Burmese SpellGraph: Learning a Task-Specific Syllable Error Graph for Myanmar Spelling Correction

Ye Kyaw Thu[1,2,*], Thazin Myint Oo[2] and Thepchai Supnithi[1]

[1]Core-AI Research Team, NECTEC, Thailand
[2]Language Understanding Lab, AI Mario Company Ltd, Myanmar

Abstract

Automatic spelling correction remains difficult for Burmese because user-generated text contains heterogeneous orthographic errors, while curated spelling resources and annotated training data remain limited. We study whether a compact, training-derived error graph can provide information complementary to sequential contextual modeling. Burmese SpellGraph represents syllables as graph nodes and learns relations from erroneous–corrected pairs, with explicit spelling-confusion edges and optional local co-occurrence edges. A lightweight GraphSAGE-style encoder injects graph representations into a Transformer encoder, while an auxiliary error detector predicts source positions affected by edits. We evaluate six supervision levels, three random seeds, graph-relation ablations, maximum-length sensitivity, support-aware error-patterns, and real model outputs. In the matched seed-42 comparison with the same auxiliary detector, SpellGraph improves syllable-position accuracy at five of six training fractions and reduces mean edit distance at all six fractions; at full supervision it gains 0.80 percentage points in position accuracy and 0.58 lower mean edit distance. Three-seed results show that gains are clearest at medium and full supervision rather than in the extreme low-resource regime. At full supervision, error-confusion-only edges outperform co-occurrence-only and the full mixed graph. A final diagnostic partitions 20,507 gold substitution events into retained error edges, observed-but-pruned pairs, and unseen pairs. It shows a significant positive retained-edge effect at full supervision but a significant negative effect at 50%, demonstrating that graph usefulness depends on resource quality rather than simple memorization of observed confusions. We release the experiment code and reproducibility scripts.

Keywords

Myanmar NLP, Burmese spelling correction, knowledge graphs, error graphs, graph neural networks, GraphSAGE, low-resource NLP, syllable-level correction

CEUR-WS.org/Vol-4316/pd-short-6.pdf

1. Introduction

Spelling correction is a basic text-normalization capability for search, text analytics, messaging, information retrieval, and downstream language technologies. Yet the problem is particularly challenging for languages for which training data, lexical resources, and mature correction benchmarks are limited. Burmese (Myanmar) is a representative case: noisy social-media writing contains non-word errors, real-word errors, phonetic or cognitive misspellings, typographic variation, repeated characters, and context-dependent mistakes. Earlier Myanmar spelling systems therefore combined dictionaries, edit distance, n-gram language models, and rule-based processing rather than relying on a single universal correction mechanism [6, 7, 8].

Recent neural work has moved toward contextual sequence modeling. Transformer architectures provide a strong sequence-to-sequence foundation [1], while Soft-Masked BERT explicitly couples error detection with correction [5]. Published Myanmar studies have likewise examined error-type classification and Transformer-based spelling correction, including explicit error-type features [9, 10]. These approaches capture sentence context well, but they do not explicitly represent the recurring relations between erroneous syllables and their observed corrections.

Graph-based spelling correction suggests a complementary direction. SpellGCN encodes phonological and visual similarity relations for Chinese spelling correction [4]; TextGCN demonstrates graph-

IJCKG 2026 Posters, Demos and Workshops, November 19–21, 2026, Bangkok, Thailand
*Corresponding author.
yekyaw.thu@nectec.or.th (Y. K. Thu); queenofthazin@gmail.com (T. M. Oo); thepchai.supnithi@nectec.or.th (T. Supnithi)
© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073
published 2026-11-06

based text representation using co-occurrence and document-word relations [3]; and GraphSAGE provides an inductive neighborhood-aggregation mechanism [2]. For Burmese, an attractive property of a graph is that its relations can be induced directly from the parallel correction resource, avoiding the need for external pronunciation or keyboard databases.

This paper asks a focused question: *Can a task-specific syllable error graph complement a Transformer for Burmese spelling correction, and when is the graph actually useful?* We treat the graph as a lightweight knowledge representation: syllable nodes are connected by typed relations whose semantics are directly tied to the correction task. This is a narrower notion than a conventional entity-centric knowledge graph, but it fits the broader knowledge-graph setting of relation learning and graph representation.

The study is organized around five questions. First, does graph augmentation improve correction quality across different supervision levels? Second, does an auxiliary error detector remain useful when graph information is added? Third, are spelling-confusion relations more informative than generic co-occurrence? Fourth, do graph benefits concentrate on training-supported error relations or rare source syllables? Fifth, can final model outputs reveal systematic correction and failure patterns that aggregate metrics alone would hide? Recent fine-grained spelling-correction work motivates the last question because coarse benchmark scores can obscure important error-type and multi-error behavior [11].

Our contributions are fourfold: (i) a compact syllable-level Burmese spelling model that augments Transformer encoding with a training-derived error graph; (ii) a controlled relation ablation separating explicit spelling-confusion and local co-occurrence information; (iii) a multi-level, three-seed evaluation and support-aware diagnostic that makes graph behavior visible rather than relying on one aggregate score; and (iv) a reproducible qualitative analysis based on actual held-out model outputs, together with public code for training, evaluation, and analysis.

2. Related Work

2.1. Myanmar Spelling Correction

Myanmar spelling correction has been studied with dictionary lookup, edit distance, language modeling, and rule-based methods. Early work addressed common non-word and real-word errors [6], while an n-gram system combined detection, candidate generation, and sentence-level correction [7]. SymSpell4Burmese introduced efficient symmetric-delete candidate generation together with Burmese training data and frequency dictionaries [8]. More recent published work examined Myanmar spelling-error classification [9] and Transformer-based correction with explicit error-type features [10]. Our approach instead learns relations directly from observed erroneous-to-corrected syllable pairs.

2.2. Neural and Graph-based Correction

The Transformer provides a general contextual sequence model [1]. Soft-Masked BERT couples error detection with correction [5]. SpellGCN shows that graph relations such as phonological and visual similarity can augment spelling correction [4]; TextGCN demonstrates graph learning over textual relations [3]; and GraphSAGE provides inductive neighborhood aggregation [2]. Our graph differs in granularity and provenance: its nodes are Burmese syllables and its primary edges are empirical spelling-confusion relations induced from correction pairs.

2.3. Fine-grained Analysis

Recent Chinese spelling-correction research argues that aggregate scores can hide systematic weaknesses and motivates fine-grained evaluation [11]. We adopt the same principle at the mechanism level by separating retained error relations, observed-but-pruned relations, and unseen pairs.

3. Burmese SpellGraph

Figure 1 gives an implementation-level view of Burmese SpellGraph. Given an erroneous syllable sequence $x = (x_1, \ldots, x_n)$ and corrected target $y = (y_1, \ldots, y_m)$, the model generates y and predicts an auxiliary binary tag for source syllables participating in edits. The graph is a learned relation resource over syllable types, not a graph database: its neighbor and weight tables are constructed from training data and then used by the neural model.

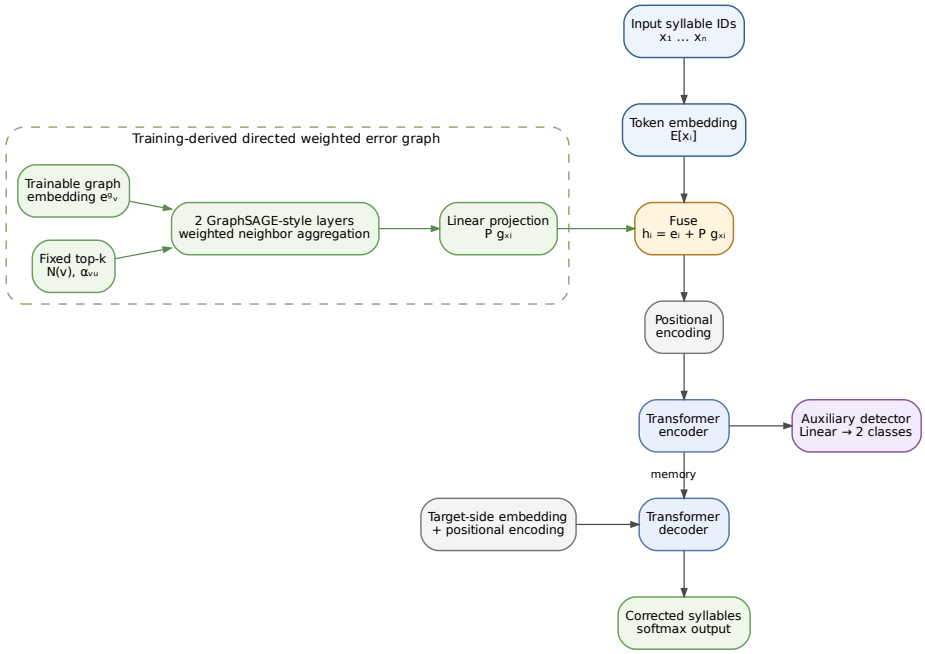


Figure 1: Implementation-level Burmese SpellGraph architecture. Input syllables use both standard token embeddings and a training-derived directed weighted graph representation. A two-layer GraphSAGE-style module aggregates fixed top-k neighbors, the projected graph vector is added before positional encoding, and the shared Transformer encoder feeds both the correction decoder and the auxiliary error detector.

3.1. Task-specific Error Graph

Each syllable is a graph node. The implementation constructs a directed weighted graph from the selected training pairs. An error-confusion edge connects a source syllable x_i to corrected syllable y_j when an alignment contains a substitution, assigning an initial weight of 3.0. Optional co-occurrence edges are added in both directions for tokens within the local context window using inverse-distance weighting. For each source node, outgoing edge weights are normalized and the top $k = 8$ neighbors are retained. Vocabulary and graph resources are rebuilt from only the selected training subset; validation and test examples never create graph edges.

This graph can be viewed as a lightweight linguistic knowledge graph: the nodes are syllables and the edge semantics describe observed spelling transformations or local usage. It is not an entity-centric knowledge graph, but it provides an explicit relational representation that can be learned and queried at the syllable level.

Table 1
Effective training resources after subset sampling.

Fraction	Training pairs	Source vocab.	Seen test pairs
10%	4,413	3,219	15,388
25%	11,032	4,690	16,782
50%	22,063	6,539	17,686
100%	44,126	9,096	18,400

3.2. Graph-aware Encoding and Joint Learning

Each syllable has a trainable graph embedding e_v^g. The implementation applies two lightweight GraphSAGE-style layers. With $h_v^{(0)} = e_v^g$, layer l combines the current self representation with a weighted aggregation of the fixed neighbors:

$$h_v^{(l)} = \sigma\left(W_s^{(l)} h_v^{(l-1)} + W_n^{(l)} \sum_{u \in N(v)} \alpha_{vu} e_u^g\right), \quad l = 1, 2. \tag{1}$$

The second-layer output is projected to the Transformer dimension and added to the standard syllable embedding before positional encoding,

$$h_i = e_i + P h_{x_i}^{(2)}. \tag{2}$$

The decoder is trained with autoregressive cross-entropy and the detector with binary cross-entropy:

$$\mathscr{L} = \mathscr{L}_{corr} + \lambda \mathscr{L}_{det}, \qquad \lambda = 0.5. \tag{3}$$

The matched Transformer+Det baseline uses the same detector loss, so the main graph comparison changes the representation rather than the auxiliary objective.

4. Experimental Design

4.1. Corpus and Splits

The corpus consists of noisy user-generated Burmese sentences paired with manually corrected forms and represented as whitespace-separated syllable tokens. The fixed split contains 44,126 training pairs, 9,455 validation pairs, and 9,455 test pairs. The training source length distribution has mean 27.33 syllables, median 20, and 95th percentile 75; the maximum is 371. A maximum sequence length of 128 covers approximately 99.25% of training sentences.

4.2. Models and Training

The Transformer uses four encoder layers and four decoder layers, model dimension 256, four attention heads, feed-forward dimension 1024, dropout 0.1, batch size 32, learning rate 5×10^{-4}, and weight decay 10^{-4}. Training runs for at most 20 epochs with early stopping patience 4. The graph dimension is 128, top-$k = 8$, and co-occurrence window 4.

We train six supervision conditions: 1%, 5%, 10%, 25%, 50%, and 100%. The main scaling study uses three seeds (13, 42, 87). For the focused matched and fine-grained analyses, seed 42 is used to keep the model pair exactly controlled. The plain Transformer has no detector loss; the matched Transformer+Det adds the same detector objective as SpellGraph. Graph relation ablations isolate error-confusion edges and co-occurrence edges.

4.3. Metrics

We report sentence exactness, syllable-position accuracy, and mean edit distance for correction. Detection precision, recall, and F1 are reported for the auxiliary detector. Position accuracy captures local correction quality while sentence exactness remains a strict end-to-end measure.

Table 2
Three-seed results (mean ± standard deviation). Exactness and position accuracy are percentages.

Train	Exact		Position		Edit	
	Tr.	SG	Tr.	SG	Tr.	SG
1%	0.00±0.00	0.00±0.00	1.81±0.05	1.67±0.02	33.40±1.24	38.81±3.87
5%	0.11±0.01	0.07±0.01	3.51±0.06	3.54±0.20	28.51±0.07	29.42±1.57
10%	0.27±0.05	0.21±0.04	4.51±0.19	4.44±0.26	25.51±1.25	26.68±2.78
25%	0.73±0.04	0.80±0.16	5.89±0.30	5.91±0.20	23.68±1.27	23.61±2.23
50%	1.98±0.13	2.14±0.04	7.43±0.07	7.54±0.34	22.45±0.53	21.48±0.69
100%	3.63±0.22	3.92±0.21	10.22±0.20	10.89±0.37	20.11±0.01	20.66±0.22

Table 3
Relation and detection-loss ablations at 100% training data, seed 42. Exact and position values are percentages.

Variant	Exact	Pos.	Edit	Det. F1
Full graph	3.98	11.15	20.13	0.798
Error-confusion only	**4.03**	**11.52**	**19.81**	0.790
Co-occurrence only	3.87	10.99	20.93	0.786
Graph, corr.-only	4.02	10.60	20.61	–
Transformer+Det	3.72	10.34	20.71	0.792

4.4. Final Support-aware Diagnostic

The final no-retraining analysis uses only held-out test substitutions from complete sentences whose source and target lengths satisfy the 128-syllable limit. It covers 9,380 of 9,455 test sentences and 20,507 gold substitution records. Each substitution is assigned to one of three mutually exclusive groups: (i) *error-edge retained*, when the exact source-to-target confusion relation is present after top-k pruning; (ii) *seen but pruned*, when the exact pair was observed in training but is absent from the final neighbor list; or (iii) *unseen pair*, when the exact confusion did not occur in the selected training subset. The same substitution masks are applied to SpellGraph and Transformer+Det. We use 4,000-sample sentence-level paired bootstrap intervals.

5. Results

5.1. Overall Supervision Scaling

Table 2 summarizes the three-seed comparison. SpellGraph is not universally better in the extreme low-resource regime: at 1%–10%, differences are small or unstable. From 25% upward, it is competitive or better on exactness and position accuracy. At 100%, SpellGraph reaches 3.92% sentence exactness and 10.89% position accuracy versus 3.63% and 10.22% for the plain Transformer. Mean edit distance is mixed across seeds, with the graph slightly worse on average at full supervision, so we do not claim universal improvement across all metrics.

The matched comparison is more diagnostic than the plain Transformer comparison. At seed 42, Transformer+Det and SpellGraph use the same auxiliary detector and differ primarily by graph augmentation. SpellGraph reduces mean edit distance at all six training fractions and improves position accuracy at five of six settings. At full supervision, position accuracy rises from 10.34% to 11.15%, while mean edit distance falls from 20.71 to 20.13.

5.2. Relation Ablation and Detection

The ablation is one of the clearest results in the paper. Error-confusion-only edges outperform co-occurrence-only edges on all three correction metrics at full supervision and even slightly outperform the full mixed graph. This suggests that relation *quality and task specificity* matter more than graph density. The detection objective remains strong: full SpellGraph obtains 0.798 F1 compared with 0.792 for Transformer+Det.

Table 4
Support-aware Graph minus Transformer+Det repair-rate difference in percentage points. Asterisk indicates a 95% paired bootstrap interval excluding zero.

Train	Repair-rate difference			Test substitution share (%)		
	Retained	Pruned	Unseen	Retained	Pruned	Unseen
10%	+0.27	**-0.70***	+0.23	61.9	13.2	25.0
25%	-0.22	-0.06	+0.08	65.9	15.9	18.2
50%	**-0.82***	+0.39	+0.04	68.9	17.3	13.8
100%	**+0.65***	+0.16	+0.09	70.9	18.8	10.3

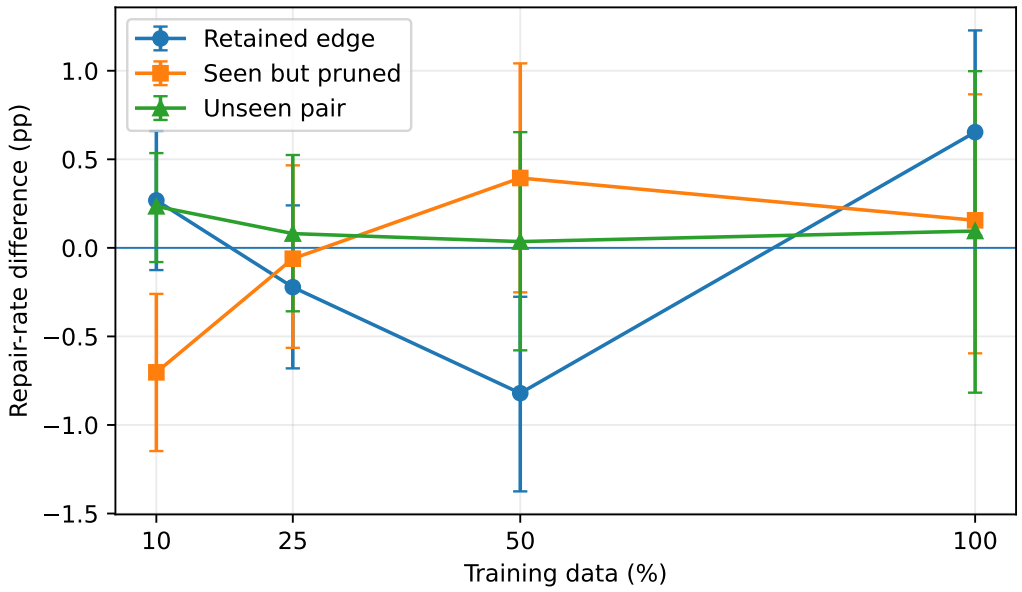


Figure 2: Repair-rate differences by graph-support status. Error bars show paired bootstrap 95% intervals. The sign changes across supervision levels emphasize resource sensitivity rather than a uniformly positive graph effect.

5.3. Maximum-length Sensitivity

The 100% seed-42 sensitivity study compares maximum lengths of 64, 96, 128, 192, and 256. The 64-syllable setting has the lowest edit distance but evaluates a different, shorter-sentence population. Longer limits do not consistently improve exactness or position accuracy; we therefore retain 128 as the main efficiency–coverage trade-off.

5.4. Support-aware Error Analysis

The final diagnostic makes the resource sensitivity more explicit. Table 4 reports repair-rate differences between SpellGraph and Transformer+Det on the same substitution events.

At 100%, the exact error relation is retained for 70.9% of substitution events and SpellGraph improves their repair rate by 0.65 points (95% CI [0.07, 1.23]). The corresponding difference for observed-but-pruned pairs is only 0.16 points and the unseen-pair difference is 0.09 points, both statistically indistinguishable from zero. This is consistent with the idea that well-supported explicit relations can help when the resource is sufficiently large.

However, the effect is not monotonic. At 50%, retained relations account for 68.9% of substitutions but SpellGraph is 0.82 points worse (95% CI [-1.37, -0.28]). At 10%, the negative effect moves to the seen-but-pruned group (-0.70 points), while the retained-edge and unseen groups remain statistically inconclusive. Thus, graph support alone does not explain model behavior. The graph appears to require a sufficiently rich surrounding resource, not merely the existence of one observed confusion.

Table 5
Real contrastive correction examples from the 100% seed-42 test run. G/T marks the system that exactly matches the gold sentence.

Case	Input	Gold	Graph	Tr.+Det
G1	ဟုက် ပ	ဟုတ် ပ	ဟုတ် ပ	ဟုတ် ပါ့
G2	ဖား သယ်	ဖား သည်	ဖား သည်	သူ ငယ် ချင်း
G3	ဟယ် သူ ဂ	ဟယ် သူ က	ဟယ် သူ က	ဟယ် သူ က သူ က
G4	ခင် မင် လျှက်	ခင် မင် လျက်	ခင် မင် လျက်	မင် ခင် လျက်
G5	စက် ရက် တယ်	ရက် စက် တယ်	ရက် စက် တယ်	ရက် ရက် စက် တယ်
T1	ကြွေ p	ကြွေ ပြီ	ကြွေ ပြီး ကြွေ ပြီ	ကြွေ ပြီ
T2	တျောက် လောက်	တစ် ယောက် လောက်	တစ် ယောက် လောက် လောက်	တစ် ယောက် လောက်
T3	ရိုက် ပေး ဘာ	ရိုက် ပေး ပါ	ဘာ ရိုက် ပေး ပါ	ရိုက် ပေး ပါ
T4	ဥာဏ် ကောင်း ချက်	ဥာဏ် ကောင်း ချက်	ဥာဏ် ဥာဏ် ဥာဏ် ကောင်း ချက် ဥာဏ် ဥာဏ် ဥာဏ် ဥာဏ် ကောင်း ချက် ကောင်း ချက်	ဥာဏ် ကောင်း ချက်
T5	ချိန်း တေ့ တာ ပေါ့	ချိန်း တွေ့ တာ ပေါ့	ချိန်း ချိန်း တာ တော့ ပေါ့	ချိန်း တွေ့ တာ ပေါ့

This result is useful for the knowledge-graph interpretation. A relation inventory is not equivalent to a useful relational representation: graph pruning, neighborhood composition, and the amount of surrounding evidence can all affect whether an observed relation becomes a productive inductive bias.

5.5. Real Correction Outputs

To complement aggregate scores, we extracted deterministic contrastive examples from the 100% seed-42 held-out test run. Among 9,271 eligible sentences, 153 are Graph wins, 129 are Transformer+Det wins, and 222 are corrected exactly by both systems. Table 5 shows ten representative cases; the complete deterministic 10+10 example files are included in the released experiment package.

The examples illustrate complementary failure modes rather than a universal winner. Graph wins often suppress an extra insertion or reordering produced by the baseline, while Transformer+Det wins include cases where graph augmentation leads to repetition or misplaced syllables. Because the table is contrastively selected, these examples are qualitative rather than frequency estimates.

6. Discussion

The matched comparison gives the cleanest evidence for graph utility because the detector objective is identical. SpellGraph improves position accuracy at five of six seed-42 fractions and lowers mean edit distance at all six. At full supervision, the position gain is 0.80 points and edit distance falls from 20.71 to 20.13. The three-seed study nevertheless shows that graph augmentation is not universally better at extreme low resource.

Relation ablation shows that task-specific knowledge is more valuable than graph density. Error-confusion-only edges outperform co-occurrence-only edges and slightly outperform the full mixed graph at 100%. The support-aware analysis then adds an important qualification: retained error edges give a significant +0.65 point repair advantage at 100%, but -0.82 points at 50%. Unseen-pair differences remain small. The graph therefore is not simply a memorization device; its usefulness depends on the surrounding resource and its interaction with sequential context.

The contrastive outputs make this behavior visible. Graph wins often avoid an extra insertion or reordering made by Transformer+Det, while Transformer+Det wins include cases where graph augmentation produces repetition or misplaced syllables. Because the examples are selected as wins, they are qualitative rather than frequency estimates. For knowledge-graph research, the central lesson is that relation semantics and support matter more than adding edges indiscriminately.

A direct next model is a support-aware error graph in which a confusion $x \rightarrow y$ receives a reliability weight based on its training count, e.g., $w_{err}(x, y) = \log(1 + c_{xy})$, with separate aggregation channels

for error-confusion and co-occurrence relations. The released evaluation code can then test whether any retained-edge advantage becomes stable across supervision levels.

Limitations include the use of one supervised Burmese resource, no controlled comparison with external phonological/visual/keyboard resources, and a substitution-focused fine-grained diagnostic. Detailed support-aware results use seed 42, while the main scaling study uses three seeds.

7. Conclusion

We presented Burmese SpellGraph, a syllable-level Transformer with a training-derived error graph and auxiliary detector. Matched experiments show gains at most supervision levels, with the clearest benefit at full supervision, while error-confusion-only edges outperform generic co-occurrence. Support-aware analysis shows resource-sensitive effects: retained relations help at full supervision but reverse at 50%, and unseen-pair differences remain small. Real outputs reveal complementary failure modes. These findings motivate support-aware, relation-specific graph construction for Burmese spelling correction and structured linguistic knowledge graphs.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT to assist with code development, experimental organization, literature-search support, and language polishing. The authors reviewed and edited the resulting material, verified the reported experimental evidence, and take full responsibility for the content of the paper.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[2] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs," in *Advances in Neural Information Processing Systems*, 2017, pp. 1024–1034.

[3] L. Yao, C. Mao, and Y. Luo, "Graph Convolutional Networks for Text Classification," in *Proc. AAAI*, vol. 33, no. 1, 2019, pp. 7370–7377.

[4] X. Cheng *et al.*, "SpellGCN: Incorporating Phonological and Visual Similarities into Language Models for Chinese Spelling Check," in *Proc. ACL*, 2020, pp. 871–881.

[5] S. Zhang, H. Huang, J. Liu, and H. Li, "Spelling Error Correction with Soft-Masked BERT," in *Proc. ACL*, 2020, pp. 882–890.

[6] A. M. Mon and T. Thein, "Myanmar Spell Checker," *International Journal of Science and Research*, vol. 2, no. 1, pp. 333–340, 2013.

[7] T. Win and W. P. Pa, "Myanmar Spelling Correction Based On N-grams Model," in *Proc. International Conference on Computer Applications*, 2021.

[8] E. P. P. Mon, Y. K. Thu, T. T. Yu, and A. W. Oo, "SymSpell4Burmese: Symmetric Delete Spelling Correction Algorithm (SymSpell) for Burmese Spelling Checking," in *Proc. iSAI-NLP*, 2021, pp. 1–6, doi: 10.1109/iSAI-NLP54397.2021.9678171.

[9] E. T. Phyu, Y. K. Thu, T. M. Oo, H. Chanlekha, and T. Supnithi, "Myanmar Spelling Error Classification: An Empirical Study of Tsetlin Machine Techniques," *Journal of Intelligent Informatics and Smart Technology*, vol. 10, pp. 1–11, 2024, doi: 10.14456/jiist.2024.1.

[10] E. T. Phyu, Y. K. Thu, H. Chanlekha, K. Funakoshi, and T. Supnithi, "Exploring the Impact of Error Type Features Integration on Transformer-based Myanmar Spelling Correction," in *Proc. JCSSE*, 2024, pp. 365–372, doi: 10.1109/JCSSE61278.2024.10613711.

[11] L. Liu, H. Wu, and H. Zhao, "Driving Chinese Spelling Correction from a Fine-Grained Perspective," in *Proc. COLING*, 2025, pp. 10727–10737.