Enhancing Hindi Feature Representation Through Fusion of Dual-Script Word Embeddings

Lianxi Wang[1,2], Yujia Tian[1*], Zhuowei Chen[✉1*]

[1]School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China
[2]Guangzhou Key Laboratory of Multilingual Intelligent Processing, Guangzhou, China

Introduction

Pretrained language models (PLMs) have revolutionized natural language processing (NLP), particularly in languages with ample resources.However, low-resource languages like Hindi face significant challenges due to the lack of processing tools, evaluation tasks, and datasets.

Hindi, written in both Devanagari and Roman scripts, poses unique complexities, influencing semantic interpretation. Leveraging insights from similar approaches in other languages, our research proposes a novel dual-script representation method for Hindi.

By combining features from both scripts, we aim to enhance the effectiveness of Hindi language models, improving performance and capturing richer semantic information. This approach, unprecedented in Hindi NLP, offers promising avenues for advancing language understanding and processing in this context.

The major contributions of this paper can be summarized as follows:

- We proposed a dual-script enhanced representation method for Hindi, which generates representation by fusing features from a single script Hindi Roberta.
- Incorporating four fusion techniques into the representation method effectively generates dual-script representation from single-script representation. These techniques include concatenation, addition, cross-attention, and convolutional neural networks.
- Evaluated the representation methods by three different categories of NLP tasks, i.e. sequence generation, text classification, and natural language inference. Results show the effectiveness of dual-script representation methods.

Experiment: Results

We evaluate our methods on three categories of NLP tasks. Namely, text classification, sequence generation, and natural language understanding. Specifically, we evaluate our model on five datasets for Text Classification (TC), Natural Language Inference (NLI), Part-of-Speech Tagging (POS Tagging), and Named Entity Recognition (NER).

- **Text Classification.**

Dataset	Model	AUC	ACC	F1
IITP-product-review	Devanagari	0.8878	0.7591	0.7062
	Romanized	0.8836	0.7457	0.7156
	Dual-CONCAT	**0.8904**	0.7438	0.7150
	Dual-ATT	0.8749	0.7514	0.7171
	Dual-ADD	0.8876	**0.7610**	**0.7350**
	Dual-CNN	0.8862	0.7457	0.7035
IITP-movie-review	Devanagari	0.7446	0.5355	0.5332
	Romanized	0.7318	0.5290	0.5209
	Dual-CONCAT	0.7408	0.5484	0.5390
	Dual-ATT	**0.7538**	0.5387	0.4368
	Dual-ADD	0.7427	0.5484	0.5532
	Dual-CNN	0.7371	**0.5903**	**0.5863**
bbc-articles	Devanagari	0.8439	0.7344	0.3670
	Romanized	0.8386	0.7356	0.3473
	Dual-CONCAT	0.8369	0.7240	0.3171
	Dual-ATT	0.8049	0.7471	0.3198
	Dual-ADD	**0.8779**	0.7564	0.3167
	Dual-CNN	0.8276	**0.7621**	0.3081

Table 1. Model performances on text classification task.

- **Natural Language Inference.**

Model	AUC	ACC	F1
Devanagari	0.6977	0.5026	0.4975
Romanized	**0.7234**	0.5343	0.5241
Dual-CONCAT	0.7127	0.5258	0.5212
Dual-ATT	0.7133	0.5279	**0.5247**
Dual-ADD	0.7112	0.5144	0.5074
Dual-CNN	0.7220	**0.5487**	0.5169
GPT-3.5-Turbo	–	0.4263	0.3911

Table 2. Model performances on NLI task.

- **Sequence Generation.**

Dataset	Model	ACC	F1
IJNLP-TFM-NER	Devanagari	0.7794	0.8576
	Romanized	0.6275	0.7651
	Dual-CONCAT	0.7735	0.8571
	Dual-ATT	0.6412	0.7539
	Dual-ADD	**0.7892**	**0.8726**
Hindi-HDTB-POS	Devanagari	0.9699	0.9702
	Romanized	0.9518	0.9520
	Dual-CONCAT	0.9698	0.9700
	Dual-ATT	0.8930	0.8935
	Dual-ADD	**0.9702**	**0.9704**

Table 3. Model performances on sequence generation tasks.

- **Training Time Cost.**

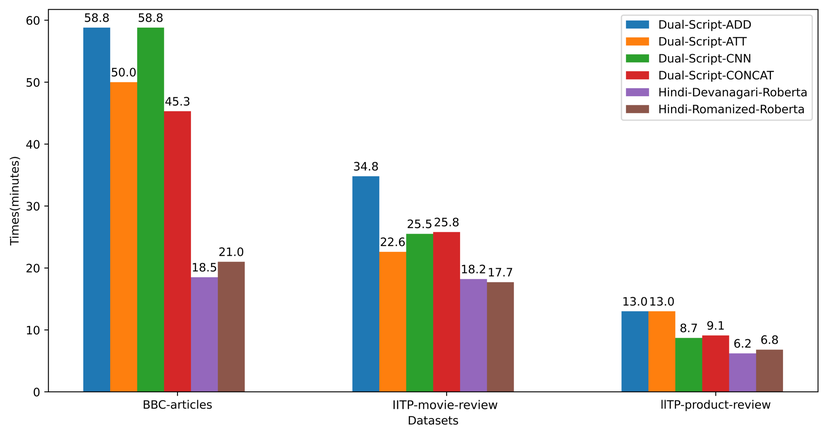

Figure 1. Training time cost of various models in text classification task.

Experiment: Discussions

In the proposed method for enhancing feature representation using dual scripts, we depart from the conventional approach where researchers typically combine various features from a single script within one language. Instead, we combine features from different scripts within the same language.

The experiment results across multiple datasets highlight the efficacy of the dual-script feature representation method, demonstrating superior performance compared to single-script approaches. Particularly in sequence generation tasks, the addition-based fusion method proves highly effective. While no single fusion method emerges as significantly superior for sequence classification tasks, dataset characteristics play a crucial role in determining optimal performance. Specifically, the findings underscore the potency of dual-script representation combined with addition-based fusion in sequence generation tasks, as well as its capacity to enhance performance in tasks like TC and NLI.

Furthermore, we assessed the performance of GPT-3.5-turbo on the Hindi NLI task, where it encountered significant challenges. This underscores the importance of developing language-specific expertise models for addressing such issues and shaping the future of language understanding AI.

Nonetheless, employing two PLMs to create a dual-script representation inevitably results in a noteworthy escalation in both inference and training time. Additionally, the heightened complexity and depth of the network introduce uncertainties and safety concerns.

Methodology: Enhanced Hindi Feature Representation

Figure 2 shows the overall process of the proposed method. The dual-script generation process typically consists of two main stages: **single-script representation generation** and **dual-script representation generation with feature fusion**.

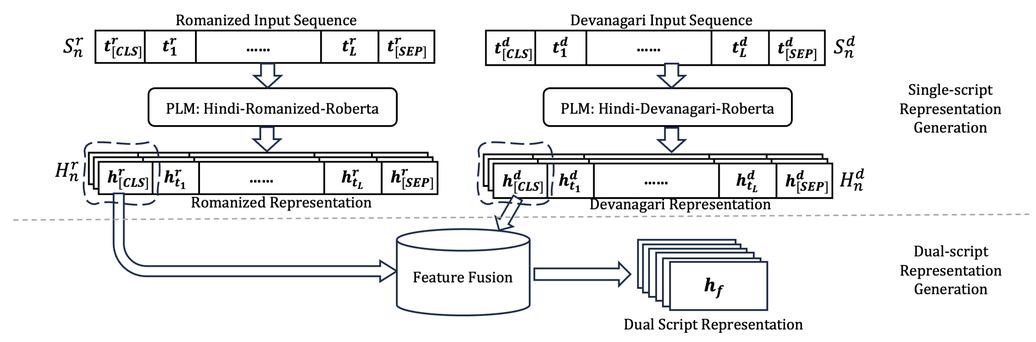

Figure 2. The dual-script sequence representation generation process.

In the single-script representation generation stage, we acquire separate single-script representations for Devanagari and Romanized Hindi text from pre-trained Roberta. In the subsequent dual-script representation generation stage, we merge these single-script representations using various fusion methods to obtain the dual-script enhanced Hindi feature representation.

Methodology: Dual-Script Representation Generation

Dual-script representation is generated by fusing single script sentence representation $h^r_{[CLS]}$ and $h^d_{[CLS]}$. To achieve a better performance of Hindi language representation by fusing features from models in two scripts, this paper introduces four effective fusion techniques, concatenation, addition, attention mechanism, and convolutional neural network (CNN).

- **Concatenation.** The representations from both scripts are straightforwardly concatenated to create a combined representation. The equation below illustrates the exact operation of these representations:

$$h_f = [h^d_{[CLS]}||h^r_{[CLS]}], \tag{1}$$

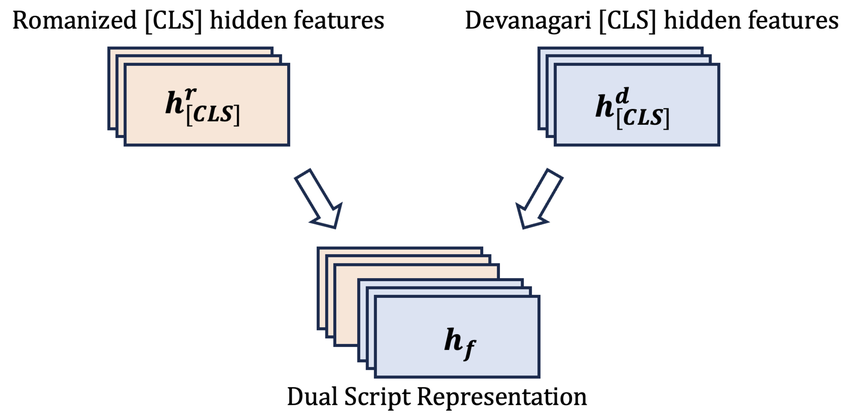

Figure 3. Concatenation feature fusion method.

- **Addition.** In this method, the Romanized text representation is initially linearly transformed using a weighted matrix W and a non-linear activation function σ. The resulting dual-script representation is obtained by adding the Devanagari representation $h^d_{[CLS]}$ to the output of the activation function. The mathematical representation of this fusion method is as follows:

$$h_f = h^d_{[CLS]} + \sigma(W \cdot h^r_{[CLS]}) + b, \tag{2}$$

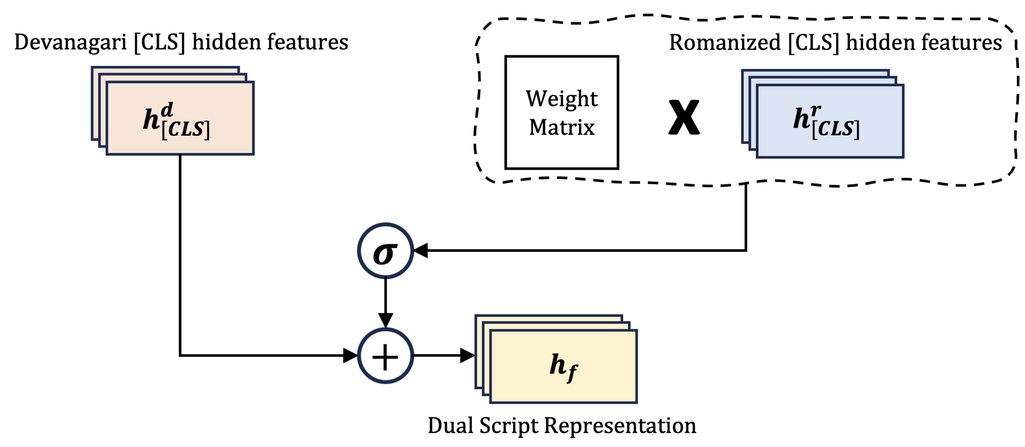

Figure 4. Addition feature fusion method.

- **Cross-attention.** In the attention fusion method, we apply the cross-attention mechanism. In our method, the Romanized text representation acts as the query while the Devanagari representation acts as key and value. Specifically, The calculation of the attention mechaism is defined as:

$$h_f = \text{Att.}(h^d_{[CLS]}, h^r_{[CLS]}, h^d_{[CLS]}), \tag{3}$$

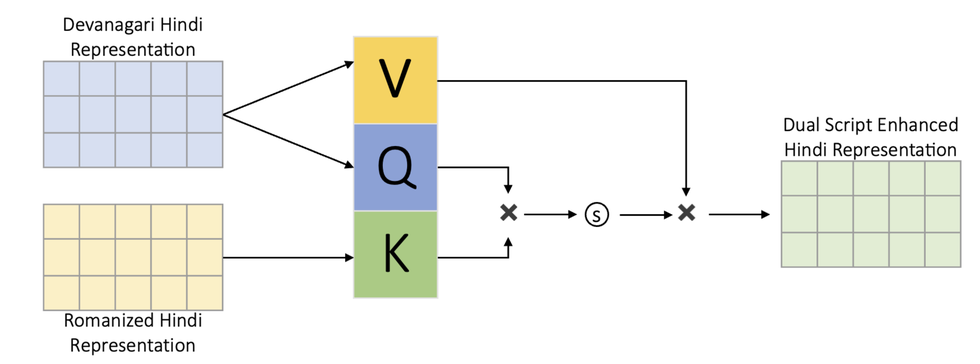

Figure 5. Cross-attention feature fusion method.

- **Convolutional Neural Networks.** After concatenation of $h^d_{[CLS]}$ and $h^r_{[CLS]}$, we further apply convolutional neural networks to extract high level features to achieve the goal of compressing information, reducing dimensions, and enhancing robustness.

$$h_f = \sigma(\text{Conv}([h^d_{[CLS]}||h^r_{[CLS]}], W)) + b, \tag{4}$$

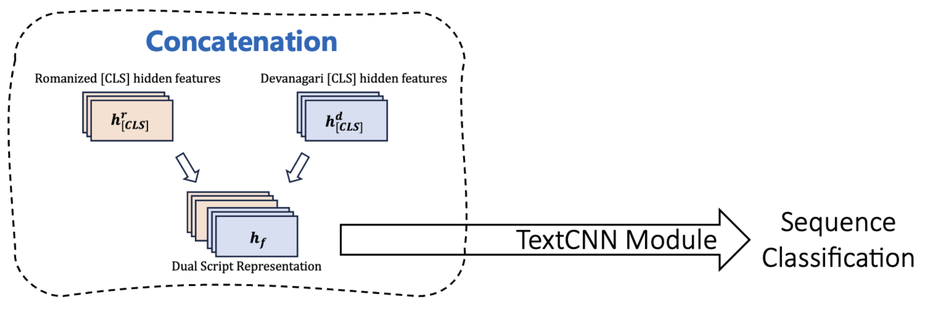

Figure 6. CNN feature fusion method.