KAConv-Text: Kolmogorov-Arnold Convolutional Networks for Burmese Sentence Classification

Anonymous ACL submission

Abstract

This paper is the first study of the application of Kolmogorov-Arnold Convolutional Networks for Text data (KAConv-Text) on Burmese sentence classification across three tasks: hate speech detection (imbalanced binary), news classification (balanced multiclass), and ethnic language identification (imbalanced multiclass). Various embedding configurations were tested, utilizing random embeddings and fastText in both static and fine-tuned settings. Experiments were conducted with embedding dimensions of 100 and 300, comparing the CBOW and Skip-gram algorithms. Baseline models included Convolutional Neural Networks (CNN) and CNNs enhanced with a Kolmogorov-Arnold Network (KAN) classification layer (CNN-KAN). The proposed KAConv-Text with fine-tuned fastText embeddings achieved the best results, with 91.23% accuracy and a weighted F1-score of 0.9109 for hate speech detection, 92.66% accuracy and a weighted F1-score of 0.9267 for news classification, and 99.82% accuracy and a weighted F1-score of 0.9982 for ethnic language identification respectively.

1 Introduction

Text classification is a key part of natural language processing (NLP). For low-resource languages like Burmese, this task is challenging due to issues such as unbalanced datasets, different dialects, and a lack of pre-trained language models. Traditional methods, like text classification with Convolutional Neural Networks (CNNs) (Kim, 2014), use linear transformations with fixed activation functions. While useful, these methods often have trouble capturing complex patterns in text, especially in languages with rich grammar and structure. This paper introduces a new approach to text classification using Text Kolmogorov-Arnold Convolutional Networks (KAConv-Text) that uses spline-based non-linearities to improve flexibility and efficiency.

Recent advancements in deep learning have seen the integration of advanced mathematical frameworks into neural architectures. Among these, Kolmogorov-Arnold Networks (KANs) (Liu et al., 2024) stand out by leveraging the Kolmogorov-Arnold representation theorem (Schmidt-Hieber, 2021), (Selitskiy, 2022) to replace linear weight matrices with learnable spline functions. Bodner et al. (2024) originally proposed integration of KAN concept with convolution for computer vision tasks studying the performance of image classification on MNIST and Fashion MNIST datasets [1].

In this paper, we adapted KAN principles to convolutional layers for text data, experimenting KAConv-Text that replace traditional one dimensional CNN kernels with spline-parameterized functions. This innovation allows the model to dynamically learn non-linear mappings directly from data, circumventing the rigidity of fixed activation functions. Inspired by SplineCNN (Fey and Lenssen, 2018), which demonstrated splines' efficacy in geometric deep learning, our work extends these ideas with KAConv-Text to sequential text data, avoiding the need for graph-based preprocessing.

In this study, we trained and evaluated the proposed KAConv-Text across three Burmese sentence classification tasks: hate speech detection (imbalanced binary), news classification (balanced multiclass), and ethnic language identification (imbalanced multiclass). Our experiments demonstrate that the proposed KAConv-Text, when paired with fine-

[1] https://github.com/AntonioTepsich/Convolutional-KANs

tuned fastText embeddings, achieve state-of-the-art performance: 91.23% accuracy (weighted F1: 0.9109) for hate speech detection, 92.66% accuracy (F1: 0.9267) for news categorization, and 99.82% accuracy (F1: 0.9982) for ethnic language identification. These results highlight KAConv-Text's ability to perform well in both imbalanced and balanced sentence classification.

2 Corpus Preparation

We prepared sentence classification datasets for three tasks: hate speech detection, news classification, and ethnic language identification. Table 1 shows the distributions of labels in each dataset.

Each dataset was divided randomly into an 80:20 ratio for each class for training and testing. Cleaned, word-segmented, in-house Burmese monolingual corpus with 256,792 sentences and 5,981,381 tokens is used to train fastText embeddings (Bojanowski et al., 2016) for the news classification and hate speech detection experiments. For ethnic language identification experiments, syllable-segmented ethnic multilingual corpus with 200,781 sentences and 2,755,734 tokens is used to train fastText embeddings.

Hate Speech Detection: This imbalanced dataset comprises 10,140 annotated syllable-segmented sentences (8,493 hateful and 1,647 non-hateful), sourced from Myanmar social media platforms and public forums. Hate speech can have different categories such as abuse, religious bias, racism, body-shaming, political animosity, sexism, potentially lethal content, educational bias (Kyaw et al., 2024b). For binary classification, we considered all categories of hate speech into a single "hate speech" class and retained the non-hateful content as the "non-hate speech" class.

News Classification: The news corpus contains 7,315 word-segmented news sentences evenly distributed across six categories. For news classification, we used the corpus, which includes six news categories - Sports, Politics, Technology, Business, Entertainment, and Environmental (Kyaw et al., 2024a).

Ethnic Language Identification: This dataset spans syllable-segmented nine ethnic language sentences which mostly share Burmese alphabets, totaling 108,016 sentences. Data was collected from the web sources and the previous machine translation researches [Oo et al. (2018), Oo et al. (2020a), Htun et al. (2021), Kyaw et al. (2020), Aye et al. (2020), Linn et al. (2020), Thu et al. (2019), Oo et al. (2020b), Oo et al. (2019), Myint Oo et al. (2019), Oo et al. (2020c), Thu et al. (2022)].

Dataset	Class	Count	Percentage
Hate Speech	Hate	8,493	83.76%
	Non-Hate	1,647	16.24%
News	Sports	1,232	16.84%
	Politics	1,228	16.79%
	Technology	1,224	16.73%
	Business	1,221	16.69%
	Entertainment	1,205	16.47%
	Environment	1,205	16.47%
Language	Burmese	19,519	18.07%
	Beik	3,385	3.13%
	Dawei	3,537	3.27%
	Mon	5,854	5.42%
	Pa'o	10,346	9.58%
	Po Kayin	10,031	9.29%
	Rakhine	9,778	9.05%
	S'gaw Kayin	36,300	33.61%
	Shan	9,266	8.58%

Table 1: Label Counts and Percentages of Sentence Classification Datasets

3 Methodologies

We explore diverse embedding strategies, including randomly initialized vectors and pre-trained fastText embeddings under both static and fine-tuned configurations. The models we explored for experiments include baseline models, standard CNNs, and CNNs with KAN classification layers (CNN-KAN), along with the proposed KAConv-Text. The architectural designs are illustrated in Figure 1.

3.1 Embeddings

Word embeddings map discrete tokens to continuous vectors, capturing semantic and syntactic relationships. We evaluate two initialization strategies:

- **Random Embeddings (Rand)**: Each token w_i in the vocabulary $\mathcal{V}$ is assigned a randomly initialized embedding vector $\mathbf{e}_i \in \mathbb{R}^d$, where d is the embedding dimension. The embedding matrix $\mathbf{E}_{\text{rand}} \in$

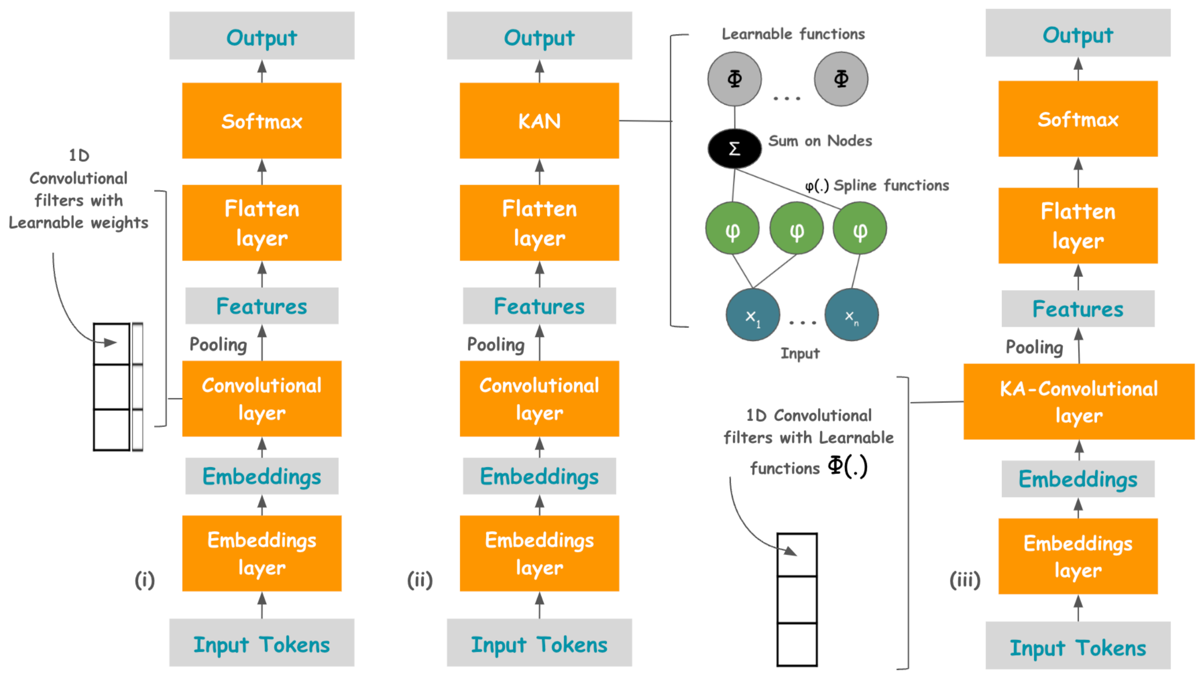


Figure 1: Architectural variations of experimented models: (i) Standard CNN, (ii) CNN-KAN with classification layer adaptation, and (iii) proposed KAConv-Text

$\mathbb{R}^{|\mathcal{V}|\times d}$ is learned during training: $\mathbf{X} = \mathbf{E}_{\text{Rand}}[w_1, w_2, \ldots, w_L]^\top \in \mathbb{R}^{L\times d}$ where L is the sequence length. This approach requires no prior linguistic knowledge but demands sufficient training data to learn meaningful representations.

- **fastText Embeddings**: We leverage pre-trained fastText embeddings $\mathbf{E}_{\text{fastText}} \in \mathbb{R}^{|\mathcal{V}|\times d}$, trained on subword n-grams. fastText represents words as bags of character n-grams, which allows it to capture subword information and handle out-of-vocabulary (OOV) words effectively. We investigated two approaches—static and fine-tuned—using embedding dimensions of 100 and 300, with both CBOW and Skip-gram models.

  1. **Static Embeddings**: The embeddings remain fixed during training. Words, including those not found in the vocabulary (which are initialized with random uniform values in [-0.25, 0.25]), are represented by their corresponding fastText embeddings. The embeddings themselves do not change during training.
  2. **Fine-tuned Embeddings**: The fastText embeddings are updated during training through backpropagation. This allows the embeddings to adapt to the specific domain of the task. The embedding vectors are refined to better capture context-specific nuances and semantics: $\mathbf{E}_{\text{fastText}} \leftarrow \mathbf{E}_{\text{fastText}} - \eta\nabla_{\mathbf{E}}\mathcal{L}$ where η is the learning rate and $\mathcal{L}$ is the loss function. Fine-tuning adapts the embeddings to domain-specific contexts.

3.2 Kolmogorov-Arnold Networks

3.2.1 Kolmogorov-Arnold Representation Theorem

The Kolmogorov-Arnold Representation Theorem (KART) (Kolmogorov, 1956) simplifies complex mathematical functions. Suppose a function f takes multiple inputs $(x_1, x_2, \ldots, x_n)$ and produces a single output. The theorem states that *any smooth function* of multiple variables can be represented using simpler functions as follows:

1. **Process each input individually**: For each input x_p, apply a simple single-variable function $\phi_{q,p}$.

2. **Combine results**: Add the outputs of these single-variable functions.

3. **Final adjustment**: Apply another set of single-variable functions Φ_q to the combined sums.

Mathematically, this is written as:

$$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right) \tag{1}$$

where $\phi_{q,p}$ are simple functions that handle each input x_p individually and Φ_q are functions that adjust the combined results to reconstruct $f(\mathbf{x})$. Instead of using fixed activation functions (like ReLU), KANs use these adaptive $\phi_{q,p}$ and Φ_q functions to learn complex patterns.

3.2.2 KAN Architecture

Kolmogorov-Arnold Networks (KANs) extend the classical Kolmogorov-Arnold representation by using learnable activation functions along graph edges. In a KAN layer, an input with n_{in} dimensions is transformed into an output with n_{out} dimensions. This layer is defined by a collection of functions: $\Phi = \{\phi_{q,p}\}, \quad p = 1, 2, \ldots, n_{\text{in}}, \quad q = 1, 2, \ldots, n_{\text{out}}$, where each $\phi_{q,p}$ is a learnable spline function.

The output of the layer is computed by applying these functions to the input features. For the lth layer, the output is given by:

$$\mathbf{x}^{(l+1)} = \sum_{i=1}^{n_l} \phi_{l,j,i}(x_i^{(l)}), \quad j = 1, \ldots, n_{l+1}. \tag{2}$$

In matrix notation, this operation is written as: $\mathbf{x}^{(l+1)} = \Phi^{(l)}\mathbf{x}^{(l)}$, where $\Phi^{(l)}$ is the function matrix for the lth layer.

In particular, the approximation error for KANs is bounded by

$$\left\| f - \left(\Phi_G^{(L-1)} \circ \cdots \circ \Phi_G^{(0)} \right) \mathbf{x} \right\|_{C^m} \leq C\, G^{-k-1+m}, \tag{3}$$

where G is the grid size, k is the order of the B-spline, and C is a constant. The notation $\Phi_G^{(L-1)} \circ \cdots \circ \Phi_G^{(0)}$ indicates that the output of one function matrix is fed as the input to the next.

3.3 Integrating with Convolution

3.3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are originally built for visual tasks with two-dimensional convolutions (LeCun et al., 1989). In our paper, the baseline CNN architecture processes embeddings $\mathbf{X}$ with one-dimensional convolutions, followed by pooling layer and a linear classifier. For a kernel $\mathbf{W} \in \mathbb{R}^{k \times d \times C}$ with C output channels:

$$\mathbf{H}_c = \text{ReLU} \left(\sum_{m=1}^{k} \sum_{n=1}^{d} \mathbf{W}_{c,m,n} \cdot \mathbf{X}_{i+m-1,n} + b_c \right) \tag{4}$$

where b_c is the bias term. The final feature representation is pooled and classified via

$$\hat{y} = \text{Softmax}(\mathbf{W}_{\text{cls}}\mathbf{h} + \mathbf{b}_{\text{cls}}). \tag{5}$$

3.3.2 KAN as classification layer

CNN-KAN approach was introduced shortly after the development of Kolmogorov-Arnold Networks (KAN) to enhance their performance in visual tasks (Cang et al., 2024). CNN-KAN models combine convolutional feature extraction with the expressive, spline-based nonlinearity of Kolmogorov-Arnold Networks (KAN) to better handle spatially structured visual data. In this paper, standard one-dimensional convolutional layers first perform feature extraction, producing feature maps that capture local text patterns. These feature maps are then passed through a KAN-inspired activation, where the conventional linear transformation is augmented with spline-parameterized non-linearities.

3.3.3 Kolmogorov-Arnold Convolution for Text

The idea of KAN is extended to one-dimensional convolution by replacing the standard dot product with a learnable nonlinear function defined via B-splines. In this approach, unlike its use in Computer Vision (Azam and Akhtar, 2024), a KAConv-Text kernel is represented as Kernel $= [\phi_1, \phi_2, \ldots, \phi_K]$ where each ϕ_m is a B-spline function with learnable parameters. The output at position i is computed by

$$y_i = \sum_m \phi_m(x_{i+m}) \tag{6}$$

with x_{i+m} denoting the input at index $i+m$.

Following the original proposal(Liu et al., 2024), each learnable function is defined by

$$\Phi(x) = w_1 \cdot \text{spline}(x) + w_2 \cdot b(x), \qquad (7)$$

where $b(x)$ is a fixed basis function Parametric ReLU (PReLU) and spline(x) is a linear combination of B-spline basis functions. The extracted feature representation is pooled and classified using softmax layers.

3.3.4 Interpretability

CNN-KAN improves interpretability by replacing the final linear classifier with a KAN layer. The spline-based activation functions in KAN are learnable and can be visualized as univariate functions, revealing how input features are nonlinearly transformed before classification. Unlike fixed activations in CNNs, these splines adapt to the data, allowing to be inspected how specific feature ranges influence class probabilities. The convolutional layers in CNN-KAN remain unchanged, retaining CNN's lack of transparency in feature extraction. As a result, while the classification stage becomes more interpretable, the extraction of intermediate positional patterns remains unclear. **KAConv-Text** enhances interpretability at the convolutional level by replacing linear kernels with learnable spline functions. Each kernel operates as a transparent, univariate nonlinear transformation, whose B-spline parameters can be directly visualized to understand its response to input. Figure 2 visualizes the learned B-spline surfaces for each convolutional kernel, where the X-axis represents spline coefficients, the Y-axis denotes individual filters, and the Z-axis reflects weight values, thereby offering an intuitive view of how these nonlinear transformations shape feature responses.

4 Experimental Setup

In this study, we explore a range of embedding strategies, including randomly initialized vectors and pre-trained fastText embeddings. These embeddings are evaluated in both static and fine-tuned configurations, with dimensionalities of 100 and 300. The fastText embeddings are trained using the CBOW and Skipgram algorithms.

We trained and evaluated three models: CNN, CNN-KAN, and KAConv-Text. The models were evaluated using accuracy and weighted F1-scores. For each model, we employ a three-layer architecture, with successive channel dimensions of 64, 128, and 256. The kernel sizes for these layers are set to 3, 4, and 5, respectively. ReLU activation is applied after each convolutional layer for CNN and CNN-KAN models, and an adaptive average pooling operation is used for pooling. In the case of KAConv-Text, we incorporate cubic splines for the convolutions, with the polynomial degree of the B-spline basis functions set to three.

For the classification part, the linear layers - Softmax for CNN and KAConv-Text, and KAN for CNN-KAN were used. All models were trained for 10 epochs with a dropout rate of 0.3 to mitigate overfitting. The Adaptive Moment Estimation (Adam) optimizer was used for model optimization, with a learning rate of 1e-3.

For model training, we utilized the Kaggle environment[2], leveraging an NVIDIA P100 GPU with 16GB of memory for efficient computation. The experiments were conducted using PyTorch[3] framework developed by Paszke et al. (2019).

5 Results and Discussion

This section examines the impact of different embedding settings on each model across various tasks and compares their performance based on accuracy and weighted F1-scores for the best-performing models in each setting.

5.1 Comparison of Different Embedding settings

For Hate Speech Detection, the Figure 3 (a) indicate that finetuned embeddings again lead to improved performance. The CNN model using 100-dimensional finetuned Skipgram embeddings obtained an F1-score of 0.9023 while the KAConv-Text model with 300-dimensional finetuned CBOW embeddings recorded the best results (0.9109 F1).

For the News classification task, the Figure 3 (b) reveal that finetuned embeddings

[2]https://www.kaggle.com/
[3]https://pytorch.org/

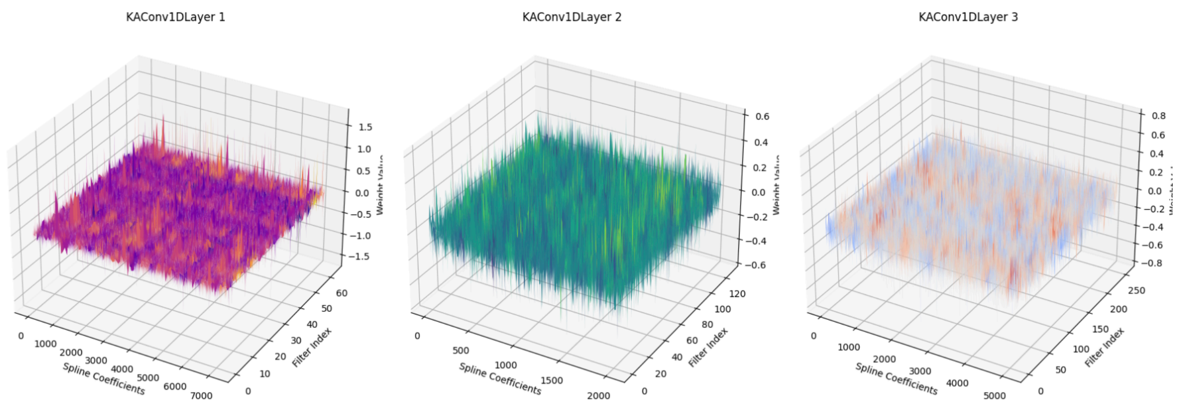


Figure 2: 3D visualization of the learned B-spline surfaces for each KAConv-Text convolutional kernel in fine-tuned fastText with dimension 300 and Skip-gram + KAConv-Text model for Hate Speech Detection

Embed	Model	Hate Speech		News		Language	
		Acc (%)	F1	Acc (%)	F1	Acc (%)	F1
Rand	CNN	88.96	**0.8895**	88.17	0.8823	**99.73**	**0.9973**
	CNN-KAN	88.81	0.8851	89.26	0.8932	**99.73**	**0.9973**
	KAConv-Text	**88.62**	0.8819	**89.67**	**0.8969**	99.71	0.9971
Static	CNN	89.90	0.9001	91.23	0.9124	99.71	0.9971
	CNN-KAN	**90.29**	**0.9017**	91.30	0.9134	99.72	0.9972
	KAConv-Text	89.16	0.8855	**91.50**	**0.9147**	**99.73**	**0.9973**
Fine-tuned	CNN	90.24	0.9023	91.98	0.9196	99.78	0.9978
	CNN-KAN	89.40	0.8946	91.77	0.9179	99.79	0.9979
	KAConv-Text	**91.23**	**0.9109**	**92.66**	**0.9267**	**99.82**	**0.9982**

Table 2: Accuracy (Acc) and Weighted F1-scores (F1) comparison of the best models across different settings on Hate Speech Detection (Binary), News (Multiclass), and Language Identification (Multiclass) classification tasks. Bolded results are the best result for each task across each embedding type.

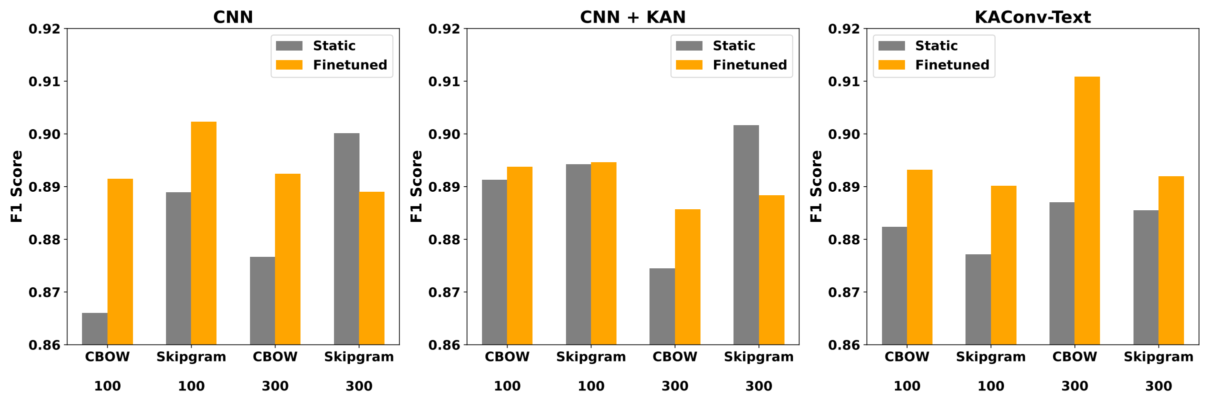
consistently yield superior results. The CNN model with 100-dimensional Skipgram embeddings achieved the highest performance, reaching an weighted F1-score of 0.9196. Meanwhile, the CNN-KAN model performed best with 100-dimensional finetuned CBOW embeddings (0.9179 F1), and the KAConv-Text model excelled with 300-dimensional finetuned Skipgram embeddings (0.9267 F1).

In the Language Identification task, the Figure 3 (c) demonstrate near-perfect performance across all models. The highest scores were observed with the 300-dimensional finetuned Skipgram setting, where the CNN model reached an F1-score of 0.9978 and the KAConv-Text model achieved 0.9982.

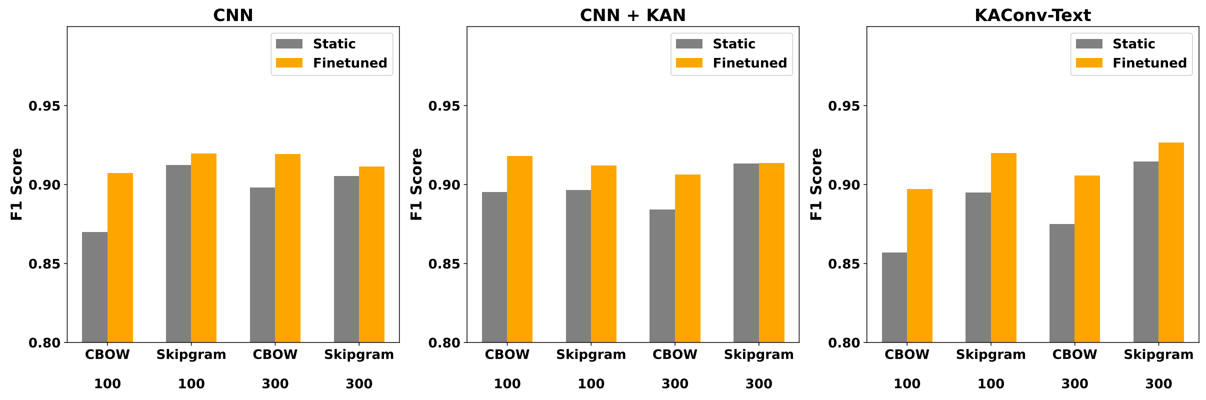
5.2 Comparison of Best Models Across Each Setting

The performance of CNN, CNN-KAN, and KAConv-Text models across three text classification tasks is summarized in Table 2. Overall, KAConv-Text with fine-tuned fastText embeddings achieved the strongest results, demonstrating the advantage of integrating KAConv-Text layers with trainable fastText embeddings.

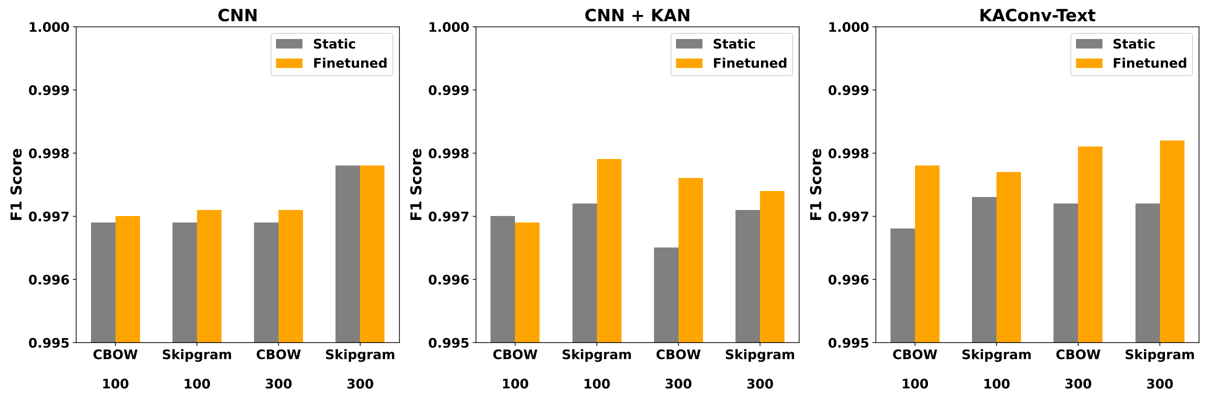
For **random embeddings (Rand)**, CNN and KAConv-Text exhibited task-dependent strengths. While CNN achieved the highest F1-score (0.8895) on Hate Speech detection, KAConv-Text outperformed others on News classification (89.67% Acc, 0.8969 F1). All



(a) F1-scores for Hate Speech Detection Task



(b) F1-scores for News Classification Task



(c) F1-scores for Language Identification Task

Figure 3: F1 Scores of Different Embeddings Settings - Dimensions (100, 300) and Algorithms (CBOW, Skip-gram) for Each Task. The figure shows the importance of finetuning and selecting appropriate embedding dimensions for maximizing model performance across different tasks. We selected best performing settings for each model across each task for further evaluation.

models achieved near-perfect results ($>$ 99.7% Acc) on Language Identification, suggesting simpler patterns suffice for this task regardless of architecture.

With **static fastText embeddings**, CNN-KAN excelled in Hate Speech detection (90.29% Acc, 0.9017 F1), whereas KAConv-Text dominated News classification (91.50% Acc, 0.9147 F1) and Language Identification (99.73% Acc). This highlights KAN's potential to enhance task-specific feature extraction when combined with fixed embeddings.

The most significant improvements emerged with **fine-tuned fastText embeddings**:

KAConv-Text surpassed CNN and CNN-KAN across all tasks, achieving 91.23% Acc (Hate Speech), 92.66% Acc (News), and 99.82% Acc (Language). The marked gains in Hate Speech (+0.99% Acc over CNN) and News (+0.68% Acc) suggest KAConv-Text combines effectively with trainable embeddings to capture nuanced textual patterns. CNN-KAN underperformed CNN in these tasks, implying that KAN as the classification layer may insufficiently leverage fine-tuned embeddings.

These results indicate that KAConv-Text's full integration of KAN concept with convolutions enhances adaptability to complex, embedding-dependent features. The marginal differences in Language Identification across models ($<$ 0.1% Acc) further suggest that architectural advantages diminish for highly separable classes. Overall, KAConv-Text emerges as the **most robust model**, particularly when paired with fine-tuned embeddings, demonstrating the efficacy of KAN in learning task-specific hierarchical representations.

6 Conclusion and Future Work

In this work, we present the first study to investigate the application of KAConv-Text layers in text classification. Our contributions include (1) three novel cleaned text classification datasets and (2) a KAConv-Text model that achieves state-of-the-art results, outperforming CNN architectures across tasks. Our findings show that integrating these layers with fine-tuned fastText embeddings significantly enhances classification performance, with the proposed KAConv-Text model consistently outperforming other architectures across the evaluated tasks.

In the future, we plan to extend our evaluation to a more diverse set of languages and datasets for cross-linguistic and cross-domain generalizability, including low-resource scenarios. Additionally, we will investigate the integration of contextual embeddings (e.g., BERT, LLM-derived features) with KAConv-Text to further improve classification accuracy while preserving model transparency. We will also release the implementation, experimental logs, and the text classification corpus for the community.

Limitations

Despite the promising results, our study has some limitations. Notably, our experiments were conducted on a limited number of datasets focused on the Burmese language, which may not fully capture the challenges posed by other languages or more diverse text classification tasks such as code-mixed text. This constraint limits the generalizability of our findings, and further research is needed to validate the proposed approach across broader linguistic and domain-specific contexts.

Ethics Statement

This research utilizes three classification datasets to develop and evaluate our proposed approach. The datasets used in this study will be publicly available and have been curated following ethical guidelines, including anonymization and compliance with relevant regulations, especially the Hate Speech dataset.

We acknowledge the sensitive nature of the Hate Speech data and have taken necessary precautions to ensure ethical data handling. We recognize the risks associated with biases in the datasets especially, the hate speech dataset and have conducted an analysis by university students who are native speakers to assess and minimize these biases.

References

Hnin Yi Aye, Yuzana Win, and Ye Kyaw Thu. 2020. Statistical machine translation between myanmar (burmese) and chin (mizo) language. In *Proceedings of The 23rd Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (Oriental COCOSDA 2020)*, pages 211–216.

Basim Azam and Naveed Akhtar. 2024. Suitability of kans for computer vision: A preliminary investigation. *ArXiv*. Https://arxiv.org/abs/2406.09087.

Alexander D. Bodner et al. 2024. Convolutional kolmogorov-arnold networks. *ArXiv*. Accessed 11 Feb. 2025.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Yueyang Cang et al. 2024. Can kan work? exploring the potential of kolmogorov-arnold networks in computer vision. Accessed 12 Feb. 2025.

Matthias Fey and Jan Eric Lenssen. 2018. Splinecnn: Fast geometric deep learning with continuous b-spline kernels. *ArXiv preprint arXiv:1711.08920.*

Hay Man Htun, Ye Kyaw Thu, Hlaing Myat Nwe, May Thu Win, and Naw Naw. 2021. Statistical machine translation system combinations on phrase-based, hierarchical phrase-based and operation sequence model for burmese and pa'o language pair. volume 6, pages 1–20.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

A. N. Kolmogorov. 1956. On the representation of continuous functions of several variables by superpositions of continuous functions of a smaller number of variables. *Proceedings of the USSR Academy of Sciences*, 108:179–182. English translation: Amer. Math. Soc. Transl., 17 (1961), pp. 369–373.

Eaint Kay Khaing Kyaw, Thura Aung, and Ye Kyaw Thu. 2024a. An empirical study of simple text augmentation on news classification for myanmar language. *Preprint.*

Nang Aeindray Kyaw, Ye Kyaw Thu, Hlaing Myat Nwe, Phyu Phyu Tar, Nandar Win Min, and Thepchai Supnithi. 2020. A study of three statistical machine translation methods for myanmar (burmese) and shan (tai long) language pair. In *2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, pages 1–6.

Nang Aeindray Kyaw, Ye Kyaw Thu, Thazin Myint Oo, Hutchatai Chanlekha, Manabu Okumura, and Thepchai Supnithi. 2024b. Enhancing hate speech classification in myanmar language through lexicon-based filtering. In *2024 21st International Joint Conference on Computer Science and Software Engineering (JCSSE)*, pages 316–323.

Yann LeCun, Bernhard Boser, John S. Denker, David Henderson, R. E. Howard, William Hubbard, and L. D. Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

Zar Zar Linn, Ye Kyaw Thu, and Pushpa B. Patil. 2020. Statistical machine translation between myanmar (burmese) and kayah languages. volume 6, pages 62–68.

Ziming Liu et al. 2024. Kan: Kolmogorov-arnold networks. *ArXiv.* Accessed 11 Feb. 2025.

Thazin Myint Oo, Ye Kyaw Thu, and Khin Mar Soe. 2019. Neural machine translation between myanmar (burmese) and rakhine (arakanese). In *Proceedings of the Sixth Workshop on NLP for Similar Languages, Varieties and Dialects*, pages 80–88.

Thazin Myint Oo, Ye Kyaw Thu, and Khin Mar Soe. 2018. Statistical machine translation between myanmar (burmese) and rakhine (arakanese). In *Proceedings of ICCA2018*, pages 304–311.

Thazin Myint Oo, Ye Kyaw Thu, Khin Mar Soe, and Thepchai Supnithi. 2019. Statistical machine translation between myanmar (burmese) and dawei (tavoyan). In *The First Workshop on NLP Solutions for Under Resourced Languages (NSURL 2019).*

Thazin Myint Oo, Ye Kyaw Thu, Khin Mar Soe, and Thepchai Supnithi. 2020a. Neural machine translation between myanmar (burmese) and dawei (tavoyan). In *Proceedings of the 18th International Conference on Computer Applications (ICCA 2020)*, pages 219–227.

Thazin Myint Oo, Ye Kyaw Thu, Khin Mar Soe, and Thepchai Supnithi. 2020b. Statistical machine translation between myanmar and myeik. In *Proceedings of the 12th International Conference on Future Computer and Communication (ICFCC 2020)*, pages 36–45.

Thazin Myint Oo, Ye Kyaw Thu, Khin Mar Soe, and Thepchai Supnithi. 2020c. Statistical machine translation of myanmar dialects. *Journal of Intelligent Informatics and Smart Technology*, 2020(April 1st Issue):14–26. Submitted December 21, 2019; accepted March 6, 2020; revised March 18, 2020; published online April 30, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703.* Accessed 14 Feb. 2025.

Johannes Schmidt-Hieber. 2021. The kolmogorov–arnold representation theorem revisited. *Neural Networks.*

Stanislav Selitskiy. 2022. Kolmogorov's gate nonlinearity as a step toward much smaller artificial neural networks. In *Proceedings of the 24th International Conference on Enterprise Information Systems (ICEIS 2022)*. University of Bedfordshire, School of Computer Science and Technology.

Ye Kyaw Thu, Thazin Myint Oo, and Thepchai Supnithi. 2022. Reinforcement learning fine-tuning for improved neural machine translation of burmese dialects. In *Workshop of Multimodal, Multilingual and Multitask Modeling Technologies for Oriental Languages (M3Oriental)*, pages 1–8, Tainan, Taiwan.

Ye Kyaw Thu, Manar Hti Seng, Thazin Myint Oo, Dee Wom, Hpau Myang Thint Nu, Seng Mai, Thepchai Supnithi, and Khin Mar Soe. 2019. Statistical machine translation between kachin and rawang. In *Proceedings of the 14th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP 2019)*, pages 329–334.