Effect Analysis of Lexical Augmentation Techniques for Improving the Performance of Transformer-Based Korean Chatbots

September 2025

[12pt]article [utf8]inputenc times
Lexical Augmentation and Its Effect on Transformer-based Korean Chatbots
Author Name

Abstract

One of the biggest challenges in developing high-quality generative chatbots is securing enough data. Data Augmentation (DA) is often discussed as a solution, but in many cases it does not lead to real performance improvements. Especially in languages like Korean, where morphological variations are complex, simple augmentation methods may even cause performance degradation. This study quantitatively analyzes the effect of a context-ignorant augmentation method—Word2Vec-based Lexical Substitution—on the performance of a Transformer-based Korean chatbot. The same model structure was trained with both the original dataset and the augmented dataset, and results were compared mainly using the BLEU score (Bilingual Evaluation Understudy Score). As expected, the augmented data significantly reduced model performance. Qualitative analysis showed that simple word substitution often distorted grammatical elements such as particles and endings, or inserted contextually inappropriate words, producing noise that hindered learning. This study shows that in Korean dialogue generation tasks, the "quality" of data is more important than the "quantity." It also suggests that future research should focus on more sophisticated augmentation methods that consider contextual and linguistic characteristics, rather than simple substitution.

Recently, many studies in Korean NLP have actively used Transformer-based language models. Park et al. (2023), *Transformer-based Korean Pretrained Language Models: A Survey*, discussed the development of Korean-specific Transformer models such as KoBERT and KcBERT. The Encoder-Decoder Transformer model adopted in this study also belongs to the same research line of Korean dialogue models.

Along with model research, this paper highlights the limitations of Lexical Substitution as a method for performance improvement.

Vu et al. (2021), *Text Data Augmentation for the Korean Language*, compared several augmentation methods such as EDA and Back-Translation, showing that the effectiveness of augmentation varies depending on the task and dataset. They pointed out that for generative tasks, simple word substitution can distort meaning—a finding that matches our results.

Cho et al. (2022), *CHEF in the Language Kitchen*, proposed an augmentation method that reflects the morphological characteristics of Korean, showing that it can reduce the contextual mismatches caused by simple substitution. Their work directly relates to the noise problem of Lexical Substitution raised in this study, and emphasizes the need for more refined methods.

1 Method

For the experiment, we used the publicly available Korean chatbot question–answer dataset `ChatbotData.csv` (provided by Songys). The dataset contains 11,876 question–answer pairs, each consisting of an utterance, a response, and a label. Labels are divided into daily life (0), negative (1), and positive (2). The data was split into train (80%), validation (10%), and test (10%).

2 Lexical Augmentation

To address data scarcity, we applied Lexical Substitution using Word2Vec embeddings. This simple augmentation method randomly selects words in a sentence and replaces them with their most similar words, generating new sentences. We replaced two words per sentence, doubling the dataset size and creating a corpus twice as large as the original.

3 Model Architecture

The model was implemented as an Encoder–Decoder Transformer using PyTorch's `nn.Transformer` module.

- Embedding Layer: Converts input tokens into fixed-dimension vectors
- Positional Encoding: Adds positional information using sine and cosine functions
- Transformer Layers: Multi-layer structure for both encoder and decoder
- Linear Projection Layer: Maps decoder outputs to the vocabulary size

Sequence masks and padding masks were applied to block attention on irrelevant positions.

4 Training Environment

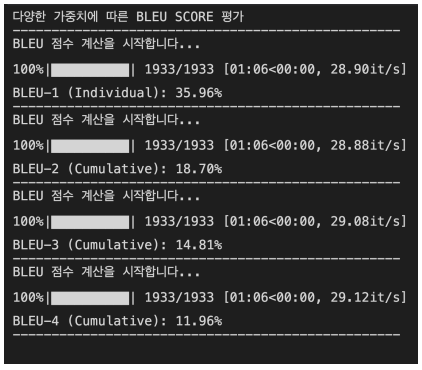
The loss function was CrossEntropy Loss, with padding tokens excluded from loss calculation. The optimizer was Adam with an initial learning rate of 2e-5. The batch size was 64 and training ran for 20 epochs.

Training was performed on an NVIDIA T4 GPU, and evaluation metrics included BLEU, Perplexity, and Accuracy.

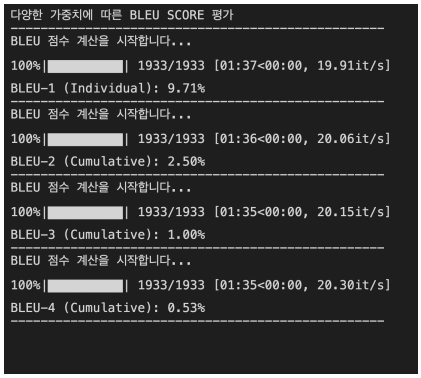
5 Results

Experiments were conducted with the same model structure, comparing training with the original dataset and the augmented dataset. All other conditions were kept identical to isolate the effect of augmentation.

Performance comparison centered on BLEU scores showed that the augmented model consistently performed worse than the original. Even when assigning higher weights to lower n-grams (given the short length of dialogues), the augmented model showed little improvement. This demonstrates that simple substitution-based augmentation does not help training and can even interfere.



(a) BLEU Score Before Augmentation



(b) BLEU Score After Augmentation

Figure 1: Comparison of BLEU scores before (a) and after (b) augmentation.

6 Limitation

By directly reviewing the augmented data, we found that while some sentences were meaningfully augmented, many contained contextually inappropriate words or simply frequent words inserted at random. Such faulty substitutions acted as noise, leading to performance degradation.

Although Lexical Substitution is simple and efficient, in agglutinative languages like Korean, a filtering process is needed to judge whether the substituted

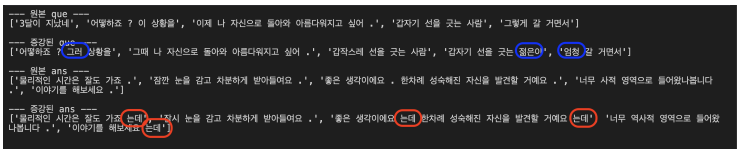


Figure 2: Examples of good bad augmentations

words fit the context. Future research should explore grammar-constrained or context-based augmentation methods.