International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 1, February 2024, pp. 711~720
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i1.pp711-720

Sensing complicated meanings from unstructured data: a novel hybrid approach

Shankarayya Shastri[1], Veeragangadhara Swamy Teligi Math[2], Patil Nagaraja Siddalingappa[3]
[1]Department of Computer Science and Engineering, GM Institute of Technology Davangere, Visvesvaraya Technological University, Belagavi, India
[2]Department of Computer Science and Engineering, RYM College of Engineering Ballari, Visvesvaraya Technological University, Belagavi, India
[3]Department of Information Science and Engineering, Bapuji Institute of Engineering and Technology Davanagere, Visvesvaraya Technological University, Belagavi, India

Article Info

Article history:

Received Aug 11, 2023
Revised Sep 13, 2023
Accepted Sep 14, 2023

Keywords:

Convolutional neural network
Natural language processing
Information extraction
Unstructured text
Complex semantics

ABSTRACT

The majority of data on computers nowadays is in the form of unstructured data and unstructured text. The inherent ambiguity of natural language makes it incredibly difficult but also highly profitable to find hidden information or comprehend complex semantics in unstructured text. In this paper, we present the combination of natural language processing (NLP) and convolution neural network (CNN) hybrid architecture called automated analysis of unstructured text using machine learning (AAUT-ML) for the detection of complex semantics from unstructured data that enables different users to make understand formal semantic knowledge to be extracted from an unstructured text corpus. The AAUT-ML has been evaluated using three datasets data mining (DM), operating system (OS), and data base (DB), and compared with the existing models, i.e., YAKE, term frequency-inverse document frequency (TF-IDF) and text-R. The results show better outcomes in terms of precision, recall, and macro-averaged F1-score. This work presents a novel method for identifying complex semantics using unstructured data.

Corresponding Author:

Shankarayya Shastri
Department of Computer Science and Engineering, GM Institute of Technology Davangere, Visvesvaraya Technological University
Belagavi, India
Email: shankarayyasresearch@gmail.com

1. INTRODUCTION

Unstructured text poses difficulties for computer programs due to its lack of a clear structure or defined data model [1]. This makes it unsuitable for conventional database models, causing issues in storage, management, and indexing due to the absence of a schema and unclear structure. Consequently, search results are less accurate due to the absence of predefined attributes. In today's digital landscape, the volume of diverse unstructured text is increasing, generated from sources like web pages, research papers, and articles [2]. This growth is driven by advancements in web technologies and text extraction tools [3]. While semantic technologies and text mining systems aid in linking text to knowledge, processing complex language and extracting intricate insights remains a challenge [4]. Information extraction (IE) algorithms help extract knowledge by identifying references and relationships between entities [5]. Yet, deeper insights require natural language processing (NLP) techniques, including convolutional neural networks (CNNs),

which are widely used in image processing and text classification, to effectively capture unstructured information from the text [6].

In NLP, CNNs are applied to categorization problems, using sliding windows for data analysis. Soni *et al.* [7] have presented a Text ConvoNet which is used for classifying the multi and binary-class text classes using CNN. This work has used intra-sentence n-gram characteristics for classifying the text. Song *et al.* [8] have presented a TextCNN technique for classifying the text corpus. In study [9], a model has been presented for classifying the text using news text using CNN. Fesseha *et al.* [10] used CNN with continuous bag-of-words, FastText, and word to vector (Word2Vec) for evaluating the text from news. Lu *et al.* [11] have proposed a model that uses 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 Bi-directional LSTM (Bi-LSTM) for classification of summary notes of the patients. Zulqarnain *et al.* [12] examined three architectures, CNN, RNN, and deep belief neural (DBN) for text classification. All these models have utilized CNN and achieved better results, but failed to classify the complex semantics when the data is unstructured.

Further, by using NLP and CNN techniques, complex semantics within unstructured text documents can be detected and analyzed. The semantic analysis involves important elements such as hyponymy, which connects generic terms (hypernyms) to their instances (hyponyms) for instance, "color" and its variations. Homonymy refers to words with the same spelling but different meanings, like "bat." Polysemy involves words with distinct yet related meanings, such as "bank." Synonymy pertains to words with similar meanings, like "author/writer." Antonymy represents symmetrical relationships between opposite words. Semantic analysis finds applications in chatbots, support systems, sentiment analysis, search engines, translation, Q&A, and grammar identification. This study uses this knowledge to propose automated analysis of unstructured text using machine learning (AAUT-ML), a hybrid NLP and CNN approach that can detect complicated semantics in unstructured data and is evaluated by measures of recall, precision, and F1-scores. This work makes the following contributions; i) Classify unstructured text into their respective domains using the NLP method, ii) Using n-gram detectors, 1-dimensional convolving filters will be employed, each of which focuses on a certain family of closely related n-grams, iii) The appropriate n-grams are extracted for decision-making through max-pooling over time, and iv) Based on data from Max-pooling, the rest of the network extracts hidden or complex semantics from unstructured text.

This paper will have the following outline. In Section 2, the existing approaches are discussed. In Section 3, the proposed methodology has been presented. In Section 4, the presented methodology is evaluated and compared with the existing works. Finally, in Section 5, the conclusion and future work of the complete work have been presented.

2. LITERATURE SURVEY

This section presents the different reviews, strategies, architectures, and methodologies used for classifying the text. There has not been a lot of effort done to examine the drawbacks of information extraction (IE) across all tasks and data kinds in a unified study. Hence, Adnan and Akbar [13], overcomes that barrier by providing a comprehensive literature evaluation of cutting-edge methods for handling all forms of big data and texts. Additionally, current issues with IE are highlighted and briefly discussed. Solutions are suggested, along with directions for future study in the field of text IE. In light of the current approaches and difficulties in the analysis of large amounts of data, the study is important. The findings and suggestions offered here have analyzed large amounts of data more efficiently overall. Gupta *et al.* [14] presented a work for extracting the relations using unsupervised machine learning by utilizing the radiology reports of patients. For implementing this work, they have used a hybrid approach, i.e., dependency-based parsed trees which will extract the IE. This work achieved an F-score of 94 percent when tested on mammography data of patients. Chang and Mostafa [15] introduced a novel supervised machine learning model and utilized the systematized nomenclature of medicine-clinical terms (SNOMED CT) [16] for evaluating their model. This work was compared with the 2018 N2C2 model [17]. The proposed model achieved a score of 0.933 when compared with [17]. Adnan and Akbar [18] has surveyed text IE using unstructured data. First, it provides a consolidated summary of IE methods across different types of unstructured data (including written content, visual content, audio content, and videos). Furthermore, it explores how the diversity, dimensions, and amount of unstructured large data pose challenges to the aforementioned established IE methods. Furthermore, possible approaches to enhance the unstructured large data IE platforms for future study are offered. Tekli *et al.* [19] introduced a model, SemIndex+ for classifying the partly structured, structured, and unstructured data. They have used weight functions for classifying the text. The SemIndex+ achieved better results in terms of precision when compared with the existing works.

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

In this two-phase proposed work, we are trying to detect complex semantics from unstructured text using NLP and CNN Techniques. In the first phase (NLP) we are pre-processing, filtering, and classifying unstructured text to specific data domains. In the second phase (CNN), we try to figure out how CNN handles text after which we use that knowledge to discover complex semantics in unstructured text. The main aim is to satisfy the objectives that are listed as follows:

a. To classify unstructured text into their respective domains using the NLP method.
b. As n-gram detectors, 1-dimensional convolving filters are employed, each of which focuses on a certain family of closely related n-grams.
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.

In this experiment setup, we are taking sample input datasets from computer science domains such as Databases, Operating systems, and Data mining unstructured text files in .txt format. Tokenization, stop word removal, and rare word removal functions must be applied to the input unstructured documents as part of the pre-processing step. Pre-processing eliminates missing or inconsistent data values brought on by human or technological faults. Pre-processing can make a dataset's accuracy and quality more precise, dependable, and consistent. The proposed work will be carried out in two distinct phases. The detailed procedures for both Phase 1 along Phase 2 are outlined below. Figure 1 provides a comprehensive block-diagram of the novel developed architecture.

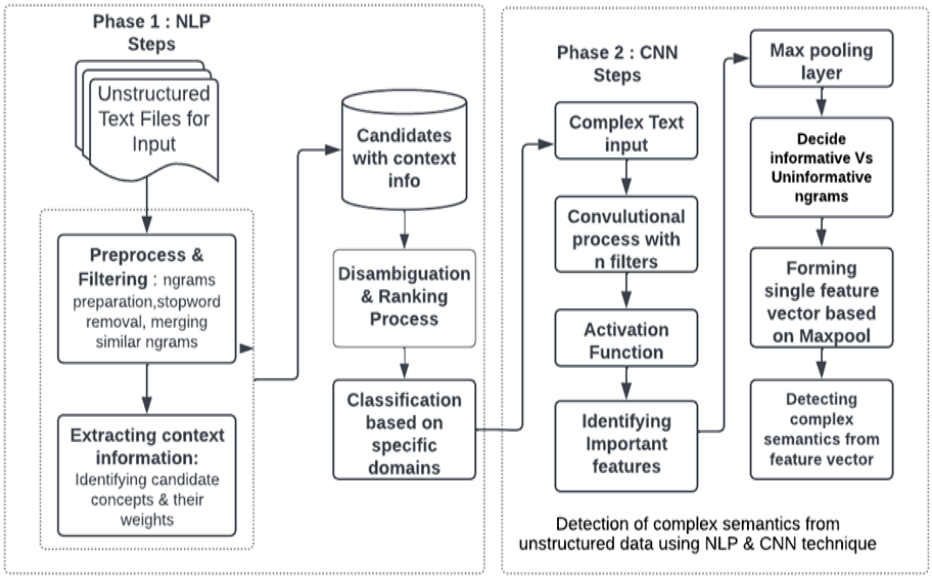


Figure 1. Novel developed block-diagram for NLP and CNN for Text classification and detection of complex semantics

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 $s = \{(c_i, f_i), \dots .\}$, 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

Learning-capable neuron biases and weights are used throughout target-function implementation. To generate an outcome, neurons receive multiple inputs, carry out a weighted average over those inputs, and then send that value through an activation mechanism. By sending the network's outcome through the softmax layer, a loss-function can usually be determined for the entire system. The outcome of a network with a softmax layer, which has full connectivity, is down-sampled.

3.2. Representation

CNN's initial level consists of an embedding level, which transforms word-indices into three-dimensional vectors. These vectors are discovered using the equivalent of a lookup-table. When considering a sentence represented as N, and words represented as W, each word is translated through its associated embedding and the highest sentence size V^{wrd} is used to determine the vocabulary-size. Once all the words have been converted to vectors, they are sent through the convolution-layer.

3.3. System structure

The developed architecture has three distinct stages. The architecture consists of two layers: the Embedding level, that maps words to embedded vectors, and then the Convolution level, that performs most part of the approach work. The sentence matrix is processed by a set of preset filters, which reduces its dimensions. The softmax level, the final one, acts as a downsampling level that can both reduce the sentence-matrix and compute the loss-function. The sentence's word-embedding can be obtained using the embedded word lookup-table. To ensure that every single sentence is handled fairly, the matrix produced by the embedding component remains padded. Once the filters have been established, the matrix continues to be reduced and convolved-features are going to be generated. These complex characteristics are finally simplified. As a next step in down-sampling, the resultant data from the convolved-features is distributed across the maximum pooling level. Various sized and shaped filtration are specified. Three, four, and five are the filter shapes employed by the suggested approach. Following that, padding is applied to each of the

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

According to conventional wisdom, filters can be thought of as n-gram detectors, with every filter looking for a unique category of n-grams and marking them with high-scores. After the max-pooling process, only the n-grams with the most favorable scores remain. Once the total number of n-grams within the max-pooled vector (which is represented through the collection of matching filters) has been determined, a conclusion can be drawn. Any filter's high-scoring n-grams (compared with the way it ranks similar n-grams) should be considered to be particularly useful for text classification. In this subsection, we enhance this perspective by posing and aiming to respond to the following issues: what data underlying n-grams can be obtained within the max-pooled vector, and in what manner is it utilized in the last classification?

3.5. Informative vs. uninformative n-grams

The pooled vector p, that belongs to the m-dimensional real space R^m, i.e., $p \in R^m$ serves as the foundation for the classification process. Every value of p_j is derived through the ReLU applied to the maximum inner product between the n-gram u_i along with the filter f_j. These values could be attributed to the specific n-gram u_i, which consists of a sequence of words $[w_i, \dots, w_{i+l-1}]$, which activated the filter. The collection of n-grams that contribute to the overall probability distribution p can be denoted as S_p. n-grams that are not present in the collection S_p cannot have any effect on the decision-making process within the classifier. However, it is imperative to consider the presence of n-grams within the set S_p. In prior investigations into the prediction-based analysis of CNN for text, the focus has been on identifying the n-grams found within the input sequence, denoted as S_p, and evaluating their respective rankings as a method of understanding the underlying prediction process. In this context, we adopt an additional complicated perspective. It is important to highlight the following: the ultimate categorization process does not necessarily consider the specific n-gram identities, but rather evaluates these individuals based on the rankings allocated through the filters. Therefore, it is imperative that the data contained in variable p is contingent upon the allocated rankings. From a conceptual standpoint, the n-grams within the set S_p can be categorized through two distinct classes: accidental and deliberate. The presence of deliberate n-grams in S_p can be attributed to their higher rankings assigned by the filtering mechanism. This suggests that these n-grams possess valuable information that is relevant to the ultimate decision-making process. In contrast, it is observed that accidental n-grams, regardless of possessing a relatively low ranking, manage to find their way into the set S_p. This occurrence can be attributed to the absence of an additional n-gram that achieved a higher ranking compared to them. Based on the analysis conducted, it is evident that the n-grams in question do not appear to possess significant informational value in relation to the classification selection at hand. Is it possible to distinguish and differentiate between intentional and unintentional n-grams? It is postulated that within the framework for every filter, a discernible threshold exists. Values surpassing this threshold are

indicative of useful information pertaining to the classification process, whereas values falling beneath the threshold are deemed inaccurate and are therefore disregarded for classification purposes. Formally, given threshold dataset (X, Y).

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

Based on our empirical findings, we determine that an ideal purity-value of 0.75 is optimal when determining the threshold of a given filter. Further, the results of the proposed work have been evaluated using three datasets and in terms of recall, precision, and macro averaged F1-score. The results are discussed in the next section.

4. RESULTS AND DISCUSSION

This section commences by providing a comprehensive overview of the system requirements, followed by a detailed examination of the dataset employed in the study. Additionally, the performance metrics utilized to evaluate the system's efficacy are thoroughly examined. The results obtained from the proposed methodology were subsequently compared to those of previous studies, specifically on the basis of recall, precision, and macro-averaged F1-score. The inclusion of a comprehensive discussion section within the present work serves to provide a thorough analysis and interpretation of the obtained results.

4.1. System requirements, datasets and performance metrics

In this section, we conduct a series of experiments on Phase 1 and Phase 2. The code was written in Python and a system having Windows 10 operating system, 16 GB RAM was used for executing the code. In this approach three different datasets namely data mining (DM) [20], operating system (OS) [21], and data base (DB) [22], [23] datasets are used for experiment purposes. The DM, OS, and DB were created in [24]. The performance of the presented approach AAUT-ML is investigated with different existing available approaches i.e., YAKE [15], TF-IDF [25], and TextR [26]. All the results and datasets have been taken from [24]. Results from the provided AAUT-ML are analyzed, and metrics such as recall, precision, and F1-score are used to determine how well the model performs. By dividing the sum of all anticipated sequences within a positive category by the precision, we can determine how many positive categories were successfully anticipated. In (9) can calculate recall, that can be described as the proportion of correctly predicted positive outcomes compared to actual positive outcomes. In machine learning, the F1-score is a crucial evaluation metric. It neatly summarizes a model's prediction ability through the combination of the precision and recall measures, which are frequently assessed 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.

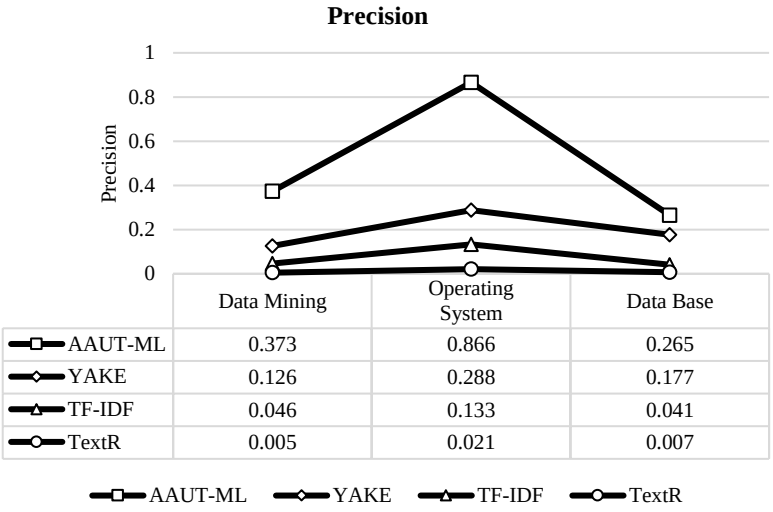
4.2. Precision

In Figure 2, the precision has been evaluated and compared with the YAKE, TF-IDF, and Text-R models. For the DM dataset, the AAUT-ML performed better when compared to YAKE, TF-IDF, and TextR by 66.21%, 87.66%, and 98.65% respectively. For the OS dataset, the AAUT-ML performed better when compared to YAKE, TF-IDF, and TextR by 66.74%, 84.64%, and 97.57% respectively. For the DB dataset, the AAUT-ML performed better when compared to YAKE, TF-IDF, and TextR by 33.20%, 84.52%, and 97.35% respectively. The proposed AAUT-ML has achieved better results for precision in comparison to the YAKE, TF-IDF, and TextR.

4.3. Recall

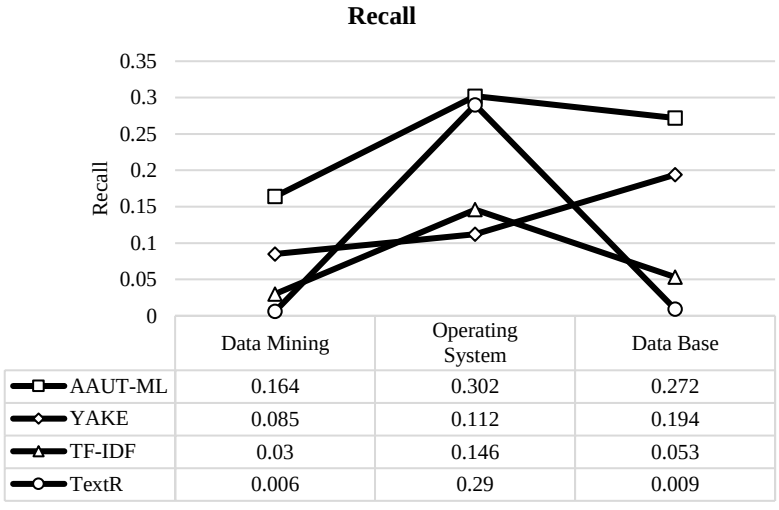
In Figure 3, the recall score has been evaluated and compared with the YAKE, TF-IDF, and Text-R models. For the DM dataset, the AAUT-ML performed better when compared to YAKE, TF-IDF, and TextR

by 48.17%, 81.70%, and 96.34% respectively. For the OS dataset, the AAUT-ML performed better when compared to YAKE, TF-IDF, and TextR by 62.91%, 51.65%, and 3.97% respectively. For the DB dataset, the AAUT-ML performed better when compared to YAKE, TF-IDF, and TextR by 28.67%, 80.51%, and 96.69% respectively. The proposed AAUT-ML has achieved better results for recall in comparison to the YAKE, TF-IDF, and TextR.



	Data Mining	Operating System	Data Base
AAUT-ML	0.373	0.866	0.265
YAKE	0.126	0.288	0.177
TF-IDF	0.046	0.133	0.041
TextR	0.005	0.021	0.007

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

In Figure 4, the macro averaged F1-score has been evaluated and compared with the YAKE, TF-IDF, and Text-R models. For the DM dataset, the AAUT-ML performed better when compared to YAKE, TF-IDF, and TextR by 64.93%, 72.22%, and 96.52% respectively. For the OS dataset, the AAUT-ML performed better when compared to YAKE, TF-IDF, and TextR by 61.88%, 86.06%, and 95.28% respectively. For the DB dataset, the AAUT-ML performed better when compared to YAKE, TF-IDF, and TextR by 30.79%, 82.88%, and 99.61% respectively. The proposed AAUT-ML has achieved better results for macro averaged F1-Score in comparison to the YAKE, TF-IDF, and TextR.

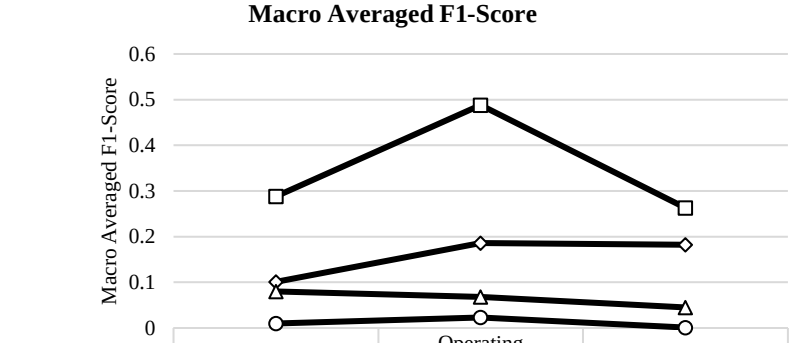


Figure 4. Macro-averaged F1-scores of AAUT-ML versus three baseline methods on three different datasets

4.5. Discussion

Table 1 shows macro averaged F1-score, recall, and precision values for four different methods (AAUT-ML, YAKE, TF-IDF, TextR) applied to three different datasets (data mining, operating system, data base). For the "operating system" dataset, it is seen that AAUT-ML seems to perform the best across all metrics, with high F1-score, recall, and precision values. Also, in the "data mining" dataset, AAUT-ML performs relatively well compared to other methods. Finally, for the "data base" dataset, YAKE and TF-IDF have similar F1-scores, recall, and precision values, with AAUT-ML performing slightly better. This work also tried to utilize the sparemax and fusedmax but both these failed to achieve higher results in comparison to 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

In this work, sensing complex meanings from unstructured data using natural language processing and convolution neural network techniques has been presented. In this analysis, different datasets namely data base, data mining, and operating systems datasets are used. Our study has challenged some conventional assumptions about how CNNs process and classify text. Firstly, we have demonstrated that max-pooling over time introduces a thresholding effect on the output of the convolution layer, effectively distinguishing between relevant and non-relevant features for the final classification. This insight allowed us to identify the crucial n-grams for classification, associating each filter with the class it contributes to. We have also highlighted instances where filters assign negative values to specific word activations, leading to low scores for n-grams containing them, despite otherwise having highly activating words. These findings contribute to enhancing the interpretability of CNNs for text classification. Our approach effectively categorizes various documents and their respective domains. We evaluated its performance using metrics such as precision, recall, and F1-score across multiple datasets, demonstrating superior results compared to existing methods. The performance of the presented work shows better results in comparison to the existing works. For future work, this work can be used for classifying corpus semantics in structured data. Different feature extraction processes can be used for structuring the data. Also, along with NLP, machine learning can be used.

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/n18-2105.

BIOGRAPHIES OF AUTHORS

Shankarayya Shastri is assistant professor in the Department of Computer Science and Engineering at GM Institute of Technology, Davangere, India with more than 16 years of teaching and research experience. He Holds a Bachelor of Engineering degree in computer science and engineering and M.Tech. degree in computer science and engineering. His research areas are text mining, data mining, big data analytics and pattern recognition. He can be contacted at email: shan.shas@gmail.com.

Veeragangadhara Swamy Teligi Math received the Bachelor of Engineering degree in computer science from the University BDT College of Engineering, Davangere and the M.Tech. degree in computer science and engineering from Dr. A.I.T, Bangalore, Karnataka. He obtained his PhD from SJJT University, Rajasthan. He used to hold several administrative posts in various engineering colleges in Karnataka. He has supervised and co-supervised more than 50 masters and 5 PhD students. He has authored or coauthored more than 40 publications: 10 proceedings and 40 journals. His research interests include data mining, web mining, big data and pattern recognition. He can be contacted at email: swamytm@gmail.com.

Patil Nagaraja Siddalingappa working as associate professor in Department of Information Science, Bapuji Institute of Engineering Technology, Davangere. His teaching experience is 13 years and his area of interest is graph database, big data, database management system, and data mining. He can be contacted at email: patilbathi@gmail.com.