ISSN 2723-6471

A Study to Detect Multi-word Expression from Text Using Deep Learning Models

Wong Jun Meng[1], Tan Yu Jie[2], Lim Tong Ming[3*]

[1,2,3] *Faculty of Computing and Information Technology, Ground Floor, Bangunan Tan Sri Khaw Kai Boh, Jalan Genting Kelang, 53300 Kuala Lumpur, Malaysia*

(Received: December 15, 2024; Revised: February 1, 2025; Accepted: April 15, 2025; Available online: June 15, 2025)

Abstract

Detecting Multi-word Expressions (MWEs) is a crucial task in Natural Language Processing (NLP) for applications in machine translation, sentiment analysis, and information retrieval. This study evaluates the performance of several deep learning models on MWE detection using two samples of varying sizes from the major consumer electronic product retailer corpus. The sample is limited to 10,000 and 15,000 rows, with each row contains 15-20 English words. Preprocessing steps include removing special symbols and emojis, converting text to lowercase, and applying the spaCy NLP library for tokenization and part-of-speech (POS) tagging. Syntactic rules are then used to identify MWEs such as verb-noun combinations and phrasal verbs, with BIO tags (B-MWE, I-MWE, O) to mark MWE boundaries. We investigated transformer-based models such as BERT, BERT-CRF, LSTM-CRF and RoBERTa-CRF using a sample of 10,000 rows; BERT, BERT-BiLSTM, BiLSTM-GloVe, and BiLSTM-GloVe-BiGRU uses a sample of 15,000. Results demonstrated that the transformer-based model, RoBERTa-CRF, excels on the smaller sample which achieves the best performance by leveraging the contextual embeddings and sequential dependency modeling. On a larger sample, the BERT-BiLSTM model emerged as the most effective model, showcasing the advantage of combining dynamic embeddings with sequential learning. In contrast, models utilizing static embeddings, such as GloVe, displayed moderate performance, highlighting their limitations in capturing contextual nuances. Comparative analysis across both samples reveals that transformer-based models like RoBERTa-CRF performed optimally on the smaller dataset, whereas hybrid models integrating with sequential architectures like BERT-BiLSTM demonstrated superior performance as dataset size increased. These findings highlight the importance of model selection based on dataset scale to optimize MWE detection. This study underscores the importance of integrating contextual and sequential deep learning techniques to improve MWE detection and provides a basis for developing more robust and scalable systems for diverse linguistic tasks.

Keywords: Multiword Expressions, Natural Language Processing, Deep Learning, Transformer Models, Recurrent Neural Networks, Contextual Embeddings, Sequence Labeling

1. Introduction

Language is a cornerstone of human communication, comprising intricate structures and expressions that convey nuanced meaning. At the heart of linguistic study lies the interaction between a lexicon, the repository of words and their attributes, and grammar, the set of rules governing how these words combine to form coherent expressions. Together, these components provide a systematic framework for analyzing and generating syntactically valid sentences [1]. Despite this structured approach, linguistic phenomena such as Multiword Expressions (MWEs) present unique challenges. A MWE is a lexical unit consisting of a sequence of two or more words, whose characteristics cannot be deduced from the properties of the individual words or their typical patterns of combination [2]. MWEs, which include idioms like “kick the bucket”, compound nouns like “software engineer” and phrasal verbs like “give up”, cannot always be interpreted by analyzing their constituent parts alone. As such, they are a critical focus in both linguistic research and computational applications.

The detection and interpretation of MWEs have significant implications for natural language processing (NLP) tasks, including machine translation, information retrieval, and sentiment analysis [3]. Advances in deep learning, particularly the use of transformer-based architectures such as BERT and RoBERTa, have shown remarkable promise in addressing these challenges by capturing the semantic and syntactic nuances of MWEs. These models leverage contextual word

***Corresponding author: Lim Tong Ming (limtm@tarc.edu.my)**
DOI: https://doi.org/10.47738/jads.v6i3.716

representations, attention mechanisms, and large-scale pretraining to generalize across languages and domains effectively. Recent research has also explored hybrid approaches, combining rule-based methods with neural architectures to enhance performance [4]. However, despite these advancements, several obstacles persist.

A lack of annotated data, variability across languages, domain-specific nuances, and semantic ambiguities create barriers to robust MWE detection. For example, idiomatic expressions often rely on cultural or contextual understanding that deep learning models may fail to capture without specialized datasets. Additionally, computational demands and a lack of standardized evaluation metrics further complicate the implementation and benchmarking of MWE detection systems. These limitations highlight the need for innovative solutions that integrate linguistic expertise with cutting-edge computational methods.

The main contributions of this study seek to address the limitations of machine learning methods in detecting MWEs, including low detection rates and the lack of large, high-quality annotated datasets. By building on these contributions, we seek to evaluate the robustness of deep learning models, develop improved detection techniques to enhance prediction accuracy, and explore strategies for creating extensive and well-annotated datasets.

2. Literature Review

2.1. Deep Learning Models

Transformer models, introduced by Vaswani in 2017, have revolutionized deep learning applications by utilizing a self-attention mechanism to effectively model contextual relationships within sequential data. Unlike traditional neural network architectures such as Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) models, transformers overcome limitations like sequential data processing bottlenecks and vanishing gradients by enabling parallel computation and capturing long-range dependencies. Their versatility has expanded their application beyond NLP to domains like computer vision, multi-modal processing, audio and speech analysis, and signal processing. Researchers have refined the vanilla transformer architecture, giving rise to innovative variants like BERT, GPT, and Vision Transformers (ViTs), tailored to specific tasks like text classification, translation, and image segmentation. Comprehensive surveys, such as the one by Islam underscore the transformative impact of transformers across fields, proposing taxonomies to highlight their application diversity while identifying future research directions. These advancements cement transformers as a cornerstone of modern deep learning research and innovation [5].

Convolutional Neural Networks (CNNs) have revolutionized feature extraction and image classification through hierarchical learning of features. Inspired by the visual cortex, CNNs mimic human visual perception by capturing local features in initial layers and progressively learning complex patterns in deeper layers. The advent of AlexNet in 2012, leveraging GPUs for extensive computation, marked a breakthrough in applying CNNs to large-scale image classification tasks, achieving unprecedented accuracy. Subsequent architectures, including VGGNet and GoogLeNet, have further refined the CNN design for enhanced performance. CNNs' distinctive layers, such as convolutional, pooling, and fully connected layers, enable efficient feature mapping and classification, evidenced by the 85.97% accuracy on the CIFAR-10 dataset reported in this study. The flexibility of CNNs extends beyond image classification to applications in video surveillance, facial recognition, and medical diagnostics, underscoring their adaptability and computational efficiency in handling complex data representations [6].

RNNs are pivotal in processing sequential data, owing to their ability to retain and utilize historical information through recurrent connections. Unlike feedforward networks, RNNs can learn temporal dependencies, making them indispensable in tasks like language modeling, speech recognition, and time-series prediction. A significant milestone in RNN development was the introduction of LSTM networks, which address vanishing gradient issues by employing gating mechanisms for controlling information flow. This innovation enabled RNNs to capture long-range dependencies effectively, overcoming the limitations of traditional architectures. Moreover, advancements such as bidirectional RNNs and the integration of attention mechanisms have further enhanced their capacity to model complex sequences. RNNs' adaptability across domains has solidified their status as a foundational model in deep learning research and applications [7].

LSTM networks have become a cornerstone in sequential data modeling due to their ability to capture long-range dependencies and mitigate the vanishing gradient problem inherent in traditional RNNs. LSTMs achieve this through gating mechanisms such as input, forget, and output gates that regulate the flow of information, allowing effective temporal modeling over extended sequences. Recent advancements, such as combining LSTMs with CNNs and Deep Neural Networks (DNNs) into unified architectures like CLDNNs, have demonstrated superior performance across tasks like speech recognition. CLDNNs leverage the strengths of CNNs for spectral feature extraction, LSTMs for temporal modeling, and DNNs for feature transformation, achieving significant improvements in word error rate (WER) compared to standalone LSTM models. These hybrid approaches highlight the versatility of LSTMs and their adaptability in addressing complex sequence processing challenges in diverse domains [8].

Generative Adversarial Networks (GANs), introduced by Ian Goodfellow in 2014, have revolutionized generative modeling by employing a game-theoretic approach to unsupervised learning. GANs consist of two neural networks, the generator and the discriminator, which are trained simultaneously in a minimax game. The generator aims to produce realistic samples, while the discriminator evaluates their authenticity against real data. This adversarial training enables GANs to learn complex data distributions effectively, leading to state-of-the-art performance in generating high-quality, realistic images. Despite their success, GANs face challenges such as instability during training and mode collapse, prompting research into improved architectures like Wasserstein GANs and Progressive GANs. Applications of GANs span image synthesis, data augmentation, and domain adaptation, underscoring their versatility and transformative impact on artificial intelligence research [9].

Autoencoders, a foundational structure in deep learning, have been instrumental in tasks involving unsupervised learning and nonlinear feature extraction. Rooted in the principles of multi-layer perceptrons, autoencoders learn to encode input data into a compact representation and reconstruct the input, thereby capturing essential data features. Variants like Sparse Autoencoders (SAEs), Denoising Autoencoders (DAEs), anNBd Variational Autoencoders (VAEs) have extended the basic architecture, addressing challenges like robustness to noise, dimensionality reduction, and probabilistic latent space modeling. Emerging adaptations such as Wasserstein Autoencoders (WAEs) and Adversarial Autoencoders (AAEs) further enhance generative capabilities and latent space regularization. Applications of autoencoders span diverse domains, including image classification, recommender systems, and medical diagnostics, highlighting their versatility. Ongoing advancements focus on improving training stability, scalability, and interpretability to address complex data distributions and application-specific challenges [10].

2.2. Deep Learning Models for Multi-word Expression (MWE)

Detecting MWE is a critical task in NLP, with applications spanning machine translation, terminology extraction, and semantic analysis. Traditional approaches, such as LSTM and CNN-based models, have been effective but struggle with issues like polysemy and non-substitutability inherent to MWEs. Recent advancements have focused on transformer-based models, such as BERT, RoBERTa, and XLM-RoBERTa, which leverage contextual embeddings to improve MWE detection accuracy. Empirical evaluations demonstrate that transformer models consistently outperform their LSTM-based counterparts, achieving higher F1 scores on benchmark datasets like the SemEval-2016 Task 10. Notably, multilingual models like XLM-RoBERTa have shown exceptional performance in cross-lingual tasks, indicating their potential for broader linguistic applications. This underscores the transformative impact of neural transformers in addressing the challenges posed by MWEs, paving the way for more robust and generalizable NLP systems [11].

Next, MWE detection has gained prominence as an essential task in NLP, addressing challenges in idiomaticity, discontinuity, and domain variability. Recent studies focus on transformer-based models, such as BERT and its variants, which leverage token classification techniques to excel in sequence annotation tasks. These models outperform traditional sequence tagging approaches like Conditional Random Fields (CRFs) by efficiently capturing syntactic and semantic dependencies. However, the choice of evaluation metrics, pre- and post-processing strategies, and corpus selection significantly influences model performance and interpretability. Benchmarks like PARSEME [12] and DiMSUM [13] shared tasks have provided multilingual datasets and domain-specific corpora to explore diverse MWE categories. Advanced tagging schemes, including BIO-based encoding and dependency tree extractions, are

employed to tackle nesting and overlaps in MWEs. This evolving landscape underscores the role of systematic experimental designs and methodological rigor in advancing MWE detection using deep learning [14].

Besides that, Bidirectional Long Short-Term Memory (BiLSTM) networks have significantly advanced the analysis of Multiword Expressions (MWEs) by effectively capturing contextual information from both past and future sequences. These models have demonstrated high adaptability across linguistic tasks, for example, the use of BiLSTM integrated with dependency treebanks to analyze Polish MWEs [15] and application of BiLSTM-CRF for identifying verbal MWEs, showcasing superior performance in capturing syntactic and semantic nuances [16]. Their strength in multilingual and cross-lingual applications has been highlighted by Taslimipoor who leveraged BiLSTM for cross-lingual transfer learning, and Rohanianwho tackled discontinuous MWEs with a BiLSTM-Graph Convolutional Network hybrid [17]. Further applications include lexical complexity prediction, where Aziz used BiLSTM to distinguish single-word from multi-word complexities [18], and translation tasks, as demonstrated by Garg, who addressed rare word problems in Neural Machine Translation for low-resource languages [19]. Additionally, Vacareanu employed BiLSTM for unsupervised semantic classification, achieving robust embeddings for MWEs. Collectively, these studies emphasize BiLSTM’s efficiency and versatility in addressing the complexities of MWEs across diverse linguistic contexts [20].

Lastly, Bidirectional Gated Recurrent Units (BiGRU) have emerged as a computationally efficient alternative to BiLSTM for analyzing MWEs. These models retain the ability to capture sequential and contextual information while reducing computational overhead. Sarlak demonstrated BiGRU's effectiveness in predicting the compositionality of verbal MWEs in Persian, highlighting its capacity to model non-linear semantic relationships [21]. Similarly, Taslimipoor illustrated the model's success in identifying translation equivalents of MWEs in cross-lingual tasks, ensuring semantic integrity during language translation [22]. BiGRU's lightweight architecture is particularly advantageous in low-resource environments, as shown by Haddad who reported superior performance compared to BiLSTM with reduced memory requirements. Premasiri found that while transformers dominate in large datasets, BiGRU provides a practical alternative for moderately sized corpora [23]. Furthermore, Piasecki and Kanclerz demonstrated the integration of BiGRU with contextual word embeddings to detect MWEs in Polish, achieving improved syntactic and semantic alignment [24]. Berend applied BiGRU in identifying domain-specific MWEs in scientific texts, achieving notable precision in recognizing technical expressions [25]. Collectively, these studies underscore BiGRU's versatility, computational efficiency, and suitability for addressing linguistic challenges across various domains and languages.

3. Methodology

The diagram (figure 1) illustrates a typical workflow for a data-driven project, starting from defining the problem and objectives to eventually deploying and monitoring the model. It emphasizes an iterative approach where each step builds upon the previous ones, involving multiple stages of analysis, model development, and evaluation. Initially, the process starts with understanding the problem and gathering relevant literature. Then, data understanding, collection, and preprocessing take place, followed by data annotation and exploratory data analysis (EDA). Feature extraction and model selection are key stages, leading to model architecture design and training. Throughout this process, model evaluation and parameter tuning occur to refine the model. Finally, once the model is ready, it undergoes deployment and continuous monitoring for performance tracking. Each of these steps is interconnected and often revisited to improve the model's efficiency and effectiveness in real-world applications.

 ISSN 2723-6471

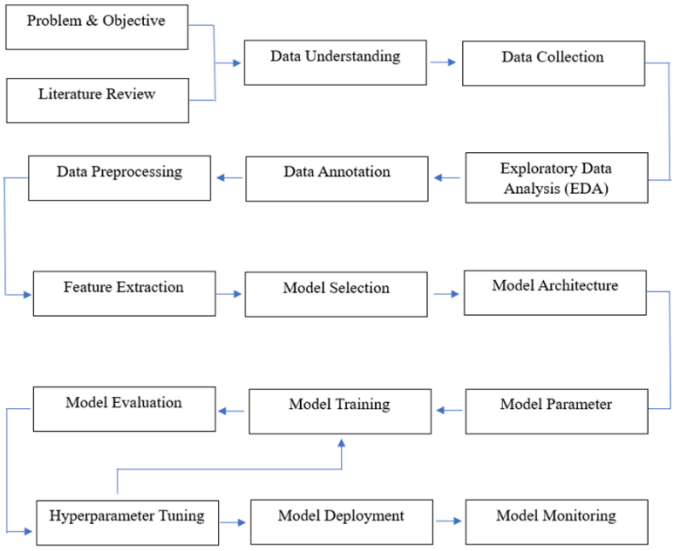

Figure 1. Flow of Methodology

For data preprocessing, the dataset is limited to the first 10,000 rows and 15,000 rows, with each row consisting of 15-20 English words. Special symbols, punctuation marks, and emojis are removed using regular expressions and text normalization techniques in Python. Specifically, regex patterns are employed to detect and eliminate non-alphanumeric characters except for spaces and apostrophes, ensuring that contractions and possessives remain intact. Emojis are filtered out using the *emoji* library, which identifies and replaces them with empty strings. The removal of special characters and emojis is essential to minimize noise in the dataset and improve model performance. These non-linguistic elements do not contribute to syntactic or semantic understanding in MWE detection and can interfere with tokenization, leading to inaccurate word boundaries and misclassification of MWEs. Additionally, retaining such symbols may introduce biases in contextual embeddings, as they lack meaningful representations in most pretrained language models. The text is then converted to lowercase to maintain consistency in tokenization and reduce data sparsity.

After cleaning, the text is processed using the spaCy NLP library, which performs tokenization and part-of-speech (POS) tagging. This information is then used to apply syntactic rules for detecting MWEs, such as verb-noun combinations or phrasal verbs, and assign BIO tags, including B-MWE, I-MWE, and O, to demarcate the boundaries of MWEs. Additionally, predefined MWEs, such as brand names and common phrases, are also identified. The final output consists of tokenized words, their corresponding POS tags, BIO labels, and the detected MWEs. This processed data is saved into both CSV and text files for subsequent use, with an additional filtering step that retains only paragraphs containing MWEs marked with "B-MWE" or "I-MWE". For hyperparameter tuning, the F1-score was utilized as the primary metric to strike a balance between precision and recall, ensuring optimal performance in MWE detection. This was particularly crucial for models such as BERT-BiLSTM and BiLSTM-GloVe-BiGRU, which necessitate meticulous parameter adjustments for enhanced effectiveness. Additionally, validation loss was monitored to mitigate overfitting, enabling the models to generalize well to unseen data.

3.1. Dataset

For the evaluation, we use a local consumer electronic product retailer dataset. The dataset consists of two samples: 10,000 rows (table 1) and 15,000 rows (table 2) of social media text, where each row consists of sentences 15-20 English words only. The data is labeled using the BIO (Begin, Inside, Outside) tagging scheme, which includes the labels B-MWE (beginning of multi-word expressions), I-MWE (inside of multi-word expressions), and O (outside any multi-word expression).

 ISSN 2723-6471

Table 1. 10,000 rows of annotated sample

Feature	Details
Total Tokens	127,567
Unique Tokens	11,659
Label Distribution	B-MWE: 11,430
	I-MWE: 13,181
	O: 100,277

Table 2. 15,000 rows of annotated sample

Feature	Details
Total Tokens	258,971
Unique Tokens	26,283
Label Distribution	B-MWE: 34,152
	I-MWE: 26,283
	O: 180,510

3.2. BERT

The BERT (Bidirectional Encoder Representations from Transformers) model architecture (figure 2) is a multi-layer Transformer encoder designed for various NLP tasks. It begins with input embeddings, where tokens from an input sequence (e.g., t1, t2, [MASK], t4) are converted into dense vector representations, and positional encodings are added to account for word order.

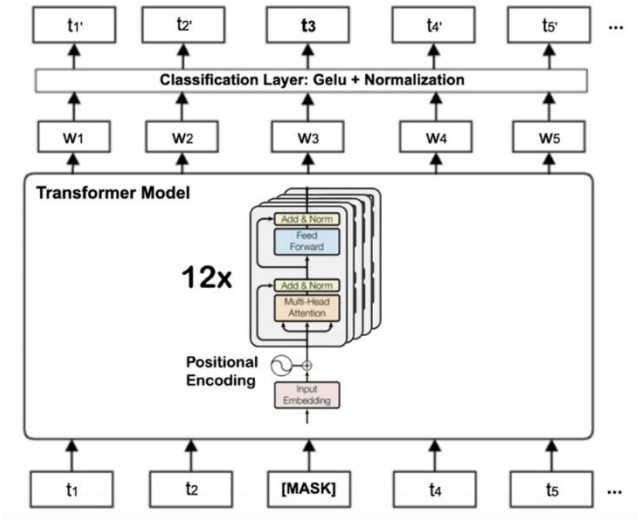

Figure 2. BERT Model Architecture

BERT's core consists of 12 stacked Transformer encoder layers, each containing two main components: multi-head self-attention to capture bidirectional contextual relationships between tokens and a feed-forward network for further processing, both stabilized with Add and Norm operations. The outputs of the final layer are contextualized embeddings for each token (e.g., W1, W2, W3), which are then passed to a classification layer with GELU activation and normalization for specific tasks, such as predicting masked tokens (MLM) or performing downstream tasks like classification. By attending to both left and right contexts simultaneously, BERT effectively captures nuanced word meanings, making it a powerful tool for tasks like MWE detection, named entity recognition, and sentiment analysis.

3.3. BERT-CRF

The BERT-CRF model architecture (figure 3) integrates BERT's powerful contextual language representations with a CRF or a linear layer for sequence labeling tasks, such as NER or MWE detection.

 ISSN 2723-6471

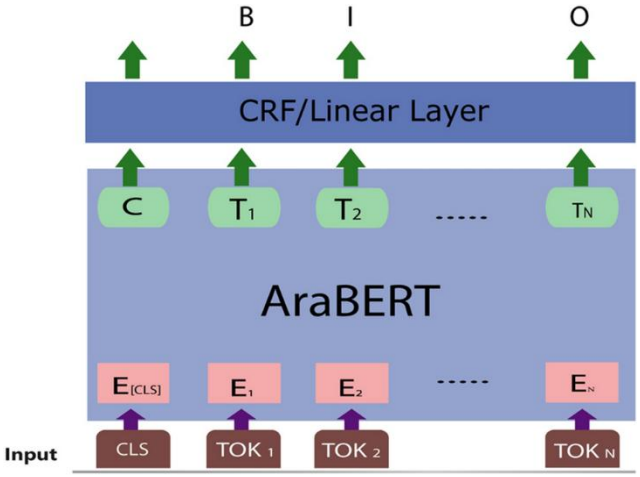

Figure 3. BERT-CRF Model Architecture

The model takes an input sequence of tokens, which includes the special [CLS] token and the actual tokens, and processes them through BERT (in this case, AraBERT for Arabic text) to generate contextual embeddings for each token. These embeddings capture the meaning of each word in its specific context. The CRF or linear layer is then applied to model dependencies between adjacent token labels, helping the model make more accurate predictions by considering the sequential structure of the data. The final output is a sequence of predicted labels (such as B, I, or O) that indicate the role of each token in the context of the task, making this model highly suitable for tasks requiring structured labeling of token sequences.

3.4. RoBERTa-CRF

The RoBERTa-CRF model architecture (figure 4) combines the powerful RoBERTa transformer model with a CRF for sequence labeling tasks like named entity recognition or part-of-speech tagging. In this setup, RoBERTa generates contextualized embeddings for each input token, capturing semantic information from the surrounding context.

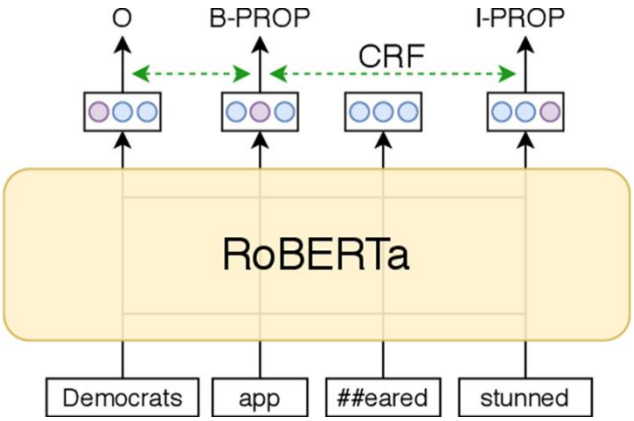

Figure 4. RoBERTa-CRF Model Architecture

These embeddings are then passed to the CRF layer, which models the dependencies between adjacent labels in the sequence. The CRF ensures that the predictions for each token take into account the relationships between neighboring tokens' labels, helping to enforce coherent sequence structures. The model incorporates standard training steps like backward propagation (B-PROP) to update the RoBERTa parameters and input propagation (I-PROP) to forward the token embeddings to the CRF for final label predictions, resulting in accurate sequence-level decisions.

3.5. LSTM-CRF

The LSTM-CRF model architecture (figure 5) combines the sequential modeling strength of BiLSTM networks with the structured prediction capability of a CRF layer. The architecture begins with word embeddings, where each token in the input sequence is represented as a dense vector capturing semantic information. These embeddings are passed to

 ISSN 2723-6471

a BiLSTM layer, which consists of a forward LSTM that processes the input from left to right and a backward LSTM that processes it from right to left. This bidirectional approach allows the model to capture contextual information from both past and future tokens for each position in the sequence.

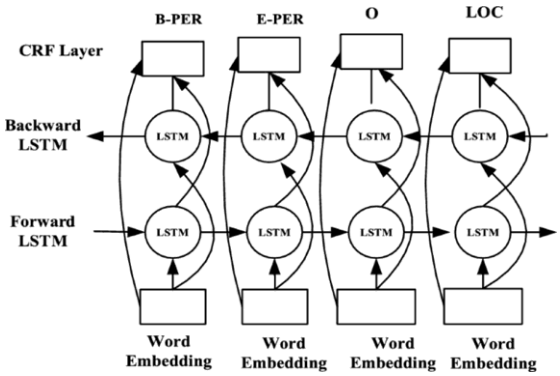

Figure 5. LSTM-CRF Model Architecture

The outputs of the forward and backward LSTMs are concatenated to create rich, context-aware representations for each token. These token-level representations are then fed into a CRF layer, which models dependencies between output labels (e.g., enforcing valid transitions such as from "B-MWE" to "I-MWE"). The CRF layer produces the most likely sequence of labels for the input, ensuring consistent and accurate predictions. This architecture is particularly effective for tasks like MWE detection, where understanding both local token relationships and global sentence structure is crucial.

3.6. BERT-BiLSTM

The BERT-BiLSTM model architecture (figure 6) is designed for NER tasks, specifically using AraBERT, a BERT variant optimized for Arabic text. The input sequence consists of tokenized words (TOK_1, TOK_2, etc.) with a special classification token (CLS). These tokens are first embedded using the AraBERT model, which generates contextualized word representations. The embeddings are then passed to a BiLSTM or BiGRU layer to capture long-range dependencies and contextual relationships.

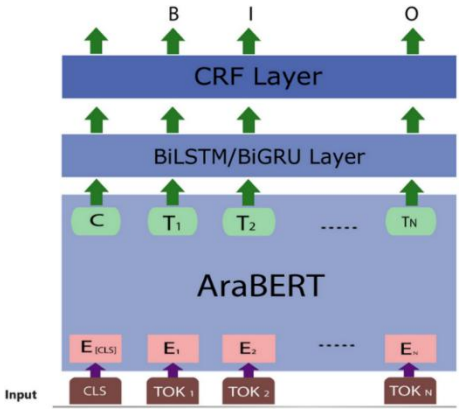

Figure 6. BERT-BiLSTM Model Architecture

The output of the BiLSTM/BiGRU layer is then fed into a CRF layer, which helps improve sequence labeling by considering dependencies between labels. The final output consists of token classifications such as "B" (Begin), "I" (Inside), and "O" (Outside), which are common in NER tasks. This hierarchical approach enhances entity recognition by combining deep contextual embeddings with sequence learning and structured prediction.

3.7. BiLSTM-GloVe

The BiLSTM-GloVe model architecture (figure 7) is designed for text classification tasks by leveraging word embeddings and bidirectional sequence learning. The input consists of words, which are first mapped to dense vector representations using a pre-trained embedding layer (such as GloVe). These word embeddings are then fed into a

 ISSN 2723-6471

BiLSTM layer, which processes the input in both forward and backward directions, capturing contextual dependencies from both past and future words.

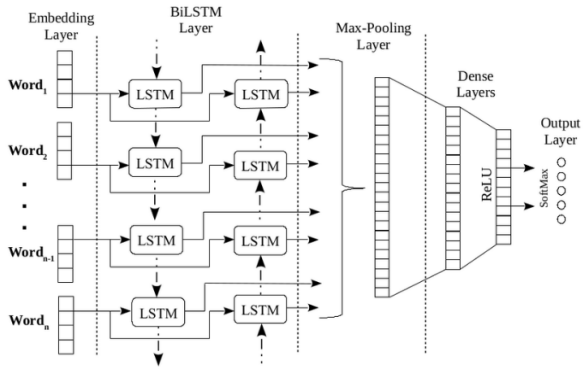

Figure 7. BiLSTM-GloVe Model Architecture

The outputs from the BiLSTM layer undergo a max-pooling operation, which extracts the most significant features across the sequence. These pooled features are then passed through fully connected dense layers with ReLU activation, enabling higher-level feature learning. Finally, the output layer applies a softmax activation function for classification, producing probabilities for different output classes. This architecture effectively combines pre-trained word embeddings, bidirectional contextual learning, and deep learning techniques for improved text classification performance.

3.8. BiLSTM-GloVe-BiGRU

The BiLSTM-GloVe-BiGRU model architecture (figure 8) integrates three key components: GloVe embeddings, a BiLSTM layer, and a BiGRU layer. First, the input text is converted into dense word vectors using GloVe embeddings, capturing semantic relationships between words. These embeddings are then fed into a BiLSTM layer, which processes the sequence in both forward and backward directions, effectively capturing long-term dependencies.

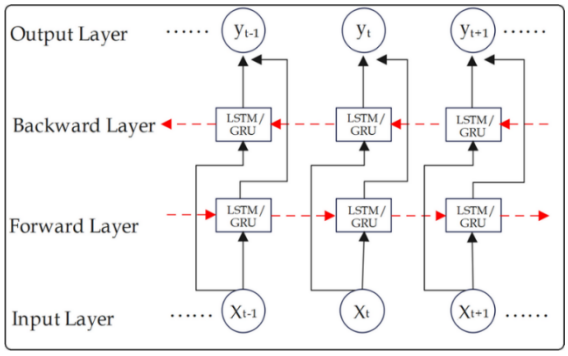

Figure 8. BiLSTM-GloVe-BiGRU Model Architecture

The outputs of the BiLSTM layer are then passed into a BiGRU layer, which further refines contextual understanding by selectively updating and resetting information. The final hidden states from the BiGRU layer are used for downstream tasks such as classification, sentiment analysis, or sequence labeling. This hybrid architecture leverages GloVe for rich word representations, BiLSTM for long-range dependencies, and BiGRU for efficient sequence learning, ensuring robust and accurate text processing.

4. Results and Discussion

4.1. Model Development

In this study, we employed four deep learning models at first such as BERT, BERT-CRF, RoBERTa-CRF and LSTM-CRF model to address the token classification task of MWE detection for dataset that consists of 10,000 rows of annotated sample, each leveraging advanced architectures and methodologies. The BERT-based model utilized

 ISSN 2723-6471

BERT's bidirectional contextual understanding to generate dynamic, context-aware embeddings that effectively capture word relationships. Fine-tuning the pretrained BERT model with BIO tagging schemes allowed for specialization in MWE detection without the need for additional sequential layers, achieving efficient and robust performance. Extending this approach, the BERT-CRF model incorporated a CRF layer, allowing it to capture sequential label dependencies more effectively than standalone BERT. The transformer encoder produced contextualized embeddings, which were then passed through the CRF layer to refine sequence predictions and model long-range dependencies between tokens. This enhancement improved coherence in MWE boundary identification by considering the dependencies between adjacent labels rather than treating token classifications independently. Both models were trained on the dataset containing 10,000 annotated MWEs and used the AdamW optimizer and cross-entropy loss for fine-tuning. Consistent preprocessing steps, such as tokenization, padding, and BIO tag alignment, ensured standardized input across experiments.

To explore alternative architectures, we implemented the RoBERTa-CRF and LSTM-CRF models. The RoBERTa-CRF model leveraged RoBERTa's robust contextual embeddings in conjunction with a CRF layer to improve sequence labeling by modeling inter-label dependencies. RoBERTa, with its dynamic masking and optimized pretraining, provided richer contextual representations than BERT, enhancing the model's ability to distinguish MWEs with subtle contextual shifts. The fine-tuning process followed a similar structure as BERT-CRF, where the CRF layer was added after the transformer encoder to optimize sequential token classification. Similarly, the LSTM-CRF model utilized bidirectional LSTM layers to capture both long-term and short-term word dependencies, with the CRF layer ensuring coherent label predictions. These models were also trained on the same 10,000 annotated sample, following rigorous preprocessing pipelines, including token alignment and padding to uniform sequence lengths, to facilitate consistency. Each model was fine-tuned with gradient-based optimization techniques and trained over multiple epochs, with metrics such as accuracy, precision, recall, and F1-score used to evaluate performance. Together, these methodologies integrate advanced language models and sequential dependency learning to achieve robust and efficient MWE detection.

To further explore models on a larger sample of 15,000 annotated MWEs which utilize the pre-trained GloVe embeddings that provide strong semantic foundations based on large-scale word co-occurrence statistics, the BiLSTM-GloVe model combines the GloVe embeddings with BiLSTM layers is hypothesized to be able to capture forward and backward dependencies, balanced by dropout regularization to mitigate overfitting. Lastly, the BiLSTM-BiGRU-GloVe model extended this approach by combining BiLSTM's capability to model long-range dependencies with BiGRU's efficiency in capturing short-term patterns, providing a comprehensive representation of sentence structure. Both GloVe-based models employed categorical cross-entropy loss and advanced optimization techniques, with the BiLSTM-BiGRU-GloVe model incorporating early stopping to ensure optimal training performance. Across all architectures, rigorous preprocessing steps, including token alignment, padding, and BIO tag conversion, ensured consistency. Systematic hyperparameter tuning and evaluation metrics like F1-score and accuracy enabled a fair comparison of model performance, showcasing the effectiveness of combining semantic and sequential dependency learning for MWE detection.

4.2. Model Training

To prevent overfitting and ensure optimal training, early stopping was applied across all models. This technique monitored validation loss, stopping training when performance ceased to improve over a set number of epochs. As a result, the reported number of epochs reflects the well-fitted values rather than arbitrarily chosen ones. All models in the 10,000 rows of annotated sample use a batch size of 16 and AdamW optimizer, but the learning rates vary, with BERT and RoBERTa models using 5e-5, while LSTM uses a higher 1e-3 learning rate. The number of epochs also differs, with BERT requiring only 2 epochs, while RoBERTa and LSTM-based models require 4 and 5 epochs, respectively.

For 15,000 annotated sample, the batch size is larger, with 32 for BERT models and 128 for BiLSTM models, reflecting the increased dataset size. The optimizer remains AdamW, but the learning rates differ—2e-5 for BERT, while BiLSTM-based models require higher learning rates (1e-3 to 1e-5, or 1e-4 for BiLSTM-GloVe-BiGRU). Transformer models converge faster, requiring only 5 epochs, whereas BiLSTM models require 15 epochs, indicating that RNN-based architectures need more training time.

This comparison highlights the fundamental trade-off between transformers and BiLSTM-based models. BiLSTM models process input sequentially, making them slower to train due to higher computational time. Since each step depends on the previous one, parallelization is limited, leading to longer convergence times. In contrast, transformer-based models, such as BERT and RoBERTa, utilize self-attention mechanisms that process tokens simultaneously, significantly reducing training time. Additionally, transformer models benefit from pretraining on large-scale corpora, allowing them to generalize more effectively and require fewer epochs for fine-tuning. BiLSTM models, on the other hand, need additional training epochs to effectively learn long-range dependencies, as they lack the extensive pretraining advantage of transformers.

4.3. Model Evaluation

The following discussion is to present the evaluation of the models developed and results of the model testing for both 10,000 rows (table 3) and 15,000 rows (table 4) of annotated sample.

Table 3. Model Evaluation for 10,000 rows of annotated sample

Model	Accuracy	Precision	Recall	F1-Score
BERT	86.28	52.17	58.21	55.03
BERT-CRF	84.99	49.02	57.45	52.90
RoBERTa-CRF	88.71	62.43	69.99	65.99
LSTM-CRF	85.77	48.80	48.37	48.59

Table 4. Model Evaluation for 15,000 rows of annotated sample

Model	Accuracy	Precision	Recall	F1-Score
BERT	57.14	80.00	51.61	63.16
BERT-BiLSTM	65.71	81.82	58.06	67.86
BiLSTM-GloVe	51.43	71.43	48.39	57.83
BiLSTM-GloVe-BiGRU	57.14	72.73	57.83	60.87

The evaluation of the results for the models trained on the sample containing 10,000 annotated rows demonstrate varying performance across accuracy, precision, recall, and F1-score metrics. These metrics were selected due to their relevance in token classification tasks like MWE detection. Precision and recall are crucial for assessing how well the model identifies MWEs while minimizing false positives and false negatives. The F1-score provides a balanced measure of both precision and recall, making it particularly useful when dealing with class imbalances while accuracy serves as a general performance indicator but is less informative in imbalanced datasets. Based on our study, the RoBERTa-based model combined with a CRF layer achieved the highest overall performance, with an accuracy of 88.71%, precision of 62.43%, recall of 69.99%, and an F1-score of 65.99%. The BERT-based model without a CRF layer performed slightly better than its CRF-enhanced counterpart in accuracy (86.28% vs. 84.99%) but showed lower F1-scores of 55.03% and 52.90%, respectively. This may be attributed to BERT's self-attention mechanism, which effectively captures contextual dependencies, reducing the need for additional sequence modeling from the CRF layer. Since MWE boundaries may already be well-defined within BERT's learned representations, the CRF's contribution can be minimal. Moreover, the CRF layer adds extra parameters and computational complexity, increasing the risk of overfitting, especially with a smaller dataset. The slight drop in accuracy for BERT-CRF suggests that the self-attention mechanism alone may be sufficient for MWE detection. Among the models, the LSTM+CRF architecture yielded the lowest scores across all metrics its sequential nature, weaker contextual representations, and limitations in modeling long-range dependencies, achieving an accuracy of 85.77% and an F1-score of 48.59%, highlighting its comparatively limited effectiveness for this task. These results emphasize the advantage of leveraging robust contextual embeddings, particularly those from transformer-based models like RoBERTa and BERT. Unlike static embeddings such as GloVe, which assign a fixed vector representation to each word regardless of context, contextual embeddings dynamically adjust word representations based on surrounding words. For example, in the sentence *"The bank raised interest rates"*, the word bank would receive different embeddings in *"He sat by the riverbank"* when using transformer models, as

self-attention mechanisms allow the model to consider surrounding words when computing embeddings. This contextual adaptation is crucial for MWE detection, as many MWEs exhibit polysemy or semantic shifts depending on their usage. Additionally, the incorporation of a CRF layer further enhances performance by enforcing label consistency, reducing independent token prediction errors, and modeling transition probabilities between labels, making predictions more structured and accurate in token classification tasks.

Whereas, the BERT-BiLSTM model outperformed all other architectures when trained on the 15,000 annotated sample, achieving the highest accuracy of 65.71%, with strong metrics across the board, including a precision of 81.82%, recall of 58.06%, and an F1-score of 67.86%. This can be explained by the fact that BERT provides deep contextualized word representations by leveraging bidirectional attention, which captures rich syntactic and semantic relationships. When combined with BiLSTM, which is well-suited for modeling sequential dependencies and long-range context, the model effectively learns patterns in MWE detection tasks. The BiLSTM layer helps retain sequential dependencies while mitigating BERT's potential shortcomings in handling long-range token dependencies, leading to improved recall and overall performance. In comparison, the BiLSTM-GloVe-BiGRU model achieved a moderate accuracy of 57.14%, precision of 72.73%, recall of 57.83%, and an F1-score of 60.87%, benefiting from the complementary strengths of BiLSTM and BiGRU but limited by GloVe's static embeddings. The BiLSTM-GloVe model showed similar limitations such as Glove embeddings are static, meaning that each word has a fixed representation regardless of the context in which it appears. This restricts the model's ability to differentiate between polysemous words, which is crucial for accurate MWE detection. The issue is particularly pronounced in code-mixed texts, where word meanings shift dynamically based on context, requiring embeddings that can adapt to varying linguistic structures. Since GloVe does not account for contextual variations, it struggles with MWEs that exhibit different meanings depending on surrounding words, leading to reduced detection accuracy. Therefore, it achieved an accuracy of 51.43%, precision of 71.43%, recall of 48.39%, and F1-score of 57.83%, demonstrating its ability to model sequential dependencies but lacking contextual flexibility. The BERT-based model, while achieving high precision (80.00%) and a strong F1-score of 63.16%, exhibited lower accuracy (57.14%) and recall (51.61%), indicating that additional sequential layers like BiLSTM are critical to fully leverage BERT embeddings. The results emphasize that combining dynamic contextual embeddings with robust sequential modeling is essential for maximizing performance in MWE detection tasks. This is because dynamic contextual embeddings adapt to different word meanings based on their surrounding context, making them more effective for NLP tasks that require nuanced understanding, such as MWE detection. However, transformer models alone may struggle with capturing sequential dependencies efficiently over long sequences. The addition of sequential architectures like BiLSTM and BiGRU ensures that the model maintains important positional and temporal information, leading to improved recall and overall performance in detecting MWEs more accurately.

5. Conclusion

To conclude based on our study, we investigated the effectiveness of various deep learning models for MWE detection using two samples comprising 10,000 and 15,000 annotated rows from the consumer electronic product retailer dataset. Our results demonstrate that combining dynamic contextual embeddings with robust sequential modeling significantly improves performance in token classification tasks. The RoBERTa-CRF model achieved the highest overall performance on the smaller dataset, highlighting the strength of RoBERTa’s contextual embeddings paired with the CRF layer's sequential dependency modeling. On the larger dataset, the BERT-BiLSTM model outperformed all other architectures, achieving the best balance between accuracy, precision, recall, and F1-score, demonstrating the advantage of integrating BERT's contextual embeddings with BiLSTM's sequential capabilities. Models utilizing GloVe embeddings, while effective in capturing semantic relationships, showed limitations due to their static nature, underlining the importance of dynamic embeddings for context-aware tasks.

While our findings highlight the effectiveness of deep learning architectures in MWE detection, certain limitations remain. The pretrained models used in this study were trained on general-domain corpora, which may not fully capture domain-specific MWEs such as legal or technical expressions, leading to misclassification or lower performance. Applying these models to specialized domains may require additional fine-tuning on domain-specific datasets or the integration of external knowledge sources. Future work can explore domain-adaptive pretraining and the incorporation of domain-specific embeddings to enhance performance in specialized linguistic contexts. In addition, deep learning

models require large amounts of training data to achieve optimal performance. Insufficient data can lead to poor generalization, reducing the model's ability to accurately detect MWEs, especially in low-resource settings. Overall, this study highlights the critical role of contextual and sequential learning in MWE detection and provides a foundation for future research aimed at improving performance on larger and more diverse datasets.

6. Declarations

6.1. Author Contributions

Conceptualization: J.M.W., Y.J.T., and T.M.L.; Methodology: Y.J.T.; Software: J.M.W.; Validation: J.M.W., Y.J.T., and T.M.L.; Formal Analysis: J.M.W., Y.J.T., and T.M.L.; Investigation: J.M.W.; Resources: Y.J.T.; Data Curation: Y.J.T.; Writing Original Draft Preparation: J.M.W., Y.J.T., and T.M.L.; Writing Review and Editing: Y.J.T., J.M.W., and T.M.L.; Visualization: J.M.W.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

We sincerely acknowledge the invaluable support and guidance provided by Lim Tong Ming, whose insightful feedback and expertise significantly contributed to the development of this article. Additionally, we appreciate the financial assistance from Tunku Abdul Rahman University of Management and Technology, which played a crucial role in supporting this work. We are also thankful to our colleagues and peers for their constructive discussions and encouragement throughout the research process. Lastly, we express our heartfelt appreciation to our families and friends for their unwavering support and motivation.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] M. Constant, "Multiword Expression Processing: A Survey," *Computational Linguistics*, vol. 43, no. 4, pp. 837–892, Dec. 2017,

[2] T. Jisha and M. Thomas, "Identification of Multiword Expressions: A Literature Study." *Computational Methods, Communication Techniques and Informatics*, vol. 2019, no. 1, pp. 385-389, 2019.

[3] W. Gharbieh, V. Bhavsar, and P. Cook, "Deep Learning Models For Multiword Expression Identification," *Association for Computational Linguistics*, vol. 2017, no. 1, pp. 1-7, 2017.

[4] J. Villena-Román, S. Collada-Pérez, S. Lana-Serrano, and C. González, "Hybrid Approach Combining Machine Learning and a Rule-Based Expert System for Text Categorization.," *Proceedings of the Twenty-Fourth International Florida Artificial Intelligence Research Society Conference*, vol. 2011, no. Jan., pp. 1-7, Jan. 2011,

[5] S. Islam, "A comprehensive survey on applications of transformers for deep learning tasks," *Expert Systems with Applications*, vol. 2023, no. Nov., pp. 122666–122666, Nov. 2023,

[6] M. Jogin, Mohana, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva, "Feature Extraction using Convolution Neural Networks (CNN) and Deep Learning," *IEEE Xplore*, vol. 2018, no. 1, pp. 1-7, 2018.

[7] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, no. 61, pp. 85–117, Jan. 2015,

[8] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks," *IEEE Xplore*, vol. 2015, no. Apr., pp. 1-7, 2015.

[9] I. Goodfellow et al., “Generative adversarial networks,” *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, Oct. 2020,

[10] S. Chen and W. Guo, “Auto-Encoders in Deep Learning—A Review with New Perspectives,” *Mathematics*, vol. 11, no. 8, pp. 1777-1789, Apr. 2023,

[11] D. Premasiri and T. Ranasinghe, “BERT(s) to Detect Multiword Expressions,” *arXiv preprint*, vol. 2022, no. Aug, pp. 1-12, 2022.

[12] A. Savary, “PARSEME Corpus Release 1.3,” *Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023)*, vol. 19, no. May, pp. 24-35, 2023.

[13] dimsum16, “GitHub - dimsum16/dimsum-data: Data for the DiMSUM shared task at SEMEVAL 2016,” GitHub, Dec. 28, 2015.

[14] C. Ramisch, A. Walsh, T. Blanchard, and S. Taslimipoor, “A Survey of MWE Identification Experiments: The Devil is in the Details,” 2023.

[15] M. Monachini and M. Eskevich, “Dependency Trees in Automatic Inflection of Multi Word Expressions in Polish,” *CLARIN Annual Conference Proceedings*, vol. 2021, no. 1, pp. 6-11, 2021.

[16] W. Gharbieh, V. Bhavsar, and P. Cook, “Deep learning models for multiword expression identification,” *in Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, vol. 2017, no. 1, pp. 54–64, 2017. doi: 10.18653/v1/S17-1006.

[17] S. Taslimipoor, O. Rohanian, and L. A. Ha, "Cross-lingual transfer learning and multitask learning for capturing multiword expressions," *in Proceedings of the Joint Workshop on Multiword Expressions and WordNet (MWE-WN 2019), Florence, Italy*, vol. 2019, no. Aug., pp. 155-161, 2019.

[18] A. Aziz, Md. Akram Hossain, Abu Nowshed Chy, Md. Zia Ullah, and M. Aono, “Leveraging contextual representations with BiLSTM-based regressor for lexical complexity prediction,” *Natural Language Processing Journal*, vol. 5, no. 1, pp. 100039–100039, Dec. 2023,

[19] K. D. Garg et al., “Framework for Handling Rare Word Problems in Neural Machine Translation System Using Multi-Word Expressions,” *Applied Sciences*, vol. 12, no. 21, pp. 11-38, Oct. 2022,

[20] R. Vacareanu, M. Valenzuela-Escaŕcega, R. Sharp, and M. Surdeanu, “An Unsupervised Method for Learning Representations of Multi-word Expressions for Semantic Classification,” *Proceedings of the 28th International Conference on Computational Linguistics*, vol. 28, no. 1, pp. 3346-3356, 2020.

[21] Mahtab Sarlak, Yalda Yarandi, and Mehrnoush Shamsfard, “Predicting Compositionality of Verbal Multiword Expressions in Persian,” *Conference on Language Resources and Evaluation (LREC)*, vol. 1, no. 1, pp. 14–23, Jan. 2023,

[22] R. Swaminathan and P. Cook, “Token-level identification of multiword expressions using pre-trained multilingual language models,” *in Proceedings of the 19th Workshop on Multiword Expressions (MWE 2023)*, vol. 2023, no. 1, pp. 1–6, 2023. doi: 10.18653/v1/2023.mwe-1.1.

[23] D. Premasiri, A. H. Haddad, T. Ranasinghe, and R. Mitkov, “Transformer-based Detection of Multiword Expressions in Flower and Plant Names,” *arXiv (Cornell University)*, vol. 2022, no. 1, pp. 1-12, Jan. 2022,

[24] M. Piasecki and K. Kanclerz, "Non-Contextual vs Contextual Word Embeddings in Multiword Expressions Detection," in Computational Collective Intelligence. ICCCI 2022, N. T. Nguyen, Y. Manolopoulos, R. Chbeir, A. Kozierkiewicz, and B. Trawiński, Eds., Lecture Notes in Computer Science, vol. 13501, Cham: Springer, 2022, pp. 1-12. doi: 10.1007/978-3-031-16014-1_16.

[25] G. Berend, “l1 Regularization of Word Embeddings for Multi-Word Expression Identification,” *Acta Cybernetica*, vol. 23, no. 3, pp. 801–813, 2018,