Institute of Technology of Cambodia
Department of Applied Mathematics and Statistics

LSTM-based Khmer Text Style Transfer Using Representation Learning

Subject: Natural Language Processing (NLP)

Name of Students	**ID**
PEL Bunkhloem	e20201314
PHETH Soriyuon	e20210674
PHOEUN Rajame	e20211748
PHOEURN Kimhor	e20210823
PHORN Sreypov	e20210166
YIN Sambat	e20210138

Lecturers: **Dr. KHON Vanny (Course)**
Mr. TOUCH Sopheak (TP)

Academic Year 2025 - 2026

1. Problem Statement

The task is to transfer text from a **source style** to a **target style** while preserving content. Formally, let:

- $X = (x_1, x_2, \ldots, x_T)$ denote the input sequence in the source style.
- $Y = (y_1, y_2, \ldots, y_{T'})$ denote the target sequence in the desired style.
- f_θ be the function implemented by our seq2seq LSTM, parameterized by θ.

We aim to model the conditional distribution:

$$P(Y \mid X; \theta) = \prod_{t=1}^{T'} P(y_t \mid y_1, \ldots, y_{t-1}, X; \theta)$$

The objective is to find θ^* that maximizes the likelihood of the target sequences in the dataset $\mathcal{D}$:

$$\theta^* = \arg\max_\theta \sum_{(X,Y)\in\mathcal{D}} \log P(Y \mid X; \theta)$$

This is a **representation learning problem**, as the encoder LSTM learns hidden states h_t that represent the content of the input sequence in a way that can be decoded into the target style.

2. Proposed Solution

We use an **encoder-decoder LSTM** for Khmer style transfer, adopting a **pretraining + fine-tuning approach** to improve performance on limited paired data.

Pretraining

Before fine-tuning, we **pretrained a single LSTM** on a large corpus of general Khmer text. The pretraining task was **self-reconstruction** ($X \to X$), meaning that given a sequence $X = (x_1, x_2, \ldots, x_T)$, the model attempts to rewrite it.

- Pretraining loss:

$$\mathcal{L}_{pretrain}(\theta) = -\sum_{X\in\mathcal{C}} \sum_{t=1}^{T} \log P(x_t \mid x_1, \ldots, x_{t-1}; \theta)$$

- Here, θ includes all LSTM weights and biases.
- Purpose: learn **robust Khmer token embeddings and sequence representations**.

The pretrained LSTM weights are then **used to initialize the encoder and/or decoder** of the style-transfer encoder-decoder model, giving the model a good starting point.

Model Architecture

Encoder LSTM

The encoder reads the input normal-style sentence X and generates **hidden states** h_t and **cell states** c_t:

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
\tilde{c}_t &= \tanh(W_c x_t + U_c h_{t-1} + b_c) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

- c_t stores long-term information.
- h_t represents the content of the input sentence.
- Encoder weights can be initialized with pretrained LSTM weights.

Decoder LSTM

The decoder generates the royal-style target sequence $Y = (y_1, \ldots, y_{T'})$ token by token:

$$
P(y_t \mid y_{<t}, X; \theta) = \mathrm{Softmax}(W s_t + b)
$$

- s_t = decoder hidden state at step t
- Each token depends on previous outputs and encoder representations.
- Decoder weights can also be initialized with pretrained LSTM weights.

Fine-Tuning for Style Transfer

The fine-tuning loss is **negative log-likelihood** over the paired dataset $\mathcal{D}$:

$$
\mathcal{L}_{finetune}(\theta) = -\sum_{(X,Y)\in\mathcal{D}} \sum_{t=1}^{T'} \log P(y_t \mid y_1, \ldots, y_{t-1}, X; \theta)
$$

- Fine-tuning adapts pretrained weights θ to the **style transfer task**.
- The model learns to **rewrite normal Khmer sentences into royal style**, while preserving content.