– Independent Work Report Spring 2024 –

A Comparative Analysis of Multilingual and Monolingual Transformer Model Performance in Korean Grammar Error Correction

Tara Shukla
Adviser: Prof. Brian Kernighan

Abstract

This paper investigates the comparative performance of monolingual and multilingual Transformer-based natural language processing (NLP) models in the domain of grammar error correction (GEC). Since the introduction of Transformer-based architectures in 2017, they have been widely adopted for many state-of-the-art (SOTA) NLP models for various tasks. HoIver, given issues of data sparsity for loIr-resourced languages in NLP, a model's ability to capture linguistic features can vary depending on the amount and type of pretraining data used. In particular, it is relevant to discuss how the performance of Transformer models varies depending on if it is pretrained on one, versus several, source languages. This study employs a comprehensive evaluation framework to assess the performance of monolingual and multilingual models in Korean GEC. Focusing on multiple monolingual and multilingual Transformer models of varied structures, and a baseline model, I perform downstream fine-tuning using a task-specific monolingual dataset. When evaluated on the test set, I find that the fine-tuned multilingual models on average outperform their monolingual counterparts in areas of overall error correction and individual error classification. However, all Transformer models outperform the statistical-method based baseline. I comparatively analyze models' performance along several metrics, then discuss implications and future directions for research.

1. Introduction

Currently, most natural language processing (NLP) processes and research are based in seven languages: English, Chinese, Urdu, Farsi, Arabic, French, and Spanish [34]. Since the advent of

Transformers in 2017, NLP researchers have leveraged this architecture to achieve state of the art (SOTA) performance in various tasks; for example, in text generation, large language models (LLMs) have achieved near human level capabilities in English [6]. In addition, scholars have attempted to expand Transformers' capabilities beyond English, and use this model architecture to build NLP models to perform tasks in other languages. The approach to non-English NLP models varies: there exist monolingual Transformer models, which are pretrained on only one language. In contrast, a multilingual model is trained on data from multiple languages, and is able to handle them collectively through learned linguistic commonalities. Many prevalent SOTA models, such as BART or T5, were originally developed (monolingually) for English and later expanded to have multilingual capacity, or pretrained from scratch to be monolingual in non-English languages. In addition, despite their broad adoption in the field, the vast majority of multilingual Transformer-based models are pretrained on disproportionately English data [18].

Therefore, there exists an imbalance between many models' knowledge of English versus non-English languages. This inequality is compounded by the fact that the model architectures themselves were developed from the context of English NLP. As will be discussed later in this paper, many design choices need to consider linguistic differences when generalizing a model to perform well in non-English contexts. In addition, many multilingual models depend on cross-lingual transfer learning between languages, depending on higher-resourced languages to help models generalize for the linguistic patterns in less resourced languages; this generalization is not always correct, and can detrimentally impact performance in the lower-resourced language [26]. Thus, advancements in English-based models might deprioritize capturing the linguistic characteristics of languages highly morphologically deviated from English, and underperform for these languages. This is especially true for low-resourced languages, for which the scarcity of processed and labeled pretraining data means that multilingual models must pull from other language representations to capture linguistic features. Scholars call this the digital language divide: the idea that differences in languages' prevalence and speakers' global socioeconomic position can cause a "lack of digital infrastructure" for a language. This digital divide spans from touch-type keyboards to NLP knowledge

bases and models [34, 33]. When NLP models fail to equitably represent the world's languages, it means that LLMs can reflect societal power imbalances and become biased towards certain cultural values or knowledge bases [26]. In a world whose socioeconomic landscape is increasingly driven by machine-learning technologies, the digital language divide in AI has important implications for equitable scientific and economic development across cultures, and the preservation of global linguistic diversity.

In this work, I conduct a comparative evaluation of the performance of multilingual and monolingual models in the grammar error correction (GEC) task in Korean. I utilize existing Python packages and resources catered to NLP and machine learning, such as NLTK, TensorFlow, KoNLPy (a Korean NLP package), and more. Models are trained and tested on existing data, comprising of sentence pairs of Korean language-learners' original and corrected texts. Then, I compare models' performance across several task-specific metrics, and discuss implications for multilingual NLP. The findings contribute to the ongoing scholarly discourse on leveraging Transformer-based architectures for multilingual NLP applications, and offer insights into their adaptability and effectiveness across varied linguistic contexts.

2. Problem Background and Related Work

2.1. Linguistic Differences

Notably, syntactic and morphological variations in languages' structures and characteristics can change the way a language is processed, and consequently how NLP models can generalize across languages.

In linguistics, morphology is the study of word formation, including processes like prefixes, suffixes, conjugation, and more. A languages' syntax refers to its rules of sentence formation and structure. Notably, English is considered a morpho-syntactically simple language (for instance, many words are conjugated with '-s' or '-es', no matter the subject), whereas languages like Tagalog, Czech, Korean, are considered more complex due to features like free word order, usage of inflection, character building, and more [43].

I take Korean as an example. In Korean, grammar is highly context dependent; for instance, honorific markers earlier in a sentence could correlate to a grammatical need for honorific verb conjugation later on. The language is also highly polysemic (i.e. many words have several meanings), is a radical pro-drop language (permits the contextual exclusion of subjects, objects, and possessors), and is agglutinative (morpheme-conjoining) [33, 27]. Thus, it is evident that linguistic variations might impact optimal processing techniques.

Scholars have found this as well; in [12], Gerz et.al conclude that a high level of morphosyntactic complexity is a predictor for worse language modeling. In [29], Park, et.al find that a language's morphology is highly correlated with surprisal in language modeling. Therefore, we observe that LLM performance in non-English morphologically complex languages is likely to compare negatively with English performance in current SOTA methods. In the next section, I discuss why this might be, in the context of Transformer models.

2.2. Transformers and Tokenization

The Transformer model architecture, created in 2017 by Vaswani et al. in "Attention is All You Need," [38], has become the foundation for many SOTA NLP models across different languages and tasks. They are a class of NLP machine learning models that rely on self-attention mechanisms to process sequences of tokens. Transformers take in tokened inputs and create input embeddings and positional encodings to capture the semantic and positional features of the input. They consist of multiple layers, each with two main subcomponents: multi-head self-attention and position-wise feedforward networks. There is an encoder and/or a decoder, which process the input and output sequences, respectively [38, 23]. This work will be comparing the performance of different models built using the Transformer architecture, whose general structure is pictured in Figure 1.

Tokenization refers to the numeric representations of text in NLP; this is the input sequence that is fed into a Transformer model. As mentioned previously, many non-English languages have highly different sturctural and morphological conditions; these can contribute to differences in how a language is best tokenized. First, pre-tokenization, some languages can benefit from

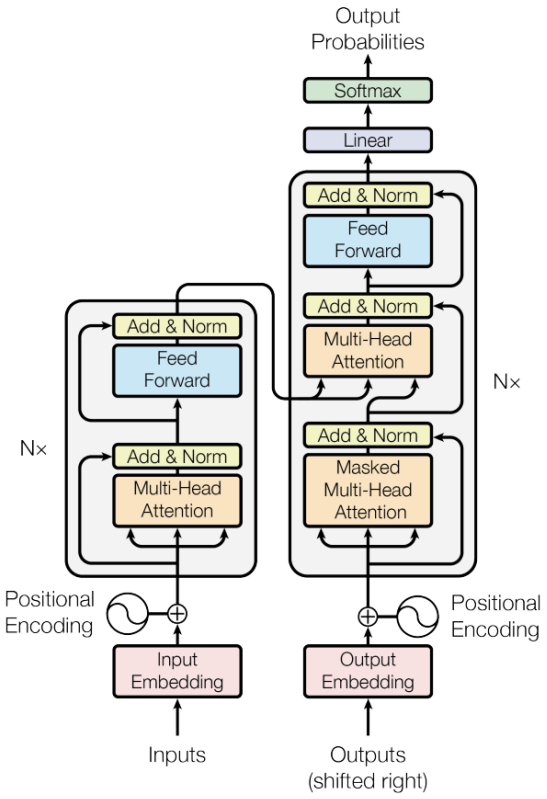

Figure 1: Tranformer model architecture: from Vaswani, et.al [38]

normalization strategies, such as unicode normalization: this is particularly true for languages with character building, like Korean, Chinese, or Japanese. Optimizing normalization strategies in a multilingual setting is an ongoing topic of study: Burlot and Yvon discuss appropriate normalization in a setting of morphological variation [5]. Thus, pre-tokenization language processing needs can vary depending on morphology.

Also, the effectiveness of tokenization strategies themselves can vary based on language. Some tokenizers split a sentence into words (word-based tokenization), whereas others reduce words to characters— in an English context, “boys” would be reduced to “b”, “o”, “y”, “s”. A middle ground encapsulating the increased complexity of word-based tokenization without dramatic parameter sizes, is subword based tokenization. This is the tokenizer type used by many SOTA Transformer models; it breaks words into smaller subwords. For example, “boys” would be reduced to “boy”, “s”, in sub-word-based tokenization [1].

However, in [8], Chung and Gildea note that for morphologically rich languages, the ambiguity of morpheme boundaries means that current tokenization strategies are sub-optimal for morphological segmentation, and thus worse at capturing linguistic nuance.

Jeon, et.al echo this finding in [16], in which they note that Korean's syllabic writing system means that " syllable character-level subword tokenization would not fully capture Korean's agglutinative grammar." They note that sub-character decomposition and morphological analysis are possible tokenization approaches that could fight this suboptimality. In [37], Toraman, et.al propose 'Morphological-level' tokenization, which feeds words through a morphological analysis tool before tokenization, and performs competitively with the SOTA subword tokenizers for the low-resourced Turkish language. Finally, in [15], Jabbar proposes MorphPiece, which uses morphological segmentation to competitively perform against statistical-based tokenizers in a series of NLP tasks. In the analysis of tokenization strategies in Korean NLP, Park, et.al find that a hybrid approach of morphological segmentation and subword Byte-Pair Encoding (BPE) tokenization is optimal for Korean [30]. Overall, current SOTA statistical methods of tokenization can be challenged by various tokenization strategies for morphologically rich languages. This means that current Transformer NLP models, which on the whole use statistical tokenizers — like WordPiece, SentencePiece, or Byte-Pair-Encoding (BPE)— could have decreased performance in morphologically rich languages due to this design choice. This furthers my point on the English-centricity of current SOTA models.

2.3. Grammar Error Correction

Now, I will introduce the task and scholarly approaches to grammar error correction. GEC is the task of identifying and rectifying grammar errors in written text: this can include morphological (word formation) errors, or syntactic (sentence formation) errors.

Scholars in machine learning NLP have developed multiple methods of dealing with the GEC task. In [4], Bryant, et.al discuss statistical and neural machine translation, recurrent neural networks, and convolutional neural networks as historical frameworks for GEC. In the context of machine learning, GEC is a sequence-to-sequence task: it involves conversion of inputs from one domain to another. It

is therefore considered a parallel task to machine translation [35]. Currently, Transformers dominate SOTA GEC approaches [4].

However, the GEC task can suffer from ambiguity in the definition of the ground truth. Bryant, et.al note that corrections are subjective and varied; as an English example, 'I like to reading" might be corrected to "I like to read" and "I like reading", in an equally correct manner [4]. This can make it easy to underestimate the performance of systems. Languages with more morphological complexity or varied syntax— like free word order— can be harder to correct via ML GEC systems. For example, in [31], Rozovskaya and Roth note that for morphologically rich languages, subject-verb agreement errors are difficult to catch because there are multiple correct placements for the subject; in pro-drop languages like Russian and Korean, the subject can also be excluded, further complicating performance evaluation.

In the context of language learning, GEC becomes more complicated; Bryant, et.al note that types and prevalence of spelling and syntactical errors differ among native and non-native writers, meaning that that inconsistent error patterns could make it more difficult for LLMs to generalize the GEC task between use cases [4]. Overall, the GEC task is a challenge to perform and evaluate, especially for morphologically rich languages like Korean.

2.4. Related Work

This section is adapted from my research proposal, [33].

NLP researchers have long been interested in exploring how multilingual models can compare to their monolingual counterparts, especially in the cases of data scarcity and morphosyntactically complex non-English languages.

As previously mentioned, researchers have attempted to adapt existing English SOTA models to be monolingual in other languages, through pre-training from scratch with a monolingual corpus. In [42], Yoon et.al source and develop multiple corpuses for Korean grammar error correction, one of which (Kor-Leaner), this work will employ. They use it to pretrain koBART (a Korean monolingual BART-based model) on the GEC task, but do not perform a comparative analysis with other models.

In [41], Wongso et.al find that in the case of Sudanese, monolingual BERT and RoBERTa models outperformed mBERT (multilingual BERT) in downstream classification tasks; similarly, in [39], Virtanen et. al note that "BERT is not enough", and implement a Finnish NLP model from scratch, with only Finnish training data. Their model 'systematically outperforms' the multilingual BERT on benchmark tasks such as sentiment analysis and text classification. Finally, in [20], Lee et.al create a small-scale monolingual Korean model, and note its higher performance on downstream tasks when compared to multilingual BERT. However, since these examples focus on pretraining or fine-tuning monolingual models, neither are able to compare multiple Transformer architecture types in a multilingual and monolingual context. In addition, none focus on sequence to sequence tasks, like GEC.

Other scholars have performed comparative analyses on monolingual versus multilingual models. In [17] Jørgensen, et. al note that translated corpuses— which are sometimes used in multilingual pretraining— sometimes do not retain high quality, and that multilingual models can be unpredictable in their handling of different languages; they can rely on 'pervasive, yet spurious' syntactical correlations amongst languages. In [28], Nicholas and Bhatia note that in some applications, multilingual models have been shown to outperform monolingual models, even in the latter's training language. However, they also discuss shortcomings of multilingual models, which can rely upon faulty machine translations, do worse with contextual semantic analysis, and are faulty the more languages they are trained on. They recommend caution in the use of multilingual LLMs, noting that they contribute to a "post-colonial structural inequality" that allows over-resourced dominant languages to supersede technological advancements which use under-resourced languages. Therefore, we see that there are conflicting scholarly reports on the comparative efficacy of pretraining in a monolingual versus multilingual context.

Finally, the performance differences between multi and monolingual NLP models is difficult to examine through analysis of commercial products. Western researchers have used NLP techniques frequently for grammar checking and classification, primarily in English, but also for non-English languages. In the United States, prevalent tools like Grammarly and Google Docs perform grammar,

spelling, and punctuation checking, among other linguistic and writing tools. For Korean specifically, many different AI-driven grammar checkers are sold as APIs, like Sapling [1] and Arvin AI.[2] Notably, these products' designs and training data are hidden to the public, so the cross, multi, or monolingual nature of their proprietary NLP models is unknowable in a research context.

Therefore, existing literature and available data is conflicted as to whether monolingual or multilingual models will do better in sequence-to-sequence tasks for a non-English, morphologically complex language. Thus, this paper aims to add to existing research on mono vs multilingual NLP model comparative analysis. Specifically, through focusing on the differences between the models' performance in grammar error correction, I will add to scholarly discourse about the scope and limitations of these models in different linguistic contexts.

3. Approach

My approach will fine-tune various pretrained monolingual Korean, and pretrained multilingual Transformer models for the grammar error detection and correction task in Korean. I will utilize a dataset of error-filled and corrected sentence pairings to train, validate, and test these models using a range of metrics, including M2-scoring precision, recall, and F0.5 scores, as well as BLEU and GLEU scores, as I will discuss in later sections. By narrowing the scope of the models' tasks to GEC, I hope to gain valuable insights on the performance of monolingual non-English versus multilingual models in this context. Where other papers have large scopes (for instance, comparing how models can capture deep syntactic features of a given language), or might themselves implement a monolingual model for comparison, I will be working with pre-trained models to focus on the comparative analysis task itself. As discussed above, grammar error type prevalencies differ between native and non-native speakers, so I will additionally define the scope to GEC for a nonnative Korean corpus. This will hopefully have implications for our understanding of the comparative performance of monolingual and multilingual models for morphologically complex non-English languages [33].

[1] https://sapling.ai/lang/korean
[2] https://arvin.chat/ai-tools/korean-grammar-checker

4. Implementation

4.1. Dataset

The data used was the National Institute of Korean Language (NIKL) learner corpus. The preprocessed version of the data was taken from the open-sourced dataset that Yoon, et.al used in [42]. The NIKL corpus is a set of 28,426 pairs of sentences. These sentences were taken from essays written by Korean language learners; they were then annotated by professional Korean tutors, and the edits were categorized into different error types [42]. This corpus is the largest corpus from Korean language learners collected to date, with 3.7 million words and extensive annotations regarding error type and token [7]. Notably, Yoon, et.al s open-source version of this corpus is cleaned and refined by Yoon, et.al, including the removal of duplicates and filtering out of empty, inconsistent, or untagged edits. As discussed above, I used the learner corpus in order to narrow the scope of the GEC task to errors commonly encountered by language learners. I applied preprocessing techniques to the dataset before using it in the fine-tuning process. This included sentence splitting between error and corrected sentences, and splitting the corpus into training, validation, and test sets, with an 80-10-10% split for training, validation, and testing.

4.2. Classification and Classification Scoring Programs

An important aspect of measuring model performance in GEC is analyzing a model's ability to correct errors of different types. In English, error types can range from subject-verb agreement errors, to article errors, and more. For English GEC error classification, the industry standard is called the Error Annotation Toolkit (ERRANT). Introduced by Byrant, et al. in [3], ERRANT is a tool that can identify and classify grammar error corrections, given an input of parallel error-filled and corrected sentences. In this work, I used KAGAS (Korean Automatic Grammatical error Annotation System), an open-source Korean ERRANT-like system proposed and developed by Yoon, et. al in [42]. In their paper, they identified 14 error types, which cover the part-of-speech categories in Korean, and include spacing and other syntactic and morphological errors common to

the language. The error types, corresponding abbreviation, and the presence of each error type in the learner corpus, are shown in Table 1, below.

Error Type	Abbreviation	Count	Percentage
INSERTION	INS	3352	5.64%
DELETION	DEL	1652	2.78%
SPELLING	SPL	6735	11.33%
PUNCTUATION	PUNCT	0	0.00%
SHORTEN	SHORT	363	0.61%
WORD STEM	WS	108	0.18%
WORD ORDER	WO	0	0.00%
NOUN	NOUN	4879	8.21%
VERB	VERB	2456	4.13%
ADJECTIVE	ADJ	411	0.69%
CONJUGATION	CONJ	4917	8.28%
PARTICLE	PART	16700	28.11%
ENDING	END	7560	12.72%
MODIFIER	MOD	1035	1.74%
UNKNOWN	UNK	9251	15.57%
TOTAL	-	59419	100.00%

Table 1: Error Type Distribution, adapted from Yoon, et.al [42]

In order to evaluate correctly and incorrectly calculated errors by type, KAGAS, like ERRANT, outputs files in an M2 format. The M2 format is a standardized format used for storing annotated corrections. Each annotation will include the index (or range of indices) indicating error position; the error type; and the corrected annotation. A sample of the M2 file annotation is seen in Figure 2, below.

Figure 2: KAGAS output: sample M2 file

```
S 나무집을 짓을 때 나무를 오랬동안 살았면 짓고 좋겠습니다.
A 3 4|||PARTICLE|||나무가|||REQUIRED|||-NONE-|||0
A 4 5|||UNCLASSIFIED|||오래|||REQUIRED|||-NONE-|||0
A 5 6|||CONJUGATION|||산|||REQUIRED|||-NONE-|||0
A 6 6|||INSERTION|||것으로|||REQUIRED|||-NONE-|||0
A 6 7|||ENDING|||짓으면|||REQUIRED|||-NONE-|||0
```

(a) *Translation: When I build a treehouse, I hope the tree will live for a long time. KAGAS has identified particle, unclassified, conjugation, insertion, and verb ending errors in this annotation.*

In order to evaluate the similarity of a model's corrections to the human annotations, I utilized

KAGAS to create M2 files, and ran them through an open-source M2 scorer.[3] The M2 scorer compares two M2 files— the "gold annotation" and the evaluating annotation— and outputs precision, recall, and F0.5 scores. In addition, I developed an M2 file parser Python script to extract the precision, recall, and F0.5 scores for each error type. To do so, I parsed the index and error type from each annotation in each sentence in the file, compared them with the gold standard annotation, and aggregated true and false positive and negative counts across all sentences and error types.

4.3. Models

In this paper, I focus on a baseline model and seven open-source models, all using the Transformer architecture. Their details and differences are discussed below.

4.3.1. Baseline: HANSPELL As a baseline model, I used HANSPELL, the same baseline that Yoon, et.al use in [42]. HANSPELL is a statistical GEC model from the Kakao API [19]. To run the baseline model, I downloaded the open-source hanspell client and ran it locally; I wrote a Python script to feed it the test set sentences and write the output to a file. I then ran the file through the same M2 classification program and my M2 file parser mentioned in the previous section. Finally, I used the koNLPY and NLTK packages to calculate BLEU and GLEU scores. This allowed me to compare the statistical model's performance with the Transformers in the context of M2, BLEU, and GLEU metrics, as I will discuss in a later section.

4.3.2. BART The BART (Bidirectional and Auto-Regressive Transformer) model was developed by Lewis, et.al in 2019, for the English language; in 2020, Liu, et.al created mBART, multilingual BART [21, 22]. BART's pretraining objective is primarily denoising autoencoding: corrupting input text with a noising function, and training the model to recreate the original text.

KoBART is a BART-architecture model trained on monolingual Korean data; this paper uses two monolingual BART-based models, KoBART and KoBART2, which are open-sourced models

[3] https://github.com/nusnlp/m2scorer

available on HuggingFace under the model cards 'hyunwoong/kobart'[4] and 'gogamza/kobart-summarization'[5], which are KoBART models further pretrained with casual conversation and news data, respectively. I also used "facebook/mbart-large-50"[6] to fine-tune mBART.

4.3.3. BERT BERT (Bidirectional Encoder Representations from Transformers) is an encoder-only Transformer model which is pretrained with masked language modeling (MLM) and next-sentence-prediction (NSP) [29]. MLM is a pretraining objective that involves masking a percentage of the input tokens, and training the model to predict the masked tokens based on the surrounding ones; it helps models capture bidirectional context and learn more robust representations of words and phrases. Meanwhile, NSP allows BERT to learn sentence-level interactions [14].

BERT is often fine-tuned on downstream tasks like GEC, needing task-specific layers on top of the pretrained encoder. I could not find an open-sourced multilingual BERT-based architecture with a generation-compatible head that was compatible with Korean, so my inclusion of BERT models is limited to koBERT. KoBERT is a BERT-based model pretrained with 70GB of Korean text from the National Institute of Korean Language. The Hugggingface open-source model card is found at "monologg/kobert".[7]

4.3.4. M2M M2M (Many-to-Many) is an encoder-decoder Transformer model introduced by Fan, et.al in [9]. It was pretrained on a vast datamined multilingual corpus, enabling it to handle translation between many different language pairs. It uses similar objectives to BERT and BART during pretraining, but scales up the Transformer model dimensions [9]. I used the open-source Facebook M2M implementation, under the model card "facebook/m2m100_418M". [8]

[4]https://huggingface.co/hyunwoongko/kobart
[5]https://huggingface.co/gogamza/kobart-summarization
[6]https://huggingface.co/facebook/mbart-large-50
[7]https://huggingface.co/monologg/kobert
[8]https://huggingface.co/facebook/m2m100_418M

4.3.5. T5 T5 (Text to Text Transfer Transformer) is an encoder-decoder Transformer architecture which is primarily pretrained using a fill-in-the-blank objective— this is similar to MLM, but is more generalizable because it allows the model to fill in masks with multiple tokens [11]. In addition, the model is text-to-text, meaning that its inputs and outputs are in text format. From HuggingFace, I used the monolingual Korean T5 model, "KETI-AIR/ke-t5-base", [9] and the multilingual "google/mt5-base"[10] as models.

4.4. Fine-tuning

The fine-tuning process for every model involved initializing the models with their pre-trained weights, then training it with the NIKL learner corpus using a sequence-to-sequence training objective. I fine-tuned the pre-trained transformer models using PyTorch and the Hugging Face Transformers library, including the Trainer API and their open-source NLP model library. I based the fine-tuning code structure on the "Translation" section of the HuggingFace NLP course.[11] Other Python libraries included sklearn, NLTK and koNLPY, NLP toolkit libraries. All model training was done in Google Colab with the L4 GPU.

During fine-tuning, I utilized a sequence-to-sequence framework, where the input sequence contained the original text with grammatical errors, and the target sequence contained the corrected text. This is in line with the approach to GEC as a machine translation task. For fine tuning, I used between 3 and 10 epochs for each model. Due to differences in model architecture and pre-training dataset size — especially between mono and multilingual versions of the same model architectures— some models needed more training epochs to learn the downstream task. I used the AdamW (Adam with weight decay) module in the HuggingFace optimization module, with a weight decay of 0.01, trying to minimize cross-entropy loss. In sequence-to-sequence tasks, cross-entropy loss is a common loss formula, in which the model will try to minimize the distance between the target value and its predicted output. The formula for cross-entropy loss is as follows:

[9] https://huggingface.co/KETI-AIR/ke-t5-base
[10] https://huggingface.co/google/mt5-base
[11] https://huggingface.co/learn/nlp-course/chapter7/4?fw=pt

$$\text{Cross-Entropy Loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\log(p(y_{i,t}))$$

With N representing the training corpus size, T the output sequence length, $y_{i,t}$ the ground truth of each token, and $p(y_{i,t})$ as the probability of the model outputting that ground truth.

For learning rate, I chose between 2e-5 and 3e-5, and for batch size I chose between 8 and 64. These parameters were chosen via monitoring the training loss and validation loss, and adjusting for over and underfitting, as well as adjusting based on computational resources.

4.5. Metrics

As mentioned above, the proper evaluation of GEC systems is an ongoing discussion in the field; to evaluate the performance of the transformer models on Korean GEC tasks, I employed a range of evaluation metrics, including a MaxMatch (M2) score metric that calculated precision, recall, and $F_{0.5}$-score at sentence levels.

Precision is calculated with the formula, $\frac{\text{True Positive}}{\text{(True Positive + False Positive)}}$.

Recall is calculated with the formula, $\frac{\text{True Positive}}{\text{(True Positive + False Negative)}}$.

Meanwhile, the $F_{0.5}$ score is the weighted harmonic mean of precision and recall, giving more weight to precision. M2 scoring is a common NLP metric for the GEC task, since it measures a model's ability to identify errors.

However, the M2 scoring method has faced criticism for treating no-correction and wrong-correction outputs the same, as pointed out by Wang, et.al in [25]. Therefore, I also measured BLEU and GLEU (Google BLEU) scores to provide insights into the models' effectiveness in correcting grammatical errors while preserving the semantic integrity of the text.

The BLEU score is a metric for measuring text similarity; ranging from 0-1, it reports a measure of the n-gram match counts between the candidate and reference text. The GLEU score, also ranging

from 0-1, is a metric developed by Google to measure text similarity. In contrast to the BLEU score, GLEU measures fluency and grammatical correctness better because it focuses on 1 and 2-gram precision score calculations [40, 10]. Thus, my main methods of model performance evaluation were M2, BLEU, and GLEU scores.

In particular, several models used BEAM search— a heuristic search algorithm outputting top k probabilistic sequences. Therefore, they outputted several versions of the corrected sentence, so in post-processing I picked the one with the highest GLEU score relative to the human-corrected corresponding gold annotation.

5. Evaluation

First, we examine the BLEU and GLEU score comparisons during training for multilingual and monolingual models. The metric plots are shown in Figures 3 and 4 below.

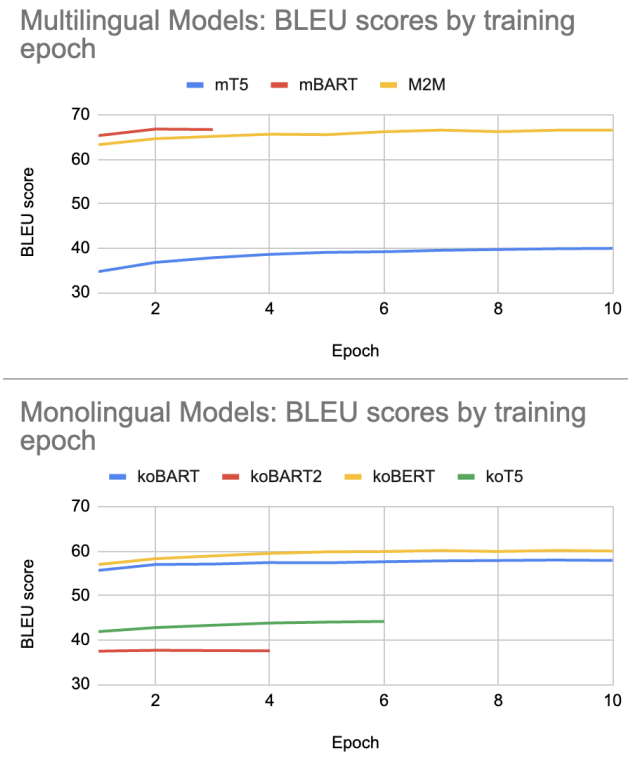

Figure 3: BLEU model training performance

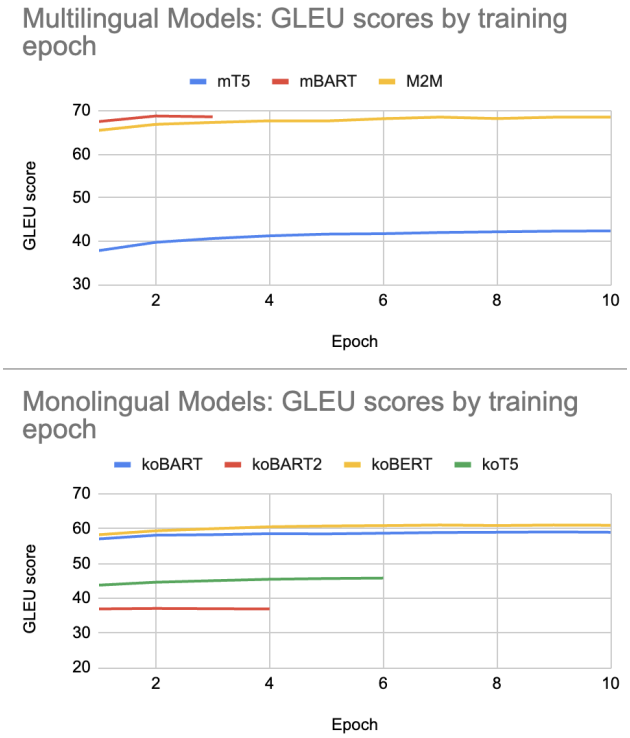

Figure 4: GLEU model training performance

On average, the monolingual models were able to improve their GLEU scores by 2.23 points during fine tuning, and their BLEU scores by 1.91 points. In contrast, the multilingual models were able to improve their GLEU scores by an average of 2.81 points, and BLEU scores by 3.29 points. This represents a 26% and 72% increase in model trainability in multilingual models through both metrics, respectively.

Now, we turn to models' performance on the test set in M2, BLEU, and GLEU metrics. The chart of model performance is shown in Table 2, below. For readability, the best performer in each category is bolded.

We find that among multilingual models, mBART outperforms its peers by most metrics, whereas among monolingual models, koT5 had the best performance. Overall, multilingual models outperform monolingual models in recall, BLEU, and GLEU scores, whereas koT5 dominates in precision and F0.5. In comparison to the baseline HANSPELL model, we find that the monolingual models outperformed the baseline via an average of 0.22, 0.31, and 0.34 in precision, recall, and F0.5 respectively. In addition, they outperformed the baseline BLEU and GLEU scores by 0.18 and 0.07,

Table 2: Test Set Performance Evaluation

Model	M2			Bleu	Gleu
	Precision	Recall	F_0.5		
Baseline	0.3118	0.0477	0.1480	0.312	0.435
KoBART	0.4879	0.3766	0.4607	0.56951	0.5821
KoBART2	0.5227	0.2795	0.4452	0.3790	0.3730
KoBERT	0.5246	0.3652	0.4825	0.59201	0.6025
KoT5	**0.6255**	0.4038	**0.5636**	0.43370	0.45066
MT5	0.6098	0.2001	0.4326	0.39306	0.4177
MBART	0.5825	**0.4370**	0.5461	**0.66261**	0.6823
M2M	0.5021	0.3784	0.4713	0.65711	**0.6781**

respectively. In contrast, the multilingual models outperformed the baseline by an average of 0.25, 0.29, and 0.35, in precision, recall, and F0.5, respectively. They outperformed the baseline BLEU and GLEU by 0.26 and 0.16 points on average.

Finally, the model precision, recall, and F0.5 scores by error type are seen in Table 3, below. For readability, the highest score in each evaluation metric, for each error type, is bolded.

Table 3: Error Analysis

Error Type	Model	INS	WS	WO	SPELL	PUNCT	SHORT	VERB	ADJ	NOUN	PART	END	MOD	CONJ	UNK
Precision	HANSPELL	0	0.371	0	33.483	0	7.143	2.821	6.452	3.935	0.259	2.157	2.581	6.676	0
	KoBART	3.95	18.18	0	54.42	0	**34.48**	**27.34**	16.25	28.46	54.93	34.56	31.02	32.4	39.6
	KoBART2	5.9	11.11	0	35.27	0	46	19.02	10.2	16.01	50.68	26.79	9.85	24.8	31.95
	KoBERT	12.09	5.88	0	48.25	0	24.02	23.35	25	24.5	55.52	32.02	32.61	29.54	36.05
	KoT5	**15.58**	21.43	0	55.6	0	29.33	25.16	**24.68**	32.99	60.91	39.41	28.98	33.14	43.55
	mT5	2.84	12.9	0	28.9	0	33.73	6.75	18.46	8.32	42.5	18.39	9.94	16.93	26.69
	mBART	13.79	**23.53**	0	**58.36**	0	34.29	26.63	20.73	**34.03**	**61.79**	**40.12**	**35.59**	**34.92**	**45.94**
	m2m	9.4	11.5	0	50.9	0	25.6	21.3	17.6	26	54.6	32.8	30.6	28.7	39.1
Recall	HANSPELL	0	0.371	0	32.341	0	7.143	2.453	6.452	3.697	0.199	1.961	2.581	6.467	0
	KoBART	4.78	18.18	0	52.23	0	**34.48**	**26.81**	16.25	27.47	52.77	33.98	29.59	32.33	38.64
	KoBART2	11.2	11.11	0	32.87	0	46	18.42	10.2	15.27	46.62	25.68	9.85	23.86	30.2
	KoBERT	13.23	5.15	0	46.5	0	24.51	22.94	25	23.82	54.1	31.32	31.56	29.56	35.63
	KoT5	15.47	19.64	0	52.89	0	29.33	24.32	**24.68**	31.9	59.47	38.35	28.13	32.81	42.34
	mT5	2.61	11.29	0	27.47	0	33.73	6.75	18.46	7.56	38.64	17.35	9.32	16.34	25.08
	mBART	**16.94**	**22.06**	0	**55.81**	0	34.29	26.73	20.12	**32.86**	**60.62**	**38.81**	**34.93**	**34.53**	**45.47**
	m2m	12.7	10.6	0	48.7	0	25.6	20.9	17.6	25.6	53.8	32.1	30.3	28.9	38.7
F0.5	HANSPELL	0	0.371	0	32.934	0	7.143	2.695	6.452	3.839	0.236	2.092	2.581	6.606	0
	KoBART	3.97	18.18	0	53.61	0	**34.48**	27.09	16.25	28	53.71	34.11	30.51	32.1	38.93
	KoBART2	5.78	11.11	0	34.45	0	46	18.82	10.2	15.69	49.07	26.35	9.85	24.4	31.23
	KoBERT	12.01	5.64	0	47.55	0	24.07	23.09	25	24.14	54.45	31.63	32.18	29.3	35.46
	KoT5	**15.42**	20.83	0	54.6	0	29.33	24.88	**24.68**	32.38	59.94	38.85	28.69	32.83	42.85
	mT5	2.76	12.37	0	28.4	0	33.73	6.75	18.46	8.05	41.01	18.02	9.73	16.72	26.09
	mBART	13.7	**23.04**	0	**57.41**	0	34.29	**26.57**	20.53	**33.48**	**60.85**	**39.52**	**35.21**	**34.61**	**45.37**
	m2m	9.3	11.2	0	50	0	25.6	21.1	17.6	25.6	53.8	32.4	30.4	28.5	38.6

From the figure, we see that mBART has the most wins in the correction of all error types. Interestingly, no other multilingual model is the best scorer for any other error type; koT5 and

koBART are the top performers for adjective and shorten error types overall, and most verb and insertion metrics.

Overall, I find that multilingual models outperform monolingual models in Korean grammar error correction. This speaks to the advantages of multilingual models' ability to leverage linguistic diversity and better adapt to varied linguistic tasks and domains. However, the evaluation of the monolingual models did show comparable results, and they even outperformed their multilingual counterparts in certain error types. This might speak to the idea that monolingual models can better capture the nuances of complex grammar, syntax, and semantics for a single language.

5.1. Punctuation and Word Order

Initially, I observed the 0.0 precision, recall, and F0.5 in the PUNCT and WO categories to be surprising considering the models' relatively good performance among other error types. However, upon closer examination of the data, I found that there were only 3 datapoints (out of 4265 sentences) that contained punctuation errors in the test set, and 19 in the total dataset (other error types ranged between 2-10k error points). In addition, there were only 4 data points that contained word order errors in the total dataset (which was of size 28,426 pairs). Therefore, the 0.0 evaluation results are expected. I have kept the WO and PUNCT categories in the results table, as Yoon, et.al had identified it as a possible grammatical error to make in written Korean [42]. In the datasets they provided, these error types have a statistically significant presence in the datasets gleaned from native speakers (in comparison to a negligible presence in our dataset taken from language-learner generated text). More research in this area is needed to discern the reason for non-native and native grammar error differences in Korean.

6. Discussion

In this work, I have found that in the task of Korean GEC, multilingual Transformer models on average outperform their monolingual counterparts. I successfully conducted a comparative analysis across multiple Transformer architecture types, and found that the monolingual koT5 model is competitive with mBART, but mBART is still the winner over all model types and architectures.

Notably, this work is limited by its scope: I have explored the task of Korean GEC with a corpus drawn from language learners. More work must be done to discover if my results hold for native corpuses, in other languages, or for different sequence-to-sequence tasks. Overall, it is significant that a multilingual approach to Korean GEC seems to perform better than a Korean-specific model. It is a promising conclusion towards Transformers' ability to capture semantic representations of languages which are underrepresented in the pretraining data. This has implications for the uses of AI in fighting the digital infrastructure divide, and in maintaining global linguistic diversity.

Analyzing this paper's conclusions from a broader perspective, we find that it sheds light on linguistic imbalances in the field currently. Since their advent, scholars have viewed the Transformer model as a highly generalizable solution for NLP tasks across multiple languages [2]. Thus, multilingual Transformers are an effective approach to NLP tasks in low-resource languages. In addition, some scholars argue that the semantic richness of morphologically complex languages is inherently more difficult to fully capture via large language models [13]. Thus, in future work, more research is needed on how morphological and syntactic variations can be represented in a vector space.

However, it's important to note that while the structure of Transformers is conducive to multilingual applications, we have noted in previous sections that very low-resource or morphologically complex monolingual models have the capacity to outperform mBART and mBERT. In addition, as previously mentioned, the subword tokenization approach is suboptimal for morphologically rich languages, and monolingual performance can be improved using morphologically-aware tokenization strategies. Thus, aspects of the Transformer architecture have the potential to be modified to better suit different languages. Some research has been done into this already, with researchers such as Soulos, [36], using induced structural biases to make Transformers more grammatically robust in different linguistic contexts. Future research in this area could explore if monolingual Transformers with different tokenization strategies and structural biases could outperform multilingual models.

In [24], McKeever comments that the immense divide between English and non-English NLP pretraining data and NLP research and development, contributes to an English-centric linguistic

hegemony that reinforces global socioeconomic power imbalances. It could be beneficial for future NLP research to focus on creating and refining Transformer models from the perspective of non-English linguistic study. In [32], Ruder remarks that in the process of pushing English-language NLP models to match human-level performance, "We have overfit to the characteristics and conditions of English-language data." If the field is to counter English dominance in NLP, we must continue research and development of language models from an originally non-English perspective, and continue investment into large-scale data collection and annotation from non-English sources. In the words of Vaswani, et.al— "Attention is all you need." Perhaps future NLP research should shift its attention towards fighting linguistic hegemony.

7. Acknowledgements

I would like to thank my IW adviser, Prof. Brian Kernighan, and the seminar assistants Dr.Wouter Haverals and Mary Naydan for their teaching and support during this project. I would also like to thank the other students in the Digital Humanities IW seminar for their advice and encouragement. I'd also like to thank my family and my friends for their support.

8. Honor Code

This paper represents my own work in accordance with University regulations. - Tara Shukla

References

[1] "Decoding text: A deep dive into tokenization strategies in natural language processing," Jan. 2024. [Online]. Available: https://baotramduong.medium.com/natural-language-processing-tokenization-30d6687ffa4b

[2] S. Bhattacharya and O. Bojar, "Unveiling multilinguality in transformer models: Exploring language specificity in feed-forward networks," 2023. [Online]. Available: https://aclanthology.org/2023.blackboxnlp-1.9.pdf

[3] C. Bryant, M. Felice, and T. Briscoe, "Automatic annotation and evaluation of error types for grammatical error correction," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2017.

[4] C. Bryant, Y. Zheng, M. R. Qorib, H. Cao, H. T. Ng, and T. Briscoe, "Grammatical error correction: A survey of the state of the art," *Computational Linguistics*, pp. 1–59, Jun. 2023.

[5] F. Burlot and F. Yvon, "Learning morphological normalization for translation from and into morphologically rich languages," *The Prague Bulletin of Mathematical Linguistics*, vol. 108, 06 2017.

[6] J. E. Casal and M. Kessler, "Can linguists distinguish between chatgpt/ai and human writing?: A study of research ethics and academic publishing," *Research Methods in Applied Linguistics*, vol. 2, no. 3, p. 100068, Dec. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S2772766123000289

[7] J. Chun and M. H. Kim, "Corpus-informed application based on korean learners' corpus: substitution errors of topic and nominative markers," *Asian-Pacific Journal of Second and Foreign Language Education*, vol. 6, no. 1, Aug. 2021.

[8] T. Chung and D. Gildea, "Unsupervised tokenization for machine translation," *CiteSeer X (The Pennsylvania State University)*, Jan. 2009.

[9] A. Dongre, "Pre-training llms: Techniques and objectives," *Medium*, Jul. 2023. [Online]. Available: https://dongreanay.medium.com/pre-training-llms-techniques-and-objectives-a75a1bf274b2

[10] K. Doshi, "Foundations of nlp explained — bleu score and wer metrics," *Medium*, May 2021. [Online]. Available: https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b

[11] A. Fan and et al., "Beyond english-centric multilingual machine translation," *arXiv.org*, Oct. 2020. [Online]. Available: https://arxiv.org/abs/2010.11125

[12] D. Gerz, I. Vulić, E. Ponti, J. Naradowsky, R. Reichart, and A. Korhonen, "Language modeling for morphologically rich languages: Character-aware modeling for word-level prediction," *Transactions of the Association for Computational Linguistics*, vol. 6, pp. 451–465, Dec. 2018.

[13] D. Gerz, I. Vulić, E. M. Ponti, R. Reichart, and A. Korhonen, "On the relation between linguistic typology and (limitations of) multilingual language modeling," *Edinburgh Research Explorer (University of Edinburgh)*, Jan. 2018.

[14] A. Issa, "Transformer, gpt-3,gpt-j, t5 and bert." *Medium*, Jun. 2023. [Online]. Available: https://aliissa99.medium.com/transformer-gpt-3-gpt-j-t5-and-bert-4cf8915dd86f

[15] H. Jabbar, "Morphpiece : Moving away from statistical language representation," *arXiv (Cornell University)*, Jan. 2023.

[16] T. Jeon, B. Yang, C. Kim, and Y. Lim, "Improving korean nlp tasks with linguistically informed subword tokenization and sub-character decomposition," 2023.

[17] R. Jørgensen, F. Caccavale, C. Igel, and A. Søgaard, "Are multilingual sentiment models equally right for the right reasons?" Jan. 2022.

[18] K. S. Kalyan, A. Rajasekharan, and S. Sangeetha, "Ammu: A survey of transformer-based biomedical pretrained language models," *Journal of Biomedical Informatics*, vol. 126, p. 103982, 2022.

[19] M. Lee, H. Shin, D. Lee, and S.-P. Choi, "Korean grammatical error correction based on transformer with copying mechanisms and grammatical noise implantation methods," *Sensors*, vol. 21, no. 8, p. 2658, Apr. 2021.

[20] S. Lee, H. Jang, Y. Baik, S. Park, and H. Shin, "KR-BERT: A small-scale korean-specific language model," *arXiv (Cornell University)*, Jan. 2020.

[21] M. Lewis and et al., "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[22] Y. Liu and et al., "Multilingual denoising pre-training for neural machine translation," *arXiv (Cornell University)*, Nov. 2020.

[23] C. Maklin, "Transformers explained," *Medium*, Aug. 2022. [Online]. Available: https://medium.com/@corymaklin/transformers-explained-610b2f749f43

[24] M. McKeever, "Linguistic hegemony and large language models," *Medium*, Jul. 2023. [Online]. Available: https://mittmattmutt.medium.com/linguistic-hegemony-and-large-language-models-fd9252855529

[25] P. Mishra, "Understanding t5 model: Text to text transfer transformer model," *Medium*, May 2021. [Online]. Available: https://towardsdatascience.com/understanding-t5-model-text-to-text-transfer-transformer-model-69ce4c165023

[26] R. Navigli, S. Conia, and B. Ross, "Biases in large language models: Origins, inventory, and discussion," *J. Data and Information Quality*, vol. 15, no. 2, jun 2023. [Online]. Available: https://doi.org/10.1145/3597307

[27] A. Neeleman and K. Szendrői, "Radical pro drop and the morphology of pronouns," *Linguistic Inquiry*, vol. 38, no. 4, pp. 671–714, 2007, accessed: Feb. 24, 2024. [Online]. Available: https://www.jstor.org/stable/40071411?seq=29

[28] G. Nicholas and A. Bhatia, "Lost in translation: Large language models in non-english content analysis," *arXiv (Cornell University)*, Jun. 2023.

[29] H. H. Park, K. J. Zhang, C. Haley, K. Steimel, H. Liu, and L. Schwartz, "Morphology matters: A multilingual language modeling analysis," *Transactions of the Association for Computational Linguistics*, vol. 9, pp. 261–276, Mar. 2021.

[30] K. Park, J. Lee, S. Jang, and D. Jung, "An empirical study of tokenization strategies for various korean nlp tasks," pp. 133–142, Dec. 2020.

[31] A. Rozovskaya and D. Roth, "Grammar error correction in morphologically rich languages: The case of russian," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 1–17, Mar. 2019.

[32] S. Ruder, "Why you should do nlp beyond english," *ruder.io*, Aug. 2020. [Online]. Available: https://www.ruder.io/nlp-beyond-english/#the-ml-perspective

[33] T. Shukla, "Junior research proposal: Digital humanities," Feb. 2024.

[34] M. Siavoshi, "The importance of natural language processing for non-english languages," *Medium*, Sep. 2020. [Online]. Available: https://towardsdatascience.com/the-importance-of-natural-language-processing-for-non-english-languages-ada463697b9d

[35] A. Solyman, W. Zhenyu, T. Qian, A. A. M. Elhag, M. Toseef, and Z. Aleibeid, "Synthetic data with neural machine translation for automatic correction in arabic grammar," *Egyptian Informatics Journal*, Dec. 2020.

[36] P. Soulos and et al., "Structural biases for improving transformers on translation into morphologically rich languages," *arXiv (Cornell University)*, Jan. 2022.

[37] C. Toraman, E. H. Yilmaz, ŞahinuçF., and O. Ozcelik, "Impact of tokenization on language models: An analysis for turkish," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 22, no. 4, pp. 1–21, Mar. 2023.

[38] A. Vaswani and et al., "Attention is all you need," *arXiv.org*, Dec. 2017. [Online]. Available: https://arxiv.org/abs/1706.03762

[39] A. Virtanen and et al., "Multilingual is not enough: Bert for finnish," *arXiv (Cornell University)*, Dec. 2019.

[40] Y. Wang, Y. Wang, K. Dang, J. Liu, and Z. Liu, "A comprehensive survey of grammatical error correction," *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 5, pp. 1–51, Oct. 2021.

[41] W. Wongso, H. Lucky, and D. Suhartono, "Pre-trained transformer-based language models for sundanese," *Journal of Big Data*, vol. 9, no. 1, Apr. 2022.

[42] S. Yoon and et al., "Towards standardizing korean grammatical error correction: Datasets and annotation," Jan. 2023.

[43] R. Zarkar, "Natural Language Processing — raghvendra.zarkar18," https://medium.com/@raghvendra.zarkar18/natural-language-processing-65f82c8dd7e0, [Accessed 01-05-2024].