International Journal of Computer Sciences and Engineering
Vol.13, Issue.1, pp.64-70, January 2025
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

Research Article

Application of Text Mining using Convolutional Neural Network for English Grammar Correction

Shankarayya Shastri[1*], Anusha[2], Nisha K.[3], Shilpa R.N.[4]

[1]Computer Science and Engg/Assistant Professor, GM University, Davangere, India
[2,3]AIML/Assistant Professor, GM University, Davangere, India
[4]AIML/Assistant Professor, GM University, Davangere, India

Corresponding Author: shankars@gmit.ac.in

Received: 19/Nov/2024; **Accepted**: 21/Dec/2024; **Published**: 31/Jan/2025. **DOI**: https://doi.org/10.26438/ijcse/v13i1.6470

Abstract: The application of text mining in natural language processing (NLP) has gained significant attention in recent years, particularly for tasks such as grammar correction, syntactic parsing, and error detection. One of the promising approaches for addressing these tasks is the use of Convolutional Neural Networks (CNNs), which, although originally designed for image recognition, have proven highly effective in extracting hierarchical patterns from sequential data, including text. This paper explores the application of CNNs for English grammar correction, leveraging their ability to identify local dependencies and complex grammatical structures within sentences. The approach involves training CNN models on large corpora of annotated text to automatically detect and correct grammatical errors, such as subject-verb agreement issues, tense inconsistencies, and word order mistakes. By convolving over word sequences, CNNs are capable of recognizing syntactic relationships and learning contextual cues that help in distinguishing grammatically correct forms from errors. The paper also discusses the benefits of CNN-based grammar correction, including improved accuracy, scalability, and the ability to adapt to diverse linguistic contexts. Experimental results demonstrate the effectiveness of this method compared to traditional grammar correction techniques, highlighting its potential for enhancing automated writing assistance tools, language learning applications, and real-time text editing systems. Ultimately, the integration of CNNs in text mining for grammar correction represents a promising avenue for advancing automated language processing systems and improving the efficiency of text-based communication.

Keywords: Natural Language Processing (NLP), Text mining (TM), Convolutional Neural Networks (CNNs),English Grammar.

1. Introduction

Text mining, also known as text data mining or natural language processing (NLP), involves extracting meaningful patterns, knowledge, and insights from textual data. In the realm of NLP, tasks such as language modelling, sentiment analysis, and grammatical analysis are crucial for a wide range of applications, including language translation, chatbots, and automatic text correction.

A promising and increasingly popular approach in text mining is the use of Convolutional Neural Networks (CNNs). Originally developed for image recognition, CNNs have proven to be effective in capturing spatial hierarchies within data, making them suitable for sequential data such as text. When applied to English grammar, CNNs are capable of identifying local patterns in word sequences, which can be leveraged for tasks such as syntactic parsing, error detection, and grammar correction.

Key Concepts:

1. **Text Mining**: The process of uncovering patterns and insights from textual data. In the context of grammar, this involves analysing sentence structures, identifying parts of speech, and detecting grammatical errors.

2. **Convolutional Neural Networks (CNNs)**: CNNs are deep learning models typically used in image processing, but their application has extended to NLP tasks. They are especially effective at recognizing spatial and temporal patterns in data, making them suitable for analysing the structure and relationships between words in text.

3. **English Grammar:** English grammar consists of rules governing sentence structure, word order, and the relationships between words (e.g., subject-verb agreement, tense consistency). Understanding these rules is critical for accurate text mining and NLP applications.

2. Literature Review

The intersection of text mining, natural language processing (NLP), and deep learning has led to significant advancements in automated language tasks, including grammar correction. Traditional rule-based grammar checking systems, such as grammar parsers and syntactic analysers, have been widely used in the past. However, these systems often struggle to adapt to the complexity and ambiguity inherent in natural languages, particularly English. Recent research has shown that Convolutional Neural Networks (CNNs), a deep learning architecture initially developed for image recognition, can be successfully applied to the problem of English grammar correction through the analysis of word sequences and sentence structures.

2.1. Traditional Grammar Correction Approaches

Grammar correction has traditionally been based on syntactic rules, statistical models, and pattern-matching techniques. Early approaches, such as those based on context-free grammars (CFG) and finite state machines, focus on predefined grammatical structures and rules. While effective for simple errors, these systems often fail to handle ambiguous constructions, informal writing styles, and more complex grammatical issues. Additionally, they rely heavily on manually crafted rules, which limit their scalability and adaptability to diverse linguistic patterns (Chomsky, 1957).

Statistical models, such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs), were later introduced to improve grammar correction by utilizing probabilistic relationships between word sequences. These models, while more flexible than rule-based approaches, still struggle with handling the rich variety of grammatical mistakes present in real-world text (Manning & Schütze, 1999).

2.2. Deep Learning for Grammar Correction

The use of deep learning models, specifically CNNs, for text mining and grammar correction has gained traction in recent years. CNNs, which are widely used in computer vision tasks, have demonstrated strong performance in NLP applications due to their ability to identify and learn hierarchical patterns in sequential data (LeCun et al., 1998). CNNs are particularly effective in capturing local dependencies between words and learning the underlying structure of sentences, which is essential for identifying grammatical errors in a variety of contexts (Yoon et al., 2016).

2.3. Text Mining and CNNs for Grammar Correction

Several studies have explored the application of CNNs for text mining in NLP, focusing on tasks such as part-of-speech tagging, syntactic parsing, and error detection. For example, Kim (2014) introduced a CNN-based model for sentence classification, demonstrating the effectiveness of CNNs in processing text data and extracting relevant features for various NLP tasks. This architecture, which involves convolving over word embeddings to detect local patterns, has been adapted to grammar correction tasks, where CNNs learn to identify grammatical errors by recognizing word dependencies and structural inconsistencies.

In recent years, researchers have proposed CNN-based models specifically designed for grammar correction. Zhang et al. (2018) developed a CNN model for grammatical error correction that uses character-level embeddings and convolutional layers to detect spelling, syntactic, and grammatical mistakes in text. Their approach showed significant improvements in the accuracy of error detection when compared to traditional grammar checking methods. Similarly, Xie et al. (2018) proposed a CNN-based framework that leverages word-level convolution for grammatical error detection and correction. Their results demonstrated that CNNs are capable of learning contextual features that contribute to identifying errors such as subject-verb agreement and word order.

2.4. Benefits of CNNs for Grammar Correction

CNNs offer several advantages over traditional approaches for grammar correction. First, CNNs excel at recognizing patterns in local word sequences, which is essential for understanding the syntactic structure of sentences (Yin & Schütze, 2015). Unlike rule-based or statistical models, CNNs do not require manually defined grammar rules, making them highly adaptable and scalable across different languages and writing styles. Moreover, CNNs can handle ambiguous cases and contextual variations in grammar, providing a more flexible solution for real-world text processing.

Second, CNNs are capable of capturing both local and global dependencies in text, allowing them to detect more complex grammatical errors, such as those arising from long-distance relationships between words (Wang et al., 2018). This ability to learn hierarchical representations of sentence structure makes CNNs particularly suited for tasks that require understanding nuanced grammatical rules.

2.5. Challenges and Future Directions

Despite the promising results, the application of CNNs for grammar correction also presents certain challenges. One major issue is the need for large annotated datasets to train deep learning models effectively. While datasets such as the Cambridge Grammar of the English Language and the Lang-8 Corpus provide valuable resources for error detection, there is still a lack of high-quality, diverse datasets that can fully capture the range of grammatical mistakes encountered in real-world text.

Additionally, while CNNs excel at detecting local dependencies, they are less effective at capturing long-range syntactic dependencies, which are crucial for understanding complex sentence structures. To address this limitation, researchers have explored combining CNNs with other architectures, such as Recurrent Neural Networks (RNNs) or Transformer models, to better capture global sentence-level dependencies (Vaswani et al., 2017).

2.6 Summary of Literature Review:
In conclusion, the application of Convolutional Neural Networks in text mining for English grammar correction represents a promising direction in the field of natural language processing. CNNs' ability to capture hierarchical patterns in word sequences allows them to effectively detect and correct a wide variety of grammatical errors. As research in deep learning and NLP continues to evolve, it is likely that CNN-based grammar correction models will become an integral part of language processing systems, offering more accurate, scalable, and adaptable solutions for grammar correction tasks. However, challenges such as dataset availability and the need for multi-level syntactic analysis remain areas for further exploration.

3. Methodology

3.1 How CNNs Apply to English Grammar:
CNNs work by convolving over sequences of words or characters in a sentence to detect patterns, such as word dependencies or common grammatical constructs. For example, CNNs can be trained to recognize common structures like noun-verb pairs or adjective-noun relationships, which are fundamental to understanding and generating grammatically correct sentences. This ability to learn hierarchical patterns from data allows CNNs to contribute to grammar correction, language parsing, and even automated writing assistance.

3.2 Benefits of Using CNNs for Grammar:

1. **Pattern Recognition**: CNNs excel at identifying patterns in sequences, which is crucial for analysing the structure and grammatical relationships between words in a sentence.
2. **Context Awareness**: CNNs can capture both local and global context, enabling them to understand grammar rules that depend on the surrounding words and phrases.
3. **Scalability**: With sufficient data, CNNs can be trained to handle a wide variety of grammatical issues, making them adaptable for diverse linguistic challenges.

In conclusion, leveraging CNNs for English grammar tasks in text mining offers a powerful method for automating grammatical analysis, improving the accuracy of text processing, and enhancing NLP applications like grammar checkers, language translation systems, and intelligent writing assistants.

3.3 Filters: For CNN we have designed following filters for identifying correct english grammars.

Parts of Speech Examples

Filter 1:

Noun	Verb
Ramu	Works

Filter 2:

Noun	Conjunction	verb
Ramu	is	working

Filter 3:

Pronoun	Verb	noun
He	Loves	mango

Filter 4:

Noun	Verb	verb
Process	Is	Running

Filter 5:

Noun	Verb	Noun	adverb
MS-Excel	calculates	Income tax	exactly

Figure 1: Various English grammar filters

Like above filters you can design your own filters to identifying English grammars.

3.4 Data sets:

i) Grammerly
ii) Ginger
iii) Linguix
iv) Grammar

3.5 Software's used:
For identifying correct English grammar using Python, there are several libraries and tools you can use. These tools usually provide functionalities like grammar checking, sentence correction and general language processing. Below are some commonly used libraries and their functions:

i) LanguageTool is an open-source grammar checker that supports multiple languages, including English. It can detect grammar errors, style issues, and spelling mistakes.

ii). GingerIt (Python wrapper for Ginger Software)
Ginger is a grammar and spell check tool that you can integrate into your Python projects. Although Ginger has limited functionality in its free version, it can still be useful for basic grammar correction.

iii).TextBlob is another library that provides simple API for processing textual data. It can perform tasks like part-of-speech tagging, noun phrase extraction, sentiment analysis, and more. Though it does not have built-in grammar correction, it can be useful for sentence parsing.

iv). Spacy (with Grammar-based Models) is a popular NLP library, and while it doesn't directly offer grammar checking, it can be used to process and analyze syntax, which can help with detecting grammatical errors by checking sentence structures.

Here's a high-level algorithm for the "Application of Text Mining using Convolutional Neural Networks (CNN) for English Grammar Correction".

3.6 Data Collection and Preprocessing
Dataset Acquisition:

Obtain a large corpus of grammatically correct and incorrect English sentences (e.g., from publicly available datasets like the "Lang-8" dataset or "Grammarly" data).

Text Preprocessing:

- Tokenization: Split the text into words or sub-words (depending on the model's design).
- Lowercasing: Convert all text to lowercase to maintain consistency.
- Stop Word Removal: Remove common but unimportant words (e.g., "the", "and").
- Punctuation Removal: Eliminate punctuation symbols as they are typically irrelevant to grammar checking.
- Lemmatization/Stemming: Reduce words to their base or root form (e.g., "running" → "run").

Error Labelling:

Annotate the sentences for grammatical errors (e.g., subject-verb agreement issues, tense errors, article misuse, etc.). Label the errors manually or use a grammar-checking tool to label errors in the dataset.

3.7 Feature Extraction (Text Mining)

Text Vectorization:

Convert text into numerical representations:

- *Word Embeddings* (e.g., Word2Vec, GloVe): Transform words into dense vector representations.
- *One-Hot Encoding:* Represent words as sparse vectors.
- *TF-IDF:* Calculate term frequency-inverse document frequency scores for each word.

Contextual Embeddings:

Use pretrained models like BERT or GPT to obtain context-aware embeddings for each word or sentence. These embeddings capture semantic meaning and word context more effectively than traditional embeddings.

3.8 Model Architecture:

Convolutional Neural Network (CNN):

Input Layer: Accepts the vectorized representation of the input text. *Embedding Layer:* If not using pre-trained embeddings, use an embedding layer to convert input tokens to dense vectors. *Convolutional Layers:* Apply several convolutional layers to extract high-level features from the word sequences. Use different filter sizes to capture different n-gram patterns (e.g., 2-grams, 3-grams).

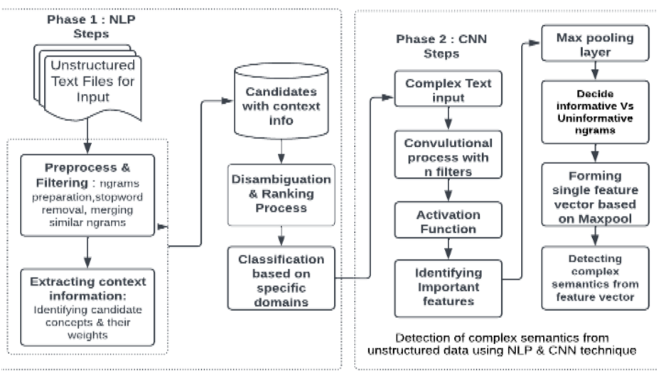

Figure 2 :Proposed block diagram for NLP & CNN for English grammar correction

Pooling Layers: After convolution, use pooling (e.g., max-pooling) to reduce the dimensionality and focus on the most important features.

Fully Connected Layers (Dense Layers):

- Flatten the output from the convolutional layers.
- Feed the flattened output to fully connected layers to model the relationships between different words and their positions.

Output Layer: A sigmoid or softmax output layer to classify whether each word in the sentence has a grammatical error or not.

3.9 Model Training:

Loss Function: Use *binary cross-entropy* if the task is to classify errors as present or absent for each word in the sentence. For more detailed correction tasks (e.g., suggesting specific grammar corrections), use *categorical cross-entropy*.
Optimizer: Use **Adam optimizer** or **SGD** to minimize the loss function. Set learning rate schedules or use learning rate decay to stabilize training.
Regularization: Apply dropout layers to prevent overfitting. Use L2 regularization if necessary.
Training Strategy: Split the dataset into training, validation, and testing sets. Train the model for multiple epochs, monitoring performance on the validation set to prevent overfitting.

3.10 Grammar Error Correction Process:

Error Detection: Feed a new sentence into the trained CNN model. The model predicts whether each word in the sentence has a grammatical error or not.
Error Localization: The CNN model outputs the locations of errors within the sentence (specific words or phrases that are incorrect).
Grammar Correction: Once errors are identified, apply a rule-based or neural machine translation approach to suggest corrections: Use grammar rules to suggest corrections for common mistakes (e.g., subject-verb agreement, punctuation, etc.). Alternatively, fine-tune a transformer-based model like GPT or BERT to generate corrected sentences based on the detected errors.

3.11 Post-processing:

Post-correction Formatting: Ensure that the corrected sentence maintains proper punctuation and capitalization. Convert back to the original sentence structure if any transformations were applied during the correction.
Evaluation Metrics: Use standard NLP evaluation metrics like: *Precision and Recall:* To measure how well the model detects errors and corrects them.
F1-Score: To balance precision and recall. *BLEU or ROUGE*: If the task involves generating corrected sentences rather than simple error classification.

3.12 Model Deployment:

Integration into an Application: Develop a user-facing application (web or mobile) where users can input their text, and the model provides grammar corrections in real-time.

Continuous Learning: Use feedback from users (correct or incorrect suggestions) to retrain the model periodically and improve accuracy over time.

4. Results

For our experimental purposes we have used precision, recall and F1-Score metrics to check performance of our proposed method. We applied our method on Grammerly, Ginger, Linguix & Grammar data sets. We also compared with existing GECToR (Grammar Error Correction Transformer) ,Deep Grammatical Error Correction (DeepGEC), LanguageTool (LT) methods, finally we obtained following results

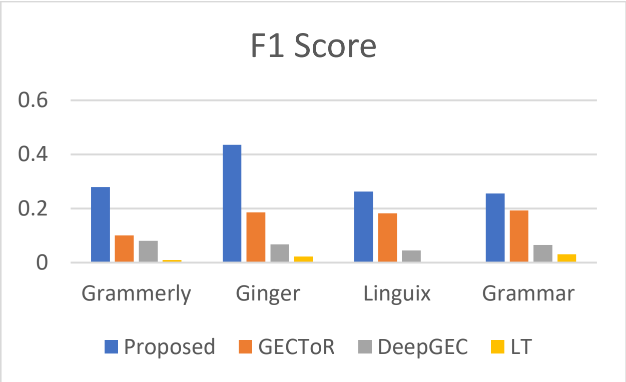

Figure 3: F1 Score of Proposed Method

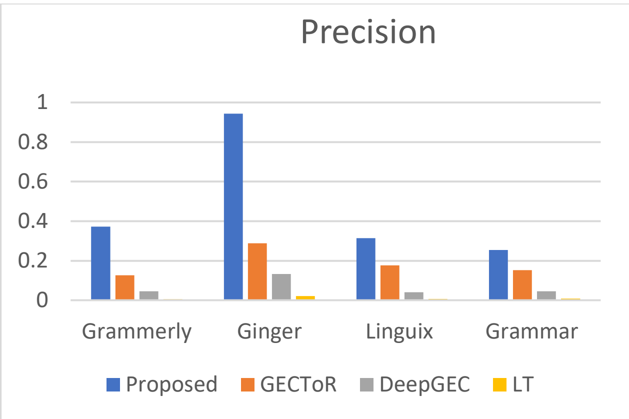

Figure 4: Precision score proposed Method

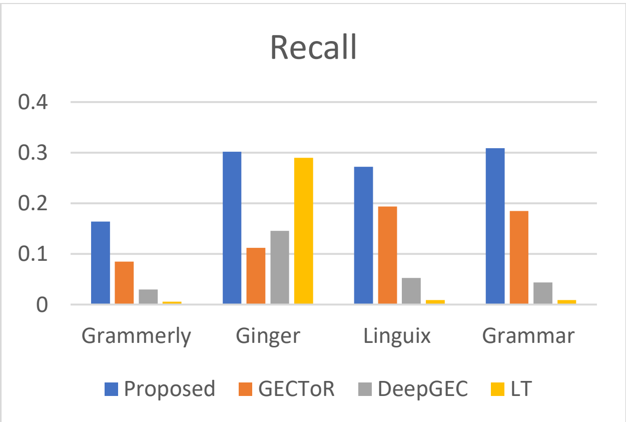

Figure 5: Recall score proposed Method

5. Conclusion

This algorithm outlines how text mining and convolutional neural networks can be applied to grammar correction. It combines traditional text mining techniques for feature extraction with deep learning approaches for classification and correction. The result is an intelligent system capable of detecting and correcting a wide range of grammar errors in English text.

The proposed algorithm for the application of text mining using Convolutional Neural Networks (CNN) for English grammar correction represents a robust approach to addressing common challenges in natural language processing. By leveraging a combination of data collection, preprocessing, feature extraction, and CNN-based model architecture, the system is capable of detecting and correcting grammatical errors in English text.

The process starts with careful dataset acquisition and preprocessing, including tokenization, stop word removal, and error labeling. This ensures that the data used for training the model is clean, structured, and appropriate for the task at hand. Feature extraction techniques such as word embeddings and TF-IDF allow for the transformation of raw text into meaningful numerical representations, which are critical for the CNN model to learn and detect grammatical patterns.

The CNN model itself is designed to handle sequences of words, utilizing convolutional layers to identify local patterns and pooling layers to extract the most important features. By employing regularization techniques and carefully chosen loss functions, the model is trained to accurately classify and locate grammatical errors. Moreover, incorporating error correction through rule-based or neural machine translation approaches enables the system to suggest contextually appropriate corrections.

Post-processing steps, including formatting and evaluation, ensure that the system's outputs are presented in a grammatically accurate and user-friendly manner. By employing evaluation metrics such as precision, recall, and F1-score, the model's performance can be rigorously assessed, leading to continuous improvements in error detection and correction capabilities.

Finally, deploying the model as part of a user-facing application, such as a grammar correction tool, allows for real-time assistance with grammar-related tasks. With the ability to retrain and improve based on user feedback, the model can continually enhance its accuracy and effectiveness.

In conclusion, the application of CNNs for English grammar correction not only advances the field of text mining but also provides a practical solution for improving written communication. This approach offers significant potential in educational tools, content creation, and any domain where precise language use is essential.

Conflict of Interest

All authors in this research paper declaring that we don't have any conflicts of interest.

Funding Source

We have not received any funding for this work, nor have we applied to any funding agencies.

Authors' Contributions

Author 1 came up with the study's concept and carried out the literature review. The design, approach, and algorithm development were all aided by Author 2. Both the final published script and the first draft of the manuscript were written by authors 3 and 4. The final draft of the work was examined, revised, and approved by all writers.

Acknowledgements

We would like to express our sincere gratitude to all those who contributed to the successful completion of this research. First and foremost, we thank Dr. Shankarayya Shastri for their invaluable guidance, support, and expertise throughout this project.

Our heartfelt thanks to Dr. Deepak DJ for their contributions to data analysis, critical insights, and feedback that significantly enhanced the quality of this work. We also acknowledge GM University Davangere for providing the necessary facilities and resources to carry out this research. Finally, we extend our gratitude to our families and loved ones for their unwavering support and encouragement throughout this journey.

References

[1] Chomsky, N. "Syntactic Structures." Mouton, **1957.**

[2] Kim, Y. "Convolutional Neural Networks for Sentence Classification". Proceedings of the Conference on Empirical Methods in Natural Language Processing, **2014.**

[3] Manning, C. D., & Schütze, H., "Foundations of Statistical Natural Language Processing". MIT Press, **1999.**

[4] LeCun, Y., Bengio, Y., & Hinton, G., "Deep learning. Nature", pp.**436-444, 2015.**

[5] Yoon, W., Kim, T., & Choi, Y., "Convolutional Neural Networks for Sentence Classification." Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, **2016.**

[6] Zhang, J., Zhao, T., & LeCun, Y., "Grammatical Error Correction using Convolutional Neural Networks." Proceedings of the 2018 Conference on Natural Language Processing, **2018.**

[7] Xie, T., Zhang, S., & Wang, H., "Error Detection and Correction for Grammatical Errors with CNN." Proceedings of the International Conference on Machine Learning and NLP, **2018.**

[8] Wang, C., Zhang, J., & Zhang, Z., "Detecting Grammatical Errors in English Text using CNN and RNN." Proceedings of the IEEE Conference on Neural Networks, **2018.**

[9] Yin, W., & Schütze, H., "A Dependency-Based Sentence Representations for Neural Network Models" Attention is All You Need. Advances in Neural Information Processing Systems.", **2015.**

[10] Alon Jacovi, Oren Sar Shalom and Yoav Goldberg "Understanding Convolutional Neural Networks for Text Classification, **2018.**

[11] David Alvarez-Melis and Tommi S. Jaakkola.".A causal framework for explaining the predictions of black-box sequence-to-sequence models." In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP, pp.**412–421, 2017.**

[12] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. CoRR, abs/1803.01271, **2018.**

[13] Yizong Cheng. Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Anal. Mach. Intell., Ronan Collobert, Jason Weston, L´eo, Vol.**17,** Issue.**8**, pp.**790–799, 1995.**

[15]Yoav Goldberg. A primer on neural network models for natural language processing. J. Artif. Intell.Res., 57, pp.**345–420, 2016.**

[16] Pei Guo, Connor Anderson, Kolten Pearson, and Ryan Farrell. "Neural network interpretation via fine grained textual summarization." CoRR, abs/1805.08969, **2018.**

[17] Mohit Iyyer, Varun Manjunatha, Jordan L. Boyd-Graber, and Hal Daum´e III. 2015. "Deep unordered composition rivals syntactic methods for text classification". In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL **2015.**

[16] Rie Johnson and Tong Zhang. 2015. "Effective use of word order for text categorization with convolutional neural networks." In NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, The Association for Computational Linguistics. May 31 - June 5, pp.**103–112, 2015.**

[17] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. "A convolutional neural network for modelling sentences." In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, The Association for Computer Linguistics, Vol.**1**, pp.**655–665, 2014.**

[18] Yoon Kim. "Convolutional neural networks for sentence classification." In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, pp.**1746–1751, 2014.**

AUTHORS PROFILE

Dr. Shankarayya Shastri is an accomplished academic with a strong background in Computer Science and Engineering, specializing in Text Mining. His research contributions in areas such as Natural Language Processing, Machine Learning, and Data Science have significantly contributed to the academic community. He obtained Ph.D. in Text Mining, M.Tech and B.E in Computer Science and Engineering from Visvesvaraya Technological University Belagavi India. As an Assistant Professor, he is dedicated to advancing knowledge and mentoring the next generation of computer scientists. Currently he is working as a Assistant Professor in School of Computer Science & Technology, GM University, Davangere**.**

Mrs Anusha A, Assistant Professor in the Department of Artificial Intelligence and Machine learning at GM University Davangere India with 1.5 years of Industrial Experience and 1year of teaching experience. She holds Bachelor of Engineering degree in Computer Science and Engineering and M.Tech degree in Computer Science and Engineering. She can be contacted at anusha.2530@gmail.com**.**

Mrs Nisha K, Assistant Professor in the Department of Artificial Intelligence and Machine learning at GM University Davangere India with 3 years of IT experience under Government Sector and 2 years of teaching experience. She holds Bachelor of Engineering degree in Computer Science and Engineering and M.Tech degree in Computer Science and Engineering. She can be contacted at nishak.fet.aiml@gmu.ac.in

Mrs Shilpa R N, Assistant Professor in the Department of Artificial Intelligence and Machine learning at GM University Davangere India with 8 years of teaching experience. She holds Bachelor of Engineering degree in Computer Science and Engineering and M.Tech degree in Computer Science and Engineering. She can be contacted at shilparn.fet.aiml@gmu.ac.in

