JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Deep Learning Algorithms for Text Mining: A Survey

V.K.Janani Sivapriya[1], Dr. R.Saravanamoorthy[2]
Dr. S. Angel[3]
[1]*Research Scholar, Department of Computer Science , Kongunadu Arts and Science College, Coimbatore-29, TamilNadu.*
[2]*Associate Professor, Department of Computer Science , Kongunadu Arts and Science College, Coimbatore-29, TamilNadu.*
[3]*Assistant Professor, Department of Cyber Security with Computer Science , PSGR Krishnamal College for Women, Coimbatore, TamilNadu.*

***Abstract* -Text mining, which focuses on the analysis and extraction of valuable information from unstructured text data, has become a crucial tool in the big data era. Over the past few years, text mining has become increasingly important. Users can now access knowledge through text mining from a wide range of sources, including print, digital, and electronic media and many others. Deep learning, a subfield of machine learning, has emerged as a powerful paradigm for tackling complex text mining tasks. Deep learning algorithms have been widely adopted for text mining, due to their ability to capture complex patterns in textual data. This survey paper aims to provide an overview of the recent advancements in the application of deep learning algorithms in text mining. This will be the source of future studies in this field.**

Keywords— Deep Learning, Text Mining, RNN, CNN, Transformer Model, Attention Mechanism, Text Classification, Topic Categorization, *Sentiment Analysis, Question Answering.*

Introduction

Text mining refers to the process of extracting valuable, engaging, and significant information or patterns from unstructured data sources. Unstructured information is very prevalent and constitutes the majority of knowledge available for an information mining project aimed at analyzing patterns [1]. Extracting valuable insights from raw text can undoubtedly provide beneficial business insights from text-based information or content. However, due to the inherent inconsistencies in natural language text, utilizing natural language processing (NLP), statistical modeling, and machine learning (ML) techniques to mine unstructured content can often be challenging [2][3]. The Unstructured content contains ambiguities due to inconsistent semantics and syntax, along with slang, language specific to different age groups and industries and text with sarcasm [4][5]. Different research efforts are underway in this area to obtain effective and comparable outcomes utilizing a range of machine learning and deep learning models [6][7][8]. Recent improvements in deep learning architectures enhance the models' abilities to capture long-range dependencies, handle sequential information, and learn from vast amounts of data. These advancements are appropriate and efficient for a variety of Text Mining tasks. The aim of this paper is to summarise the use of Deep Learning Algorithms in the process of Text Mining.

Deep Learning:

The idea behind Deep Learning is to mimic and recreate how the human brain operates. Because Human brain seems to be one of the powerful tools for learning adapting skills and applying them and neurons in the brain are responsible for this. In human brain there are approximately 100 billion neurons which are deeply interconnected with each other to form a complex network. In computers the structure of human brain is recreated by an artificial structure called Artificial Neural Network (ANN) where the nodes in it are similar to the neurons. The ANN have an input layer, hidden layers and an output layer. When there are many numbers of hidden layers connected with each other which processes the input data and produces the output value, it is said that deep learning happens.

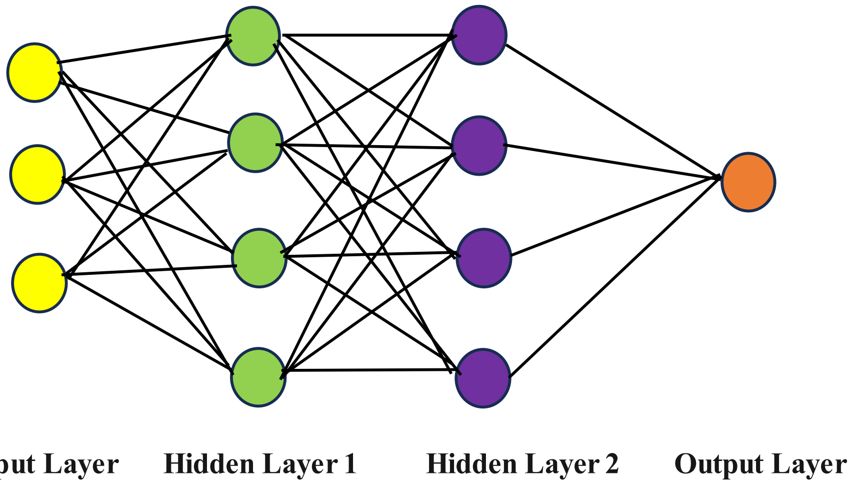

Figure.1.0: Artificial Neural Network

Deep Learning Algorithms:

1. **Recurrent Neural Network (RNN)**

A basic and widely-used algorithm in deep learning for text mining is the **Recurrent Neural Network (RNN).** It is a type of neural network designed for processing sequential data by retaining information from previous inputs through connections between nodes over time, i.e., RNN algorithm process input data step-by-step by preserving the order in which words or data points appear.

RNNs are particularly well-suited for text mining in which understanding the order and context of data is essential, as they have an internal memory that captures dependencies across sequences. The memor y of RNN is attained through a "hidden state" that stores information about previous inputs, making it capable of understanding context in sequences. It processes sequences by iterating through the sequence elements and maintaining a state that contains information relative to what it has seen so far. In effect, RNN is a type of neural network that has an internal loop (see figure 1.1).

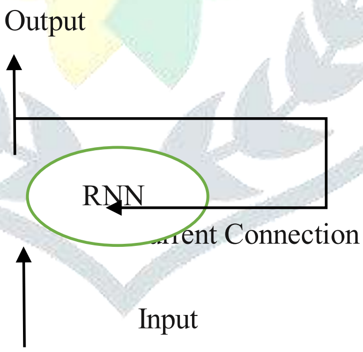

Figure:1.1 – A Recurrent Network: A Network with a Loop.

Recurrent Neural Networks (RNNs) incorporate feedback loops in their recurrent layers which effectively allows them to retain information over time. However, standard RNNs have significant challenges when it comes to learning long-term dependencies. This limitation is predominantly due to the exponential decay of the loss function's gradient, known as the vanishing gradient problem, which hinders training effectiveness for temporal sequences. To solve these problems, researchers develop intricate RNN architectural models. The most popular design, LSTM which stands for Long Short-Term Memory networks. These are a type of Recurrent Neural Network (RNN) that incorporates specialized units alongside standard units [9]. LSTM units include a “memory cell” that can retain information for extended periods. A series of gates are used to control when data is added to memory, when it is retrieved, and when it is discarded. This design enables LSTMs to learn and understand long-term dependencies effectively. By learning the long-term dependencies, and with the help of the internal memory system, LSTMs can solve the disappearing gradients issue [10]. Recognition of Speech and Modern Machine Translation often rely on LSTMs.

2. **Convolutional Neural Networks**

Convolutional neural networks (CNNs) get their name from the mathematical process known as convolution, which assesses how compatible its input functions are. These networks are typically used when data is formatted or requires representation in a 2D or 3D data map. In data map representations, the proximity of data points typically reflects their correlation in information. In convolutional neural networks (CNNs) dealing with image inputs, the data map shows that the pixels of an image are highly correlated with their neighboring pixels. As a result, the convolutional layers have three dimensions: width, height, and depth. This understanding likely accounts for the fact that a substantial amount of research focused on CNNs is conducted in the field of computer vision [11].

A CNN processes an image that is represented as an array of numbers. Through a series of mathematical operations, it transforms the image into a different output space. This process is known as feature extraction, which is essential for highlighting and capturing important aspects of the image. The features extracted can be utilized for various analysis and tasks. For instance, image classification seeks to sort images into specific predefined categories. Other applications include identifying the objects within an image and pinpointing their locations. This does not imply that CNN is limited to just image data; there are also studies that have explored its application with text data. Text Classification and Sentiment Analysis are the considerable applications of CNN in processing text data in Natural Language Processing (NLP).

When applying CNNs to NLP tasks, sentences or documents are represented in matrix form. Each row of the matrix corresponds to a linguistic unit, like a word or a character. Most CNN architectures develop representations of words or sentences during their training phase. Different CNN architectures have been utilized in various classification tasks, including Sentiment Analysis and Topic Categorization [12],[13]-[15]. CNNs have also been used for Relation Extraction and Relation Classification tasks [16][17].

3. **Transformer-based Models**

Primarily Transformers perform Parallelization. Long range dependencies in sequences are encoded by Transformers. Transformers generate text one word at a time. It generates long and elaborate sentences and it does them one word at a time. For Example: If the prompt or question is “Hello, how are you?”, Transformer model does not generate the answer for this prompt, instead it generates the next word that would follow the given prompt, i.e., “Hello, how are you doing”.

Transformer models are versatile and powerful, capable of tasks like chatting, answering questions, and writing code. Despite needing large datasets and computing power, their architecture is relatively simple, consisting of blocks like attention and neural networks. The architecture of Transformer models involves components like attention, feed forward neural networks, and embeddings, which work together to generate text one word at a time. The concept of generating text one word at a time, as seen in Transformer models, is similar to predictive text suggestions in messaging apps, showcasing a common application of this approach.

In recent years, the transformer model has revolutionized Natural Language Processing (NLP). Models like BERT, GPT-3, and RoBERTa have excelled at numerous NLP tasks, surpassing the performance of RNNs. The Transformer Model is set apart or unique due to its self-attention mechanism, allowing it to focus on multiple parts of the input sequence simultaneously and evaluate their significance.

Self-Attention Mechanism:

The Transformer Architecture is driven by the self-attention mechanism which simplifies and increases the processing of the Transformer Network. Unlike RNNs, Transformers can handle and process all inputs simultaneously which enables the connection of inputs to complicated context. The input sequence, consisting of "key," "value," and "query," is transformed into vectors for self-attention. In this setup, keys and values represent the input data, while queries are used to assess the context.

Transformer-Based Models for Text Mining

A new era in NLP is emerged due to the advent of transformers. These Transformer models are powerful pre-trained language models. These models, typically trained on extensive text datasets, have shown incredible proficiency in understanding contextual information and subtle linguistic details. Following are notable Transformer based models,

A. **BERT (Bidirectional Encoder Representations from Transformers)**:

BERT, developed by Google, transformed natural language processing by learning the context in both directions during training. Its pre-training tasks include Masked Language Modeling (MLM), which enables the model to grasp context by predicting omitted words.

B. GPT (Generative Pre-trained Transformer) Series:

The GPT series (GPT-2 and GPT-3), designed by OpenAI, is renowned for its focus on generative tasks that involve creating text. Among these, GPT-3 is particularly notable, as it boasts an impressive 175 billion parameters, making it one of the most extensive language models ever developed. Its vast scale allows it to exhibit extraordinary capabilities in generating human-like text, enabling it to understand and produce language with a level of coherence and relevance that was previously unattainable.

C. RoBERTa:

RoBERTa or Robustly Optimized BERT Approach, was created by Facebook AI Research. This model is created to enhance the performance of BERT. It achieves this by optimizing the training objectives and hyperparameters of BERT. Training becomes robust and more efficient because this model makes use of the Masked Language Modeling (MLM) objective like BERT and it is not focused on the Next Sentence Prediction (NSP) which is one of the pre-training objectives followed by BERT.

D. XLNet:

XLNet is an improvement over BERT and other Transformer-based models. This pre-trained model is proposed by Google to overcome few limitations in training part of BERT. It uses Permutation Language Modeling instead of Masked Language Model (MLM) which is used by BERT. In Permutation Language Modeling, all possible permutations of the sequence are considered during the training process. No specific tokens are considered. This permits XLNet to learn the context bidirectionally, similar to BERT and prevents the issues related to artificial masking of tokens.

E. T5 (Text-to-Text Transfer Transformer):

Google introduced T5 in 2020. It is specifically created for natural language processing (NLP) tasks and treats all such tasks as text-to-text problems. This approach implies that both the input and output consist of text strings, no matter the type of task involved.

4. Attention Mechanisms:

Attention Mechanism is the one behind all the transformer models and it is the core to the Large Language Models (LLM). It is inspired by the human visual processing system. It is a technique that allows the neural network to focus on specific parts of an input sequence. This is done by assigning weights to different parts of the input sequence with the most important parts receiving the highest weights. Focusing on specific part of input sequence gives the neural network model the capability to prioritize relevant information. This improves the model's ability to capture essential features.

Attention mechanism was first used by Bahdanau for machine translation. The traditional method of machine translation was to use sequence to sequence models. The challenge with this model is that if the sentence was long and detailed, the translation was incorrect and inaccurate. The main idea of attention mechanism is that the context vector access is given to the entire input sequence, instead of the last hidden state. Even though the length of the sentence increases, the context vector can still contain the context of the sentence. Hence it can be translated correctly and accurately.

5. Word Embedding Techniques

Word embeddings are approaches in natural language processing (NLP) that denote words as dense, low-dimensional vectors in continuous space. These vectors capture the meanings, connections, and contextual details of words, making them crucial for numerous NLP applications such as machine translation, sentiment analysis, and text generation. These techniques convert textual data into numerical formats, which are suitable for machine learning models, especially those based on deep learning.

Word embeddings are powerful tools in Text Mining especially NLP tasks that offer several key characteristics. One of the main advantages is dimensionality reduction, which transforms sparse, high-dimensional text data, such as one-hot encoding, into dense vector representations. This process not only minimizes the number of computational resources required but also enhances the efficiency of machine learning models. Additionally, word embeddings excel at capturing semantic relationships, enabling them to recognize connections between words, including synonyms, antonyms, and even analogies. For example, through specific vector operations, one can express relationships like "Lord - Man + Lady ≈ Queen." Furthermore, these embeddings provide contextual understanding by representing the meaning of a word based on its usage within a given context, thereby contributing to more nuanced and accurate interpretations of language.

Few Word Embedding Techniques are: Word2Vec, GloVe, FastText, Contextualized embeddings (e.g., BERT, ELMo).

In a paper, Yu et al. [18] introduced an improved word vector model designed to customize pre-trained word vectors for sentiment analysis (SA). This word vector model does not depend on labeled datasets and can be applied as a post-processing step to any pre-trained word vectors generated by various word embedding models, such as GloVe and Word2Vec. GloVe, or Global Vectors for Word Representation, is an algorithm that operates without supervision to create vector representations of words [19]. Based on their findings, Dhingra et al. [20] suggested utilizing GloVe vectors that have been pre-trained for initialization, as they showed better performance than other embeddings. They also recommended substituting infrequent words with a unique token during training, which can then be used in the testing phase to handle out-of-vocabulary (OOV) words.

Conclusion:

Deep Learning algorithms has robust capabilities for dealing unstructured textual data. This survey outlines the features and use of RNN, CNN, Transformer-based models, Attention mechanisms and Word Embedding techniques. The performance and efficiency of Text mining tasks like Natural Language Processing, Text classification, Question Answering, Topic Modelling are enhanced due to the advancements in attention mechanisms, pre-trained models, and word embedding techniques. By reviewing the existing literature, this paper serves as a valuable reference for researchers and practitioners, paving the way for continued advancements in text mining through deep learning.

References:

[1] A. C. Pandey, Rajpoot, and D. Singh, “Improving sentiment analysis using hybrid deep learning model,” Recent Patents on Computer Science, 2019.

[2] A. C. Pandey and D. S. Rajpoot, “Spam review detection using spiral cuckoo search clustering method,” Evolutionary Intelligence, pp. 1–18, 2019.

[3] V. Gupta, A. Mehta, A. Goel, U. Dixit, and A. C. Pandey, “Spam detection using ensemble learning,” in Harmony Search and Nature Inspired Optimization Algorithms. Springer, 2019, pp. 661–668.

[4] A. C. Pandey, D. S. Rajpoot, and M. Saraswat, “Twitter sentiment analysis using hybrid cuckoo search method,” Information Processing & Management, vol. 53, no. 4, pp. 764–779, 2017.

[5] R. Ghai, S. Kumar, and A. C. Pandey, “Spam detection using rating and review processing method,” in Smart Innovations in Communication and Computational Sciences. Springer, 2019, pp. 189–198.

[6] L. Zhang, S. Wang, and B. Liu, “Deep learning for sentiment analysis: A survey,” Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 8, no. 4, p. e1253, 2018.

[7] A. Singhal, P. Sinha, and R. Pant, “Use of deep learning in modern recommendation system: A summary of recent works,” arXiv preprint arXiv:1712.07525, 2017.

[8] A. C. Pandey, S. R. Seth, and M. Varshney, “Sarcasm detection of amazon alexa sample set,” in Advances in Signal Processing and Communication. Springer, 2019, pp. 559–564.

[9] P. Yenigalla, S. Kar, C. Singh, A. Nagar, and G. Mathur, “Addressing unseen word problem in text classification,” in International Conference on Applications of Natural Language to Information Systems. Springer, 2018, pp. 339–351.

[10] Muriel F. Franco et al., “SecRiskAI: A Machine Learning-Based Approach for Cybersecurity Risk Prediction in Businesses,” 2022 IEEE 24th Conference on Business Informatics (CBI), Amsterdam, Netherlands, pp. 1-10, 2022. [CrossRef] [Google Scholar] [Publisher Link].

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, pp. 1097–1105, 2012.

[12] Y. Kim, “Convolutional neural networks for sentence classification,” arXiv preprint arXiv:1408.5882, 2014. [32] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, “Parsing natural scenes and natural language with recursive neural networks,” in Proceedings of the 28th international conference on machine learning (ICML-11), pp. 129–136, 2011.

[13] C. dos Santos and M. Gatti, “Deep convolutional neural networks for sentiment analysis of short texts,” in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 69–78, 2014.

[14] R. Johnson and T. Zhang, “Effective use of word order for text categorization with convolutional neural networks,” arXiv preprint arXiv:1412.1058, 2014.

[15] R. Johnson and T. Zhang, “Semi-supervised convolutional neural networks for text categorization via region embedding,” in Advances in neural information processing systems, pp. 919–927, 2015.

[16] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2335–2344, 2014.

[17] T. H. Nguyen and R. Grishman, "Relation extraction: Perspective from convolutional neural networks," in Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing, pp. 39–48, 2015.

[18] L.-C. Yu, J. Wang, K. R. Lai, and X. Zhang, "Refining word embeddings for sentiment analysis," in Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 534– 539.

[19] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," arXiv preprint arXiv:1802.05365, 2018.

[20] B. Dhingra, H. Liu, R. Salakhutdinov, and W. W. Cohen, "A comparative study of word embeddings for reading comprehension," arXiv preprint arXiv:1703.00993, 2017.

[21] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pretraining of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[22] A. Baevski, S. Edunov, Y. Liu, L. Zettlemoyer, and M. Auli, "Clozedriven pretraining of self-attention networks," 2019.

[23] E. F. Tjong Kim Sang and F. De Meulder, "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition," in Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4, pp. 142–147, Association for Computational Linguistics, 2003.

[24] K. Clark, M.-T. Luong, C. D. Manning, and Q. V. Le, "Semisupervised sequence modeling with cross-view training," arXiv preprint arXiv:1809.08370, 2018.

[25] Z. Geng, G. Chen, Y. Han, G. Lu, and F. Li, "Semantic relation extraction using sequential and tree-structured lstm with attention," Information Sciences, vol. 509, pp. 183–192, 2020.

[26] X. Han, T. Gao, Y. Lin, H. Peng, Y. Yang, C. Xiao, Z. Liu, P. Li, M. Sun, and J. Zhou, "More data, more relations, more context and more openness: A review and outlook for relation extraction," arXiv preprint arXiv:2004.03186, 2020.

[27] R. Socher, J. Pennington, E. H. Huang, A. Y. Ng, and C. D. Manning, "Semi-supervised recursive autoencoders for predicting sentiment distributions," in Proceedings of the conference on empirical methods in natural language processing, pp. 151–161, Association for Computational Linguistics, 2011.

[28] X. Wang, Y. Liu, S. Chengjie, B. Wang, and X. Wang, "Predicting polarities of tweets by composing word embeddings with long shortterm memory," in Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), vol. 1, pp. 1343–1353, 2015.

[29] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in Proceedings of the 2013 conference on empirical methods in natural language processing, pp. 1631–1642, 2013.

[30] C. Ma, W. E. Zhang, M. Guo, H. Wang, and Q. Z. Sheng, "Multidocument summarization via deep learning techniques: A survey," arXiv preprint arXiv:2011.04843, 2020.