introduces Automated Analysis of Unstructured Text using Machine Learning (AAUT-ML), a hybrid framework that integrates Natural Language Processing (NLP) and Convolutional Neural Networks (CNN) to identify intricate semantic patterns within unstructured text. This approach enables users to extract formal semantic knowledge from large text corpora efficiently. To assess the effectiveness of AAUT-ML, we conducted experiments using three datasets: Data Mining (DM), Operating System (OS), and Database (DB). Our model was compared against existing techniques, including YAKE, Term Frequency-Inverse Document Frequency (TF-IDF), and Text-R. The results indicate significant improvements in precision, recall, and macro-averaged F1-score, highlighting the superior performance of our approach. This work presents a novel method for uncovering deep semantic structures in unstructured text, contributing to advancements in automated text analysis and knowledge extraction.

1. INTRODUCTION

Due to its unclear structure and absence of a specified data model, unstructured text presents challenges for computer systems [1]. Its lack of a schema and ambiguous structure render it inappropriate for traditional database models, leading to problems with indexing, management, and storage. As a result, the lack of predefined criteria makes search results less accurate. The amount of varied unstructured text produced by sources such as web pages, research papers, and articles is growing in today's digital environment [2]. Developments in web technology and text extraction tools are the main drivers of this expansion [3]. Although text mining tools and semantic technologies help connect text to knowledge, it is still difficult to parse complicated language and derive sophisticated insights [4]. By locating references and connections between things, information extraction (IE)

algorithms aid in the extraction of knowledge [5]. However, convolutional neural networks (CNNs) and other natural language processing (NLP) methods are necessary for deeper insights.They are frequently employed to efficiently extract unstructured information from text in image processing and text classification [6].

CNNs are used in NLP to analyze data using sliding windows in categorization challenges. Text ConvoNet, a CNN-based method for categorizing multi- and binary-class text classes, was introduced by Soni et al. [7]. This work has classified the text using intra-sentence n-gram characteristics. A TextCNN method for categorizing the text corpus was introduced by Song et al. [8]. A CNN-based model for text classification utilizing news text was presented in study [9]. Fesseha et al. [10] evaluated the text from news using CNN with continuous bag-of-words, FastText, and word to vector (Word2Vec). For the purpose of classifying patient summary notes, Lu et al. [11] have proposed a model that makes use of CNN, bidirectional-encoder-representation using transformers (BERT), transformer encoder, and four neural networks: recurrent neural network (RNN), long-short term memory (LSTM), gated-recurrent unit (GRU), and bidirectional LSTM (Bi-LSTM). For text classification, Zulqarnain et al. [12] looked at three architectures: CNN, RNN, and deep belief neural (DBN). Although all of these models used CNN and produced superior results, they were unable to classify the complicated semantics in unstructured data.

Furthermore, complex semantics in unstructured text texts can be found and examined through the use of NLP and CNN approaches. Important components of the semantic analysis include hyponymy, which links generic concepts (hypernyms) to their instances (hyponyms), such as "color" and its variants. Words that have the same spelling but distinct meanings are said to be homophones, such as "bat." Polysemy refers to the use of terms with different but related meanings, as "bank." Words with similar meanings are referred to be synonyms, such as "author/writer." The symmetrical interactions between opposing words are symbolized by Antonymy. Chatbots, support systems, sentiment analysis, search engines, translation, Q&A, and grammatical recognition are all areas where semantic analysis finds use. Using this information, the paper suggests automated analysis of unstructured text using machine learning (AAUT-ML), a hybrid NLP and CNN technique that is assessed by recall, precision, and F1-scores and may identify complex semantics in unstructured data. The contributions of this work are as follows: i) Use the NLP method to classify unstructured text into their respective domains; ii) Employ 1-dimensional convolving filters with n-gram detectors, each of which focuses on a particular family of closely related n-grams; iii) Use max-pooling over time to extract the relevant n-grams for decision-making; and iv) The rest of the network extracts hidden or complex semantics from unstructured text based on data from Max-pooling.

The following will be the outline for this paper. The methods that are now in use are examined in Section 2. The suggested methodology has been introduced in Section 3. The methodology that has been described is assessed and contrasted with previous research in Section 4. Lastly, the conclusion and next steps for the entire project have been outlined in Section 5.

2. LITERATURE SURVEY

The various reviews, approaches, architectures, and techniques utilized to categorize the material are shown in this section. The limitations of information extraction (IE) across all tasks and data types have not been thoroughly investigated in a single study. Therefore, by offering a thorough literature review of state-of-the-art techniques for managing various types of massive data and texts, Adnan and Akbar [13] get beyond that obstacle. There is also a

brief discussion and highlight of current IE concerns. Along with recommendations for further research in the subject of text IE, solutions are offered. The study is significant given the methods used today and the challenges associated with analyzing vast volumes of data. Overall, the results and recommendations presented here have improved the efficiency of the analysis of vast volumes of data. Using patient radiography information, Gupta et al. [14] reported a method using unsupervised machine learning to extract the relations. They have taken a hybrid approach to this task, using dependency-based parsed trees to extract the IE. When tested on patient mammography data, this study obtained an F-score of 94%. A new supervised machine learning model was presented by Chang and Mostafa [15], who evaluated it using the systematized nomenclature of medicine-clinical terminology (SNOMED CT) [16]. The 2018 N2C2 model was compared to this work [17]. When compared to [17], the suggested model had a score of 0.933. Adnan and Akbar [18] used unstructured data to survey text IE. Initially, it offers a comprehensive overview of IE techniques for various kinds of unstructured data, such as written, visual, audio, and video content. Additionally, it examines how the previously described proven IE methodologies are challenged by the diversity, dimensionality, and volume of unstructured big data. Additionally, potential methods for improving the unstructured huge data IE platforms for further research are presented. A model called SemIndex+ was presented by Tekli et al. [19] for the classification of partially structured, structured, and unstructured data. To categorize the text, they employed weight functions. When compared to the previous works, the SemIndex+ produced more accurate results.

. From the above study, it can be seen that very little work has been done on classifying the unstructured data directly, without data cleaning and data pre-processing. In addition, comprehensive research is offered in [13], which discusses methods for data extraction from unstructured data. Furthermore, [14]–[17] none of these papers have dealt with the issue of data extraction from unstructured sources. Both [18], [19] discuss methods for extracting information from unstructured sources, but neither paper examines these methods in relation to other datasets. The models [18], [19] may work properly for the given respective datasets given in [18], [19]. Hence in this work, we utilize the NLP and CNN and build a model called AAUT- ML to extract the text from the unstructured data. This work will classify the unstructured complex semantics into their respective domains using the NLP method. Also, by using n-gram detectors, 1-dimensional convolving filters are employed, each of which focuses on a certain family of closely related n-grams. The appropriate n-grams are extracted for decision-making through max-pooling over time. Based on data from Max-pooling, the rest of the network extracts hidden or complex semantics from unstructured text.

3. METHODOLOGY FOR SENSING COMPLICATED MEANINGS FROM UNSTRUCTURED TEXT FROM UNSTRUCTURED DATA USING NLP AND CNN TECHNIQUE

We are attempting to use CNN and NLP techniques to extract complicated semantics from unstructured text in this two-phase planned endeavor. We pre-process, filter, and categorize unstructured text according to particular data domains in the first phase (NLP). We attempt to understand how CNN processes text in the second phase (CNN), and then we apply that understanding to uncover intricate semantics in unstructured text. Satisfying the following goals is the primary objective:

a. To use the NLP approach to categorize unstructured text into the appropriate domains.

b. 1-dimensional convolving filters, each of which focuses on a particular family of closely related n-grams,

are used as n-gram detectors.

c. The appropriate n-grams are extracted for decision-making through max-pooling over time.

d. Based on data from Max-pooling, the rest of the network extracts hidden or complex semantics from unstructured text.

Sample input datasets from computer science fields like databases, operating systems, and data mining unstructured text files in the.txt format are being used in this experiment design. The pre-processing step requires that the input unstructured documents be subjected to tokenization, stop word removal, and rare word removal functions. Pre-processing removes inconsistent or missing data values caused by technical or human error. Pre-processing can improve the precision, dependability, and consistency of a dataset's quality and accuracy. There will be two main phases to the proposed work. The following describes the specific steps for both Phase 1 and Phase 2. A thorough block diagram of the newly created architecture is shown in Figure 1.

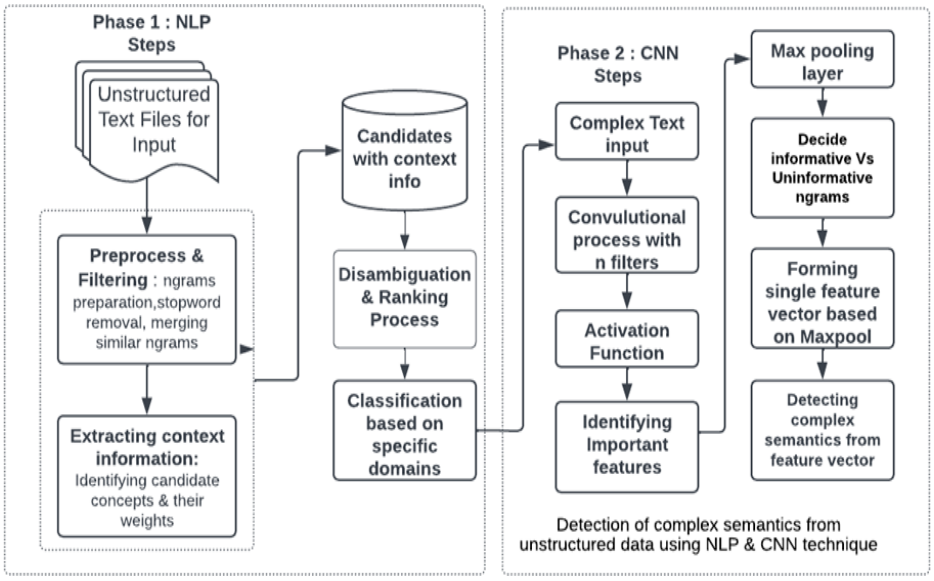


Figure 1. Novel developed block-diagram for NLP and CNN for Text classification and detection of complex semantics

Phase 1. Pre-processing and filtering candidate concepts

Step 1: Given a collection of unstructured text files, denoted as $D = \{d_1, d_2, d_3, \dots d_m\}$, describing a collection of concepts denoted as $C = \{c_1, c_2, c_3, \dots . c_m\}$, where both n and m are greater than 1, the objective is to identify and categorize the most pertinent concepts from C. The initial step involves tokenizing the unstructured textual documents, represented as $d_i \in D$, into n-grams that serve as the preliminary candidate concepts, denoted as $c_i \in C$.

$$d_i \in D > Tokenization\{c_1, c_2, c_3, \dots, c_m\}$$

Step 2: In these n-grams, common stop words are eliminated based on a predefined stop-word list, and the occurrences of the remaining candidates c_i in D are tallied to generate a set of tuples denoted as $s = \{(c_i, f_i), \ldots .\}$, comprising each candidate c_i and its corresponding frequency f_i.

Step 3: To diminish noise, n-grams with low frequency, short length, and infrequent occurrences are eliminated from both the set of n-grams (T) and the concept set (C) by applying a specified frequency threshold f_t.

Step 4: Only meaningful unigrams ($n >= 1$) are retained only if it occurs at-least f_t times more frequently than any larger n-gram containing the same unigram within it (refer to Figure 1).

Step 5: In cases where multiple higher-order n-grams c_j encompass c_i, f_i that makes reference to the highest frequently encountered c_j, then the remaining n-grams in C are merged based on two rules: firstly, plural tokens are filtered out if their singular form is also present as shown in Figure 1, and secondly, current participle of a normal verb is not used if there is an alternative form that does not include it.

Step 6: Among the remaining candidates in C, those without a corresponding $DBpedia$ entry is filtered out.

Step 7: Initial context information denoted as C_{info}, is generated, and the documents d_i are classified using specific-domain.

Phase 2. Sensing complicated meanings

Following is how CNN works for text processing. The three-layer CNN framework is used in this work's implementation. CNN's fundamental capabilities are analogous to those found in the visual-cortex in the brains of animals. CNN performs admirably in text-classification tasks. Classifying texts follows a process similar to those of categorizing images, with the exception that words are represented by vectors within a matrix rather than pixels.

3.1. Target-function

Target-function implementation makes advantage of learning-capable neuron weights and biases. Neurons receive a number of inputs, perform a weighted average over those inputs, and then transmit that value through an activation mechanism to produce a consequence. A loss-function for the entire system can typically be found by passing the network's result through the softmax layer. The output of a fully connected network with a softmax layer is down-sampled..

3.2. Representation

Word-indices are converted into three-dimensional vectors by CNN's embedding level, which is its first level. The lookup-table equivalent is used to find these vectors. Each word is translated using its corresponding embedding when a sentence is represented as N and words are represented as W. The vocabulary-size is then calculated using the largest sentence size, $Vwrd$. Following their conversion to vectors, all of the words are passed through the convolution layer.

3.3. System structure

The developed architecture consists of three major stages. The architecture is made up of two layers: the Embedding level, which translates words to embedded vectors, and the Convolution level, which does the majority of the approach work. A set of preset filters are

applied to the sentence matrix, reducing its size. The fourth level, softmax, operates as a downsampling level, reducing the sentence matrix while also computing the loss function. The embedded word lookup table can be used to determine the sentence's word embedding. To ensure that each sentence is treated fairly, the matrix generated by the embedding component is kept padded. Once the filters have been constructed, the matrix will be decreased further, and convolved features will be formed.

Embedded sentences so that the resulting sentence matrix possesses an identical shape and size. Word vectors
$w_1, \dots, w_n \in R^d$ are the result of embedding every symbol in the n-words text being entered in the form of d-dimensional vector data. The generated $d \times n$ matrix can be utilized to transmit a sliding-window across the text within a convolutional level. In accordance with each l-word n-gram.

$$u_i = [w_i, \dots, w_{i+l-1}] \in R^{d \times l} ; 0 \leq i \leq n - l \quad (1)$$

where matrix $F \in R^{n \times m}$. Max-pooling applied along the n-gram dimension yields $p \in R^m$. The non-linearity of rectified linear unit (ReLU) is used to process R^m. The distribution across the classes used for classification is then generated by a linearly fully connected layer $W \in R^{c \times m}$, that then outputs the class with the highest strength. In execution, we employ a range of window widths, from $l \in L, L \in N$, by chaining together the outputs p^l vectors of numerous convolution levels. It is important to take into account that the procedures described here also work for dilated convolutions. This is represented as (2) to (6).

$$u_i = [w_i, \dots, w_{i+l-1}] \in R^{d \times l} ; 0 \leq i \leq n - l \quad (2)$$

$$u_i = [w_i, \dots, w_{i+l-1}] \quad (3)$$

$$F_{ij} = \langle u_i f_i \rangle \quad (4)$$

$$p_j = ReLU(maxF_{ij}) \quad (5)$$

$$o = softmax(W_p) \quad (6)$$

3.4. Identification of important features

It is commonly believed that filters can be compared to n-gram detectors, where each filter searches for a distinct category of n-grams and assigns high scores to them. Only the n-grams with the best scores are left after the max-pooling procedure. Conclusions can be made after determining the total number of n-grams in the max pooled vector, which is represented by the collection of matching filters. For text categorization, any filter's high-scoring n-grams (in contrast to how it ranks comparable n-grams) should be regarded as very helpful. We broaden this viewpoint in this subsection by raising and attempting to address the following questions: what information about n-grams can be found in the max-pooled vector, and how is it applied in the final classification?

3.5. Informative vs. uninformative n-grams

The classification method is based on the pooling vector p, which is a part of the m-dimensional real space Rm, i.e., $p \in Rm$. The ReLU applied on the greatest inner product between the n-gram ui and the filter fj yields each value of pj. These numbers may be

explained by the particular n-gram ui, which activated the filter and is composed of the words [wi,…,wi+l−1]. The group of n-grams that make up the total probability distribution p is represented by the symbol Sp. The classifier's decision-making process cannot be impacted by n-grams that are absent from the collection Sp. Nonetheless, the existence of n-grams in the set Sp must be taken into account. Prior research on the prediction-based analysis of CNN for text has concentrated on locating the n-grams (represented by the symbols Sp) in the input sequence and assessing their relative rankings in order to comprehend the underlying prediction process. We take on a more complex viewpoint in this situation. It is crucial to emphasize that the final classification procedure assesses these people using the rankings determined by the filters rather than necessarily taking into account the particular n-gram IDs. As a result, it is essential that the information in variable p depend on the assigned rankings. From a conceptual perspective, there are two different sorts of n-grams in the set Sp: accidental and deliberate. Because of their higher scores given by the filtering system, Sp contains intentional n-grams. This implies that these n-grams include important data that is pertinent to the final decision-making procedure. On the other hand, it is noted that unintentional n-grams, even if they have a low ranking, are able to enter the set Sp. This phenomenon can be explained by the lack of a second n-gram that outperformed them in terms of ranking. It is clear from the study that the n-grams in question don't seem to have any informational value in respect to the categorization choice that is being made. Are deliberate and inadvertent n-grams able to be distinguished from one another? It is assumed that there is a measurable threshold in the framework for each filter. While numbers below this threshold are considered erroneous and are therefore ignored for classification purposes, values above this barrier are suggestive of valuable information relevant to the classification process.

Formally, the threshold dataset(X,Y) is provided

$$\text{Purity(f.t)}= \frac{|\{(x,y) \in (X,Y)_f \mid x \geq t \ \& \ y=true\}|}{|\{(x,y) \in (X,Y)_f \mid x \geq t\}|} \quad (7)$$

Our empirical results lead us to conclude that, when determining the threshold of a particular filter, an optimum purity-value of 0.75 is desirable. Additionally, the outcomes of the suggested work have been assessed in terms of macro averaged F1-score, recall, and precision using three datasets. The following section discusses the findings.

4.RESULTS AND DISCUSSION

This part begins with a comprehensive discussion of the system requirements, followed by a thorough assessment of the dataset used in the study. Furthermore, the performance measures used to assess the system's efficacy are thoroughly reviewed. The suggested methodology's results were compared to those of earlier studies, specifically in terms of recall, precision, and macro-averaged F1-score. The inclusion of a complete discussion section in the present study allows for a thorough analysis and interpretation of the acquired data.

4.1. System requirements, datasets and performance metrics

In this section, we carry out a series of experiments in Phases 1 and 2. The code was written in Python and executed on a system running Windows 10, with 16 GB of RAM. This approach uses three different datasets for experimentation: data mining (DM) [20], operating

system (OS) [21], and data base (DB) [22], [23]. The DM, OS, and database were created in [24]. The performance of the current approach, AAUT-ML, is compared to several known approaches, including YAKE [15], TF-IDF [25], and TextR [26]. All results and datasets were obtained from [24]. Results from the given AAUT-ML are examined, and measures like as recall, precision, and F1-score are used to assess the model's performance. By dividing the sum of all anticipated sequences within a positive category by the precision, we can determine how many positive categories were properly predicted. (9) can calculate recall, which is defined as the proportion of accurately anticipated positive events compared to actual positive outcomes. In machine learning, the F1 score is an important evaluation metric. It elegantly describes a model's prediction ability by combining the accuracy and recall measurements, which are commonly measured using (10).

$$Precision = \frac{Key_{corrected}}{Key_{predicted}} \quad (8)$$

$$Recall = \frac{Key_{corrected}}{Key_{predicted}} \quad (9)$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (10)$$

where $Key_{corrected}$ is the sum of all the predicted key-phrases that were found to be a good match with the standard key-phrases, and $Key_{predicted}$ is the sum of all the predicted key-phrases from the document.

4.2. Precision

The precision has been assessed and contrasted with the Text-R, YAKE, and TF-IDF models in Figure 2. The AAUT-ML outperformed YAKE, TF-IDF, and TextR for the DM dataset by 66.21%, 87.66%, and 98.65%, respectively. AAUT-ML outperformed YAKE, TF-IDF, and TextR by 66.74%, 84.64%, and 97.57%, respectively, on the OS dataset. The AAUT-ML outperformed YAKE, TF-IDF, and TextR for the DB dataset by 33.20%, 84.52%, and 97.35%, respectively. When compared to the YAKE, TF-IDF, and TextR, the suggested AAUT-ML has produced greater precision results.

4.3. Recall

The recall score has been assessed and contrasted with the Text-R, YAKE, and TF-IDF models in Figure 3. The AAUT-ML outperformed YAKE, TF-IDF, and TextR by 48.17%, 81.70%, and 96.34%, respectively, on the DM dataset. The AAUT-ML outperformed YAKE, TF-IDF, and TextR by 62.91%, 51.65%, and 3.97%, respectively, on the OS dataset. Regarding the DB dataset, In comparison to YAKE, TF-IDF, and TextR, the AAUT-ML outperformed them by 28.67%, 80.51%, and 96.69%, respectively. When compared to the YAKE, TF-IDF, and TextR, the suggested AAUT-ML has produced superior recall outcomes.

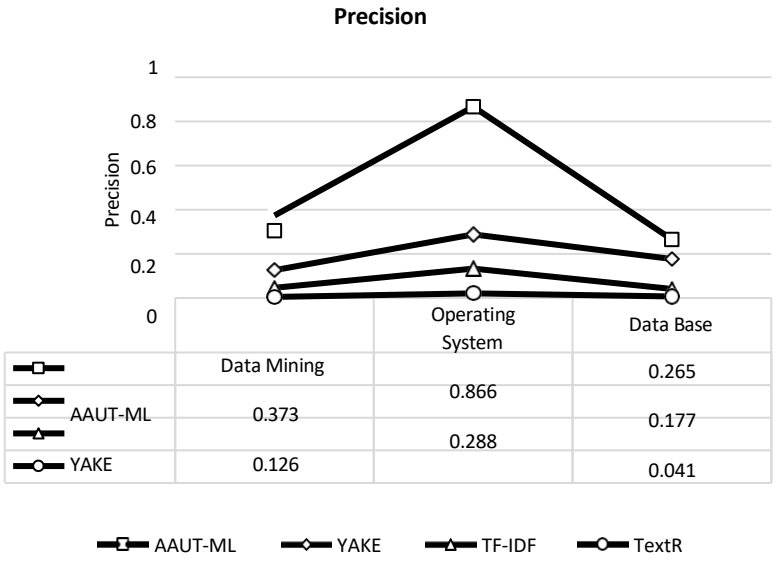
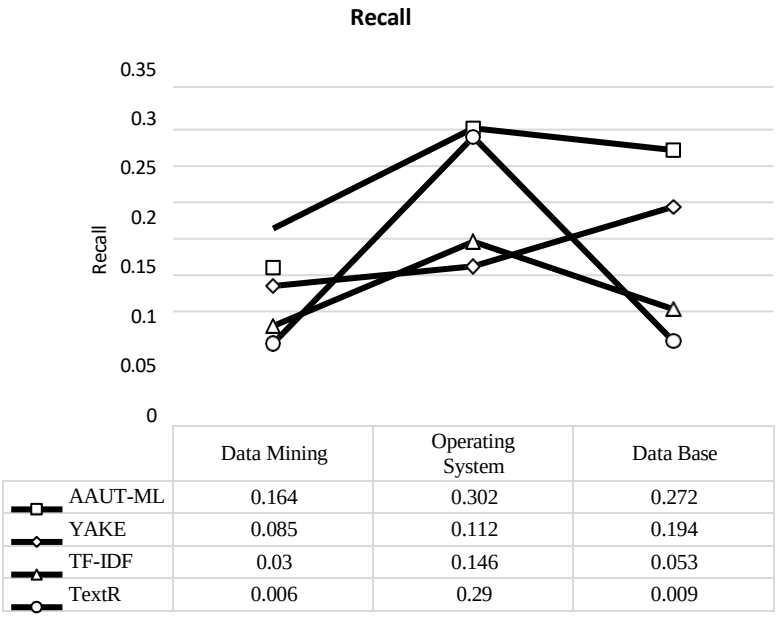


Figure 2. Macro-averaged precision scores of AAUT-ML versus three baseline methods on three different datasets



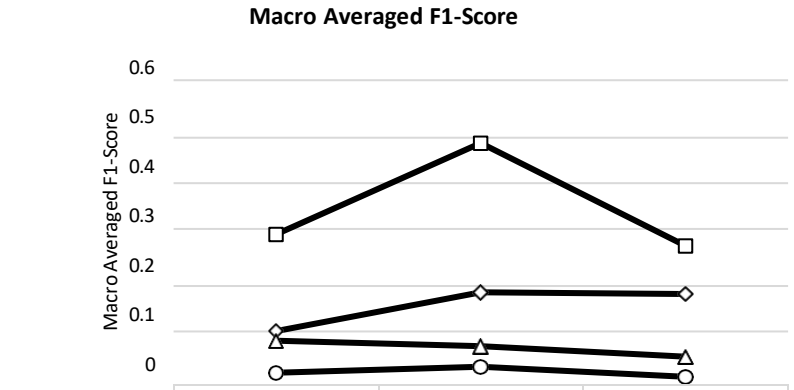
	Data Mining	Operating System	Data Base
AAUT-ML	0.164	0.302	0.272
YAKE	0.085	0.112	0.194
TF-IDF	0.03	0.146	0.053
TextR	0.006	0.29	0.009

Figure 3. Macro-averaged Recall scores of AAUT-ML versus three baseline methods on three different datasets

4.4. Macro-averaged F1-score

The macro averaged F1-score has been assessed and contrasted with the Text-R, YAKE, and TF IDF models in Figure 4. The AAUT-ML outperformed YAKE, TF-IDF, and TextR by 64.93%, 72.22%, and 96.52%, respectively, on the DM dataset. The AAUT-ML for the OS dataset performed 61.88%, 86.06%, and 95.28% better than YAKE, TF-IDF, and TextR, respectively. The AAUT-ML outperformed YAKE, TF-IDF, and TextR for the DB

dataset by 30.79%, 82.88%, and 99.61%, respectively. Compared to the YAKE, TF-IDF, and TextR, the suggested AAUT-ML has produced superior results for the macro averaged F1-Score.



	Data Mining	Operating System	Data Base
AAUT-ML	0.288	0.488	0.263
YAKE	0.101	0.186	0.182
TF-IDF	0.08	0.068	0.045
TextR	0.01	0.023	0.001

Figure 4. Macro-averaged F1-scores of AAUT-ML versus three baseline methods on three different datasets

4.5 Discussion

Table 1 displays the macro averaged F1-score, recall, and accuracy values for three distinct datasets (data mining, operating system, and database) using four distinct approaches (AAUT-ML, YAKE, TF-IDF, and TextR). It appears that AAUT-ML performs the best for the "operating system" dataset out of all metrics, showing good recall, accuracy, and F1-score values. Additionally, AAUT-ML outperforms other techniques in the "data mining" dataset. Lastly, YAKE and TF-IDF show comparable F1-scores, recall, and accuracy values for the "data base" dataset, whereas AAUT-ML performs marginally better. Additionally, this effort attempted to use the fusedmax and sparemax, but neither of these were able to outperform the softmax.

Table 1. Comparative study

	Macro Averaged F1-Score				Recall				Precision			
	AAUT-ML	YAKE	TF-IDF	TextR	AAUT-ML	YAKE	TF-IDF	TextR	AAUT-ML	YAKE	TF-IDF	TextR
Data mining	0.288	0.101	0.08	0.01	0.164	0.085	0.03	0.006	0.373	0.126	0.046	0.005
Operating system	0.488	0.186	0.068	0.023	0.302	0.112	0.146	0.29	0.866	0.288	0.133	0.021
Data base	0.263	0.182	0.045	0.001	0.272	0.194	0.053	0.009	0.265	0.177	0.041	0.007

5.CONCLUSION

This article presents the use of convolution neural networks and natural language processing techniques to extract complicated meanings from unstructured data. Several datasets, including database, data mining, and operating systems datasets, are employed in this investigation. Certain widely held beliefs about how CNNs interpret and categorize text have been called into question by our research. First, we have shown that max-pooling over

time creates a thresholding impact on the convolution layer's output, thereby differentiating between characteristics that are pertinent and those that are not for the final classification.
By linking each filter to the class it belongs to, this realization enabled us to determine the essential n-grams for classification. Additionally, we have indicated situations in which filters give particular word activations negative values. resulting in poor scores for n-grams that contain them, even though the words are otherwise very stimulating. The interpretability of CNNs for text classification is improved by these discoveries. Our method classifies different documents and their corresponding domains in an efficient manner. Using criteria like precision, recall, and F1-score across several datasets, we assessed its performance and found that it outperformed other approaches. When compared to previous works, the performance of the proposed work demonstrates superior outcomes. This technique can be applied to the classification of corpus semantics in subsequent research. in organized data. The data can be structured using a variety of feature extraction techniques. Additionally, machine learning can be applied in conjunction with NLP.

REFERENCES

[1] *E. Camilleri and S. J. Miah, "Evaluating latent content within unstructured text: an analytical methodology based on a temporal network of associated topics," Journal of Big Data, vol. 8, no. 1, Sep. 2021, doi: 10.1186/s40537-021-00511-0.*

[2] *D. Antons, E. Grünwald, P. Cichy, and T. O. Salge, "The application of text mining methods in innovation research: current state, evolution patterns, and development priorities," R&D Management, vol. 50, no. 3, pp. 329–351, 2020, doi: 10.1111/radm.12408.*

[3] *S. Sulova, "A conceptual framework for the technological advancement of e-commerce applications," Businesses, vol. 3, no. 1, pp. 220–230, Mar. 2023, doi: 10.3390/businesses3010015.*

[4] *D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: state of the art, current trends and challenges," Multimedia Tools and Applications, vol. 82, no. 3, pp. 3713–3744, Jul. 2023, doi: 10.1007/s11042-022-13428-4.*

[5] *M. Y. Landolsi, L. Hlaoua, and L. Ben Romdhane, "Information extraction from electronic medical documents: state of the art and future research directions," Knowledge and Information Systems, vol. 65, no. 2, pp. 463–516, Nov. 2023, doi: 10.1007/s10115-022-01779-1.*

[6] *A. K. Sharma, S. Chaurasia, and D. K. Srivastava, "Sentimental short sentences classification by using CNN deep learning model with fine tuned Word2Vec," Procedia Computer Science, vol. 167, pp. 1139–1147, 2020, doi: 10.1016/j.procs.2020.03.416.*

[7] *S. Soni, S. S. Chouhan, and S. S. Rathore, "TextConvoNet: a convolutional neural network based architecture for text classification," Applied Intelligence, vol. 53, no. 11, pp. 14249–14268, Oct. 2023, doi: 10.1007/s10489-022-04221-9.*

[8] *P. Song, C. Geng, and Z. Li, "Research on text classification based on convolutional neural network," in Proceedings - 2nd International Conference on Computer Network, Electronic and Automation, ICCNEA 2019, IEEE, Sep. 2019, pp. 229–232. doi: 10.1109/ICCNEA.2019.00052.*

[9] *Y. Zhu, "Research on news text classification based on deep learning convolutional neural network," Wireless Communications and Mobile Computing, vol. 2021, pp. 1–6, Dec. 2021, doi: 10.1155/2021/1508150.*

[10] *A. Fesseha, S. Xiong, E. D. Emiru, M. Diallo, and A. Dahou, "Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya," Information (Switzerland), vol. 12, no. 2, pp. 1–17, Jan. 2021, doi: 10.3390/info12020052.*

[11] *H. Lu, L. Ehwerhemuepha, and C. Rakovski, "A comparative study on deep learning models for text classification of unstructured medical notes with various levels of class imbalance," BMC Medical Research Methodology, vol. 22, no. 1, Jul. 2022, doi: 10.1186/s12874-022-01665-y.*

[12] *M. Zulqarnain, R. Ghazali, Y. M. M. Hassim, and M. Rehan, "A comparative review on deep learning models for text classification," Indonesian Journal of Electrical Engineering and Computer Science (IJEECS), vol. 19, no. 1, pp. 325–335, Jul. 2020, doi: 10.11591/ijeecs.v19.i1.pp325-335.*

[13] *K. Adnan and R. Akbar, "An analytical study of information extraction from unstructured and multidimensional big data," Journal of Big Data, vol. 6, no. 1, Oct. 2019, doi: 10.1186/s40537-019-0254-8.*

[14] *A. Gupta, I. Banerjee, and D. L. Rubin, "Automatic information extraction from unstructured mammography reports using distributed semantics," Journal of Biomedical Informatics, vol. 78, pp. 78–86, Feb. 2018, doi: 10.1016/j.jbi.2017.12.016.*

[15] *E. Chang and J. Mostafa, "Cohort identification from free-text clinical notes using SNOMED CT's hierarchical semantic relations," AMIA ... Annual Symposium proceedings. AMIA Symposium, vol. 2022, pp. 349–358, 2022.*

[16] *E. Chang and J. Mostafa, "The use of SNOMED CT, 2013-2020: A literature review," Journal of the American Medical Informatics Association, vol. 28, no. 9, pp. 2017–2026, Jun. 2021, doi: 10.1093/jamia/ocab084.*

[17] *S. Henry, K. Buchan, M. Filannino, A. Stubbs, and O. Uzuner, "2018 N2C2 Shared task on adverse drug events and medication extraction in electronic health records," Journal of the American Medical Informatics Association, vol. 27, no. 1, pp. 3–12, Oct. 2020, doi: 10.1093/jamia/ocz166.*

[18] *K. Adnan and R. Akbar, "Limitations of information extraction methods and techniques for heterogeneous unstructured big data," International Journal of Engineering Business Management, vol. 11, Art. no. 184797901989077, Jan. 2019, doi: 10.1177/1847979019890771.*

[19] *J. Tekli, R. Chbeir, A. J. M. Traina, and C. Traina, "SemIndex+: A semantic indexing scheme for structured, unstructured, and partly structured data," Knowledge-Based Systems, vol. 164, pp. 378–403, Jan. 2019, doi: 10.1016/j.knosys.2018.11.010.*

[20] *C. C. Aggarwal, Data mining: The textbook. Synthesis Collection of Technology, 2015.*

[21] *R. Ramakrishnan and J.Gehrke, Database management systems. India, 2014.*

[22] *VOCW, "Operating systems," cnx.org. https://cnx.org/contents/epUq7msG@2.1:vLiqr17-@1/Process (accessed Aug. 10, 2023).*

[23] *OpenStax, "OpenStax | free textbooks online with no catch," cnx.org. http://cnx.org/content/col10785/1.2/ (accessed Aug. 10, 2023).*

[24] *S. Gul, S. Räbiger, and Y. Saygın, "Context-based extraction of concepts from unstructured textual documents," Information Sciences, vol. 588, pp. 248–264, Apr. 2022, doi: 10.1016/j.ins.2021.12.056.*

[25] *A. Bougouin, F. Boudin, and B. Daille, "TopicRank: Graph-based topic Ranking for keyphrase extraction," in 6th International Joint Conference on Natural Language Processing, IJCNLP 2013 - Proceedings of the Main Conference, Nagoya, Japan: Asian Federation of Natural Language Processing, Oct. 2013, pp. 543–551. [Online]. Available: https://aclanthology.org/I13-1062*

[26] *F. Boudin, "Unsupervised keyphrase extraction with multipartite graphs," in NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, Association for Computational Linguistics, 2018, pp. 667–672. doi: 10.18653/v1/n18-210*