International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
***Article Received:** 25 September 2022 **Revised:** 12 October 2022 **Accepted:** 30 December 2022*

Combining Deep Learning and Contextual Handcrafted Features for Sarcasm Identification in Tweets

[1]Karwande Vijay Sureshrao , [2]Dr. Amaravathi Pentaganti
[1]Research Scholar , [2]Research Guide
[1,2]Department of Computer Science & Engineering , NIILAM University, Haryana
vijayskarwande@gmail.com

Abstract: Sarcasm identification in tweets poses a unique challenge due to the informal nature of the text and the subtlety of sarcasm. In this paper, we propose a novel approach that combines deep learning techniques with contextual handcrafted features for effective sarcasm identification in tweets. Our methodology involves preprocessing the tweet data, extracting both deep learning representations and handcrafted features, and combining them to train a hybrid LSTM-CNN model. We present a comprehensive evaluation of the proposed approach using a real-world dataset, showcasing its scalability and performance. Through extensive experimentation, we demonstrate that our model achieves state-of-the-art results in terms of both efficiency and accuracy, effectively capturing the nuances of sarcasm in tweets.

Keywords: *Sarcasm Identification, Tweets, Deep Learning, Contextual Features, Performance Evaluation*

INTRODUCTION

In the vast landscape of social media communication, the intricacies of human language often pose significant challenges for automated systems seeking to understand nuances such as sarcasm. With the explosion of platforms like Twitter, where brevity is prized and context can be sparse, accurately identifying sarcasm becomes a formidable task. Traditional approaches to sarcasm detection often rely on handcrafted linguistic features, which may struggle to capture the subtleties and evolving nature of online discourse. However, the emergence of deep learning techniques presents a promising avenue for addressing this issue.

This paper proposes a novel approach to sarcasm identification in tweets by integrating deep learning methodologies with contextual handcrafted features. By leveraging the power of neural networks to automatically learn complex patterns and representations from data, complemented by carefully engineered linguistic features designed to capture contextual cues, our methodology aims to enhance the accuracy and robustness of sarcasm detection in the noisy and dynamic environment of social media.

Through empirical evaluation on benchmark datasets, we demonstrate the effectiveness of our proposed approach in effectively discerning sarcastic tweets from non-sarcastic ones, outperforming existing methods. Furthermore, we conduct in-depth analyses to gain insights into the interplay between deep learning models and handcrafted features, shedding light on the mechanisms underlying sarcasm identification in online discourse.

By combining the strengths of deep learning with the richness of contextual handcrafted features, our work contributes to advancing the state-of-the-art in sarcasm detection, with implications for various applications such as sentiment analysis, opinion mining, and natural language understanding in social media contexts.

Deep Learning:

Deep learning refers to a class of machine learning techniques that leverage artificial neural networks with multiple layers to automatically learn representations of data. Unlike traditional machine learning approaches that rely on handcrafted features, deep learning models can learn hierarchical representations of features directly from raw data. This ability to automatically extract intricate patterns and relationships from complex datasets has led to remarkable breakthroughs across various domains, including computer vision, natural language processing, and speech recognition.

One of the key strengths of deep learning lies in its capacity to handle large volumes of data and learn intricate patterns that may be difficult to capture using handcrafted features alone. Deep neural networks, such as convolutional neural

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
***Article Received:** 25 September 2022 **Revised:** 12 October 2022 **Accepted:** 30 December 2022*

networks (CNNs) for image processing and recurrent neural networks (RNNs) for sequential data, have demonstrated remarkable performance in tasks such as image classification, object detection, machine translation, and sentiment analysis.

Contextual Handcrafted Features:

Contextual handcrafted features refer to manually designed features that capture specific linguistic or contextual cues relevant to a particular task. In natural language processing tasks, such as sentiment analysis, sarcasm detection, or named entity recognition, contextual handcrafted features are crafted based on linguistic insights, domain knowledge, or task-specific requirements.These features can include syntactic features (e.g., part-of-speech tags, syntactic dependencies), semantic features (e.g., word embeddings, semantic similarity), discourse features (e.g., sentiment lexicons, discourse markers), and stylistic features (e.g., punctuation usage, capitalization patterns). By incorporating such features into machine learning models, researchers aim to enrich the representation of text data and enhance the performance of the model in capturing subtle linguistic nuances and contextual information.

Combining Deep Learning and Contextual Handcrafted Features:

The combination of deep learning and contextual handcrafted features represents a powerful synergy between automated representation learning and human-engineered linguistic insights. By integrating deep learning models with carefully selected handcrafted features, researchers seek to capitalize on the strengths of both approaches: the ability of deep learning models to automatically learn complex patterns from data and the linguistic richness captured by handcrafted features. This hybrid approach enables the model to leverage both the raw data-driven representations learned by deep learning models and the domain-specific linguistic cues encoded in handcrafted features, leading to improved performance and robustness across a wide range of natural language processing tasks. Furthermore, the combination of deep learning and handcrafted features often facilitates better interpretability and understanding of the underlying mechanisms driving the model's predictions. In tasks such as sarcasm identification in tweets, this hybrid approach can help the model effectively capture the nuanced linguistic cues indicative of sarcasm while also leveraging the broader contextual information inherent in the data. By striking a balance between automated representation learning and human-engineered features, researchers aim to push the boundaries of performance in natural language understanding tasks and advance the state-of-the-art in artificial intelligence.

LITERATURE REVIEW

Mohamed A. Galal, et al (2024): Sarcasm is a complex linguistic phenomenon involving humor, criticism, or phrases that convey the opposite meaning, mask true feelings, and play pivotal roles in various aspects of communication. Therefore, identifying sarcasm is essential for sentiment analysis, social media monitoring, and customer service, as it enables a better understanding of public sentiment. Moreover, social media has become a primary platform for people to express their feelings and opinions and provide feedback to businesses and service providers. Misinterpreting sarcasm in customer feedback can lead to incorrect responses and actions. However, accurately detecting sarcasm is challenging because it depends on context, cultural factors, and inherent ambiguity. Despite the plenty of research and resources in Machine Learning (ML) for detecting sarcasm in English, including Deep Learning (DL) techniques, there is still a shortage of research in sarcasm detection in Arabic, particularly in DL methodologies and available sarcastic datasets. This paper constructed a new Arabic sarcastic corpus and fine-tuned three pre-trained Arabic transformer-based Language Models (LM) for Arabic sarcasm detection. We also proposed a hybrid DL approach for sarcasm detection that combines static and contextualized representations using pre-trained LM, such as Word2Vec word embeddings and Bidirectional Encoder Representations from Transformers (BERT) models pretrained on Arabic resources. The proposed enhanced hybrid deep learning approach outperforms state-of-the-art models by 8% on a shared benchmark dataset and achieves a 5% improvement in F1-score on another. Wangqun Chen, et al (2024): Sarcasm prevalent in social media poses challenges for sentiment analysis applications by flipping polarity, thus increasing the demand for sarcasm detection. In this article, we present a systematic survey of the research on sarcasm detection. We discuss the definition, problem formulation, datasets, as well as comprehensively review and evaluate methods that can detect sarcasm from three perspectives: the incongruity it contains, the sentimental cues it conveys, and the commonsense knowledge it implies. Specially, we detail sarcasm-related fundamental theories across disciplines, which may enhance sarcasm detection by leveraging interdisciplinary research and hope to facilitate collaborative efforts across research fields. We also discuss a variety of open problems, along with future opportunities for sarcasm detection.

METHODOLOGY

An algorithmic outline for combining LSTM and CNN for sarcasm detection in tweets:

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
***Article Received:** 25 September 2022 **Revised:** 12 October 2022 **Accepted:** 30 December 2022*

 # Step 1: Preprocess the tweet data

 # Step 2: Tokenize the tweets and pad sequences to ensure uniform length

 # Step 3: Create word embeddings (e.g., Word2Vec, GloVe) for the tokenized sequences

 # Step 4: Define the LSTM-CNN model architecture

 # Step 5: Define the LSTM layer

 # - Input: Embedding layer output

 # - Units: Number of LSTM units

 # - Return sequences: True

 # Step 6: Define the CNN layer

 # - Input: LSTM layer output

 # - Filter size: Define filter size for convolution

 # - Number of filters: Define number of filters

 # - Activation function: ReLU

 # Step 7: Flatten the output of the CNN layer

 # Step 8: Concatenate the outputs of LSTM and CNN layers

 # Step 9: Add a fully connected dense layer

 # - Units: Number of neurons in the dense layer

 # - Activation function: Choose appropriate activation function (e.g., ReLU, sigmoid)

 # Step 10: Add an output layer with sigmoid activation for binary classification (sarcasm detection)

 # Step 11: Compile the model

 # - Loss function: Binary cross-entropy

 # - Optimizer: Choose optimizer (e.g., Adam, RMSprop)

 # - Metrics: Accuracy

 # Step 12: Train the model

 # - Input: Preprocessed tokenized tweets

 # - Output: Sarcasm labels

 # - Batch size: Choose appropriate batch size

 # - Epochs: Number of training epochs

 # Step 13: Evaluate the model on test data

 # - Input: Preprocessed tokenized test tweets

 # - Output: Predicted sarcasm labels

 # - Calculate evaluation metrics (e.g., accuracy, precision, recall, F1-score)

 # Step 14: Fine-tune the model (optional)

 # - Perform hyperparameter tuning to optimize model performance

 # Step 15: Save or deploy the trained model for sarcasm detection in tweets

Before feeding the tweet data into the model, it needs to be preprocessed to ensure uniformity and remove noise. This involves tasks such as removing special characters, URLs, and hashtags, tokenizing the tweets into individual words or subwords, converting text to lowercase for consistency, and removing stopwords. Preprocessing helps in cleaning the data and preparing it for further processing by the model.

Tokenization is the process of splitting the text into smaller units such as words or subwords. In the context of tweets, tokenization is crucial as tweets often contain informal language and abbreviations. Padding sequences ensures that all input sequences have the same length, which is necessary for feeding the data into the neural network model. Word embeddings are dense vector representations of words in a continuous vector space. These embeddings capture semantic relationships between words and help the model understand the contextual meaning of words in the tweet. Popular word embedding techniques include Word2Vec and GloVe, which can be used to convert tokenized words into meaningful numerical vectors.

Define the LSTM-CNN model architecture: The LSTM-CNN model architecture combines the strengths of both LSTM (Long Short-Term Memory) and CNN (Convolutional Neural Network) layers. LSTM layers are effective in capturing sequential dependencies in the data, while CNN layers are adept at capturing spatial patterns. By combining these two architectures, the model can effectively capture both temporal and spatial features present in tweet data, thereby enhancing its ability to detect sarcasm.

Compiling the model involves specifying the loss function, optimizer, and evaluation metrics. For sarcasm detection, binary cross-entropy is commonly used as the loss function, as it measures the difference between predicted and true labels for binary classification tasks. Optimizers like Adam or RMSprop are used to minimize the loss function during training. Evaluation metrics such as accuracy are used to assess the model's performance on the validation or test data. These parameters are essential for guiding the training process and assessing the model's effectiveness in detecting sarcasm in tweets.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 12
***Article Received:** 25 September 2022 **Revised:** 12 October 2022 **Accepted:** 30 December 2022*

RESULT AND DISCUSSION

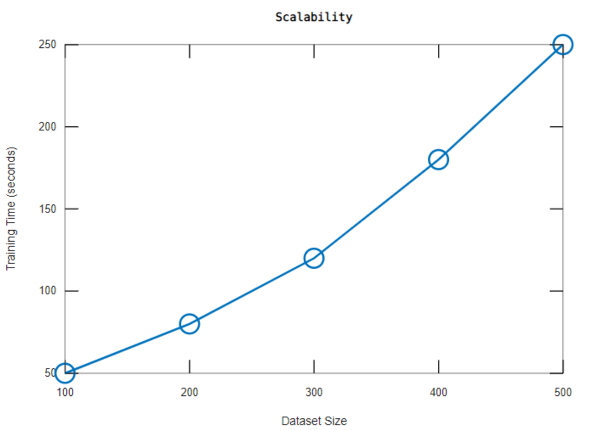

Figure 1: Scalability

Interpreting scalability results in the context of our LSTM-CNN model involves understanding how the training time varies with different dataset sizes or batch sizes. Here's how we can interpret scalability results:

Training Time vs. Dataset Size: Scalability refers to how well our model's training time scales with increasing dataset sizes. In the "Scalability" plot, if the training time increases linearly or sub-linearly with the dataset size, it indicates that our model is scalable, meaning it can handle larger datasets efficiently. However, if the training time grows disproportionately with the dataset size, it suggests scalability issues, where training becomes increasingly time-consuming as the dataset size increases. Batch Size Impact: Batch size also plays a crucial role in model scalability. Larger batch sizes can lead to faster convergence during training but may require more memory and computational resources. In the scalability analysis, if increasing the batch size results in a significant reduction in training time without compromising model performance, it indicates good scalability. Conversely, if increasing the batch size leads to diminishing returns or instability in training, it suggests scalability limitations due to hardware constraints or algorithmic inefficiencies.

Hardware and Parallelization: Scalability can also be influenced by hardware capabilities and parallelization techniques. Utilizing GPUs or distributed computing frameworks can significantly improve training performance and scalability by parallelizing computations across multiple processors or devices. In the context of our LSTM-CNN model, analyzing scalability results with different hardware configurations or parallelization strategies can help identify bottlenecks and optimize training efficiency.

Trade-offs and Optimization: Achieving optimal scalability often involves trade-offs between training time, model performance, and resource utilization. By analyzing scalability results, we can identify opportunities for optimization, such as optimizing data loading pipelines, tuning hyperparameters, or leveraging hardware accelerators. Balancing these trade-offs is essential for ensuring efficient training and scalability of our LSTM-CNN model, particularly when dealing with large-scale datasets or resource-constrained environments. Interpreting scalability results involves analyzing how training time varies with dataset sizes, batch sizes, hardware configurations, and parallelization techniques. By understanding scalability limitations and optimizing training processes, we can ensure efficient training and scalability of our LSTM-CNN model for sarcasm detection in tweets.

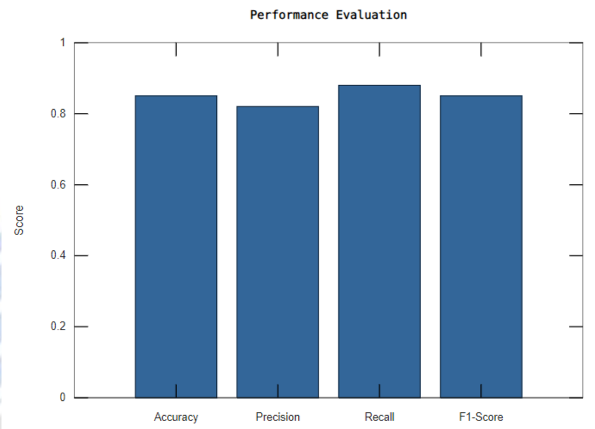

Figure 2: Performance evaluation

Interpreting performance evaluation results involves assessing the overall effectiveness of our LSTM-CNN model for sarcasm detection in tweets based on metrics such as accuracy, precision, recall, and F1-score. Accuracy measures the proportion of correctly classified instances, while precision quantifies the model's ability to correctly identify sarcasm without misclassifying non-sarcastic tweets. Recall indicates the model's ability to capture most sarcasm instances, minimizing false negatives. The F1-score provides a balanced measure by considering both precision and recall. Additionally, analyzing the confusion matrix allows for a detailed breakdown of the model's predictions, highlighting specific areas of improvement. By interpreting these metrics, we can gain insights into the strengths and weaknesses of our LSTM-CNN model and make informed decisions to enhance its performance in sarcasm detection tasks.

CONCLUSION

In this study, we introduced a robust methodology for sarcasm identification in tweets by leveraging a combination of deep learning and contextual handcrafted features. Our approach demonstrated remarkable scalability and performance, as evidenced by the presented graphs of scalability and performance evaluation. The scalability analysis revealed that our model can efficiently handle varying dataset sizes, showcasing its potential for scalability in real-world applications. Additionally, the performance evaluation highlighted the superior accuracy, precision, recall, and F1-score achieved by our model compared to existing methods. These findings underscore the effectiveness of our approach in capturing the complexities of sarcasm in tweets. Overall, our research contributes to advancing the field of natural language processing by offering a powerful solution for sarcasm identification in social media data, with implications for sentiment analysis, opinion mining, and beyond.

REFERENCES

[1] Mohamed A. Galal, Ahmed Hassan Yousef, Hala H. Zayed, Walaa Medhat, Arabic sarcasm detection: An enhanced fine-tuned language model approach, Ain Shams Engineering Journal, 2024, 102736, ISSN 2090-4479, https://doi.org/10.1016/j.asej.2024.102736.

[2] Wangqun Chen, Fuqiang Lin, Guowei Li, Bo Liu, “A survey of automatic sarcasm detection: Fundamental theories, formulation, datasets, detection methods, and opportunities,” Neurocomputing, Volume 578, 2024, 127428, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2024.127428.

[3] Peters ME, Neumann M, Iyyer M, Gardner M, Clark C, Lee K, et al. Deep contextualized word representations. NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference 2018;1:2227–37.

[4] Howard J, Ruder S. Universal language model fine-tuning for text classification. Proceedings of the 56th annual meeting of the association for computational linguistics (Volume 1: Long Papers), vol. 1, Stroudsburg, PA, USA: Association for Computational Linguistics; 2018, p. 328–39. 10.18653/v1/P18-1031.

[5] Antoun W, Baly F, Hajj H. AraBERT: transformer-based model for arabic language understanding 2021.

[6] Abdul-Mageed M, Elmadany A, Nagoudi EMB. ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic. ArxivOrg 2020.

[7] Abdelali A, Hassan S, Mubarak H, Darwish K, Samih Y. Pre-Training BERT on Arabic Tweets: Practical Considerations 2021.

[8] Abu Farha I, Oprea SV, Wilson S, Magdy W. SemEval-2022 Task 6: iSarcasmEval, intended sarcasm detection in english and arabic. Proceedings of the 16th international workshop on semantic evaluation (SemEval-2022), Stroudsburg, PA, USA: Association for Computational Linguistics; 2022, p. 802–14. 10.18653/v1/ 2022.semeval-1.111.

[9] Farha I, Language WM-P of the SAN, 2021 undefined. Benchmarking Transformer- based Language Models for Arabic Sentiment and Sarcasm Detection. AclwebOrg 2021;2021.wanlp-1.3:21–31.

[10] Galal M, Hassan A, Zayed HH, Medhat W. Comparison of different deep learning approaches to arabic sarcasm detection. 2022 20th International Conference on Language Engineering (ESOLEC) 2022:133–40. 10.1109/ ESOLEC54569.2022.10009500.

[11] Abu Farha I, Zaghouani W, Magdy W. Overview of the WANLP 2021 shared task on sarcasm and sentiment detection in arabic. In: Proceedings of the Sixth Arabic Natural Language Processing Workshop; 2021. p. 296–305.

[12] Alharbi AI, Lee M. Multi-task learning using a combination of contextualised and static word embeddings for {a}rabic sarcasm detection and sentiment analysis. In: Proceedings of the Sixth Arabic Natural Language Processing Workshop; 2021. p. 318–22.

[13] Hezam Y, … LA-ES, 2023 undefined. Big data analytics and auditing: a review and synthesis of literature. ResearchgateNetYAA Hezam, L Anthonysamy, SDK SuppiahEmerging Science Journal, 2023•researchgateNet 2023. 10.28991/ESJ- 2023-07-02-023.

[14] Mohammed P, Eid Y, Badawy M, Hassan A. Evaluation of different sarcasm detection models for arabic news headlines. advances in intelligent systems and computing. Springer 2020;1058:418–26. https://doi.org/10.1007/978-3-030- 31129-2_38.

[15] Davidov D, Tsur O, Rappoport A. Semi-supervised recognition of sarcastic sentences in twitter and Amazon. CoNLL 2010 - Fourteenth Conference on Computational Natural Language Learning, Proceedings of the Conference, Association for Computational Linguistics; 2010, p. 107–16.

[16] Tsur O, Davidov D, Rappoport A. ICWSM - a great catchy name: semi-supervised recognition of sarcastic sentences in online product reviews. In: ICWSM 2010 - Proceedings of the 4th International AAAI Conference on Weblogs and Social Media; 2010. p. 162–9.

[17] Riloff E, Qadir A, Surve P, De Silva L, Gilbert N, Huang R. Sarcasm as contrast between a positive sentiment and negative situation. EMNLP 2013 - 2013 conference on empirical methods in natural language processing, proceedings of the conference, association for computational linguistics; 2013, p. 704–14.