JOURNAL OF
EMERGING INVESTIGATORS

Article

Rhythmic lyrics translation: Customizing a pre-trained language model using stacked fine-tuning

Jiwon Chong[1], Jaehyok Chong[2]
[1]Seoul Scholars International Math and Science, Seoul, South Korea
[2]SAP Labs Korea, Seoul, South Korea

SUMMARY

Neural machine translation (NMT) is a software that uses neural network techniques to translate text from one language to another. As NMT models are on the rise, the focus is on translating everyday mundane sentences. However, it is also necessary to start paying attention to the translation of domain-specific text, such as lyrics or poetry. For example, even one of the most famous NMT models—Google Translate—failed to give an accurate English translation of a famous Korean nursery rhyme, "Airplane" (비행기). To teach the model to retain specific information other than semantics, we need specific data which contains the exact information that we are attempting to teach. In the case of rhythmically accurate lyrics translation—translated lyrics that can be used to sing along to the original melody—we need corresponding data, containing lyrical and rhythmical properties, all the while being semantically accurate. However, as there is not enough data that fits our criteria, we propose a novel method we call 'stacked fine-tuning'. We fine-tuned a pre-trained model first with a dataset from the lyrics domain, and then with a smaller dataset containing the rhythmical properties, to teach the model to translate rhythmically accurate lyrics. To evaluate the effectiveness of our approach, we translated two famous Korean nursery rhymes to English and matched them to the original melody. Our stacked fine-tuning method resulted in an NMT model that could maintain the rhythmical characteristics of lyrics during translation while single fine-tuned models failed to do so.

INTRODUCTION

In human interactions, language is at the center of communication. With frequent interactions with people from other countries, real-time translators such as Google Translate or Naver Papago (especially for Korean-English translations) are becoming popular (1, 2). While most of these neural machine translation (NMT) models aim to provide semantically accurate—in terms of its literal meaning—translations, they fail to retain other characteristics that the sentences may have, such as rhythms or rhymes. As such, these models cannot translate sentences with specific properties such as poetry or lyrics. Currently, lyrics translation remains a task for professional translators. Automation of the process could be used in designing an app or a website that could take a song and translate it into another language with the same melody, lyrical qualities, and of course, semantics. There are two major parts to this: the translation of lyrics while maintaining its rhythm and the generation of the audio of the original artist singing the song in the newly translated language. However, as translating the lyrics is a challenging task itself, this study focuses on translating lyrics while maintaining their rhythmic accuracy.

We could not find any studies on translating lyrics, let alone doing so while maintaining rhythmic characteristics. However, we could find past studies on poetry translation, which is like lyrics in the sense that both have rhythm, rhymes, and sensory details that cannot be inferred by conventional NMT models (3–5). Researchers first used rhythm and rhyme constrained decoding methods, but the translations often failed to keep the semantics of the original poetry (3, 4). To preserve the semantics and figurative characteristics, other researchers used a pre-trained language model and fine-tuned the model with poetic data on multiple language pairs (5).

In translating lyrics, three problems needed to be addressed. The primary problem was the lack of data. For training NMT models, we needed a certain type of dataset known as parallel data, which are in the form of multiple sets of sentences and their translations (6). As neural machine translation has not yet been globally explored, it was incredibly difficult to find parallel data for less popular languages such as Korean, especially for specific domain datasets—in our case—lyrics (5, 6). The second problem was the task of retaining semantics and sensory details used within the lyrics. General-purpose translation models might translate the meaning and fluency, but not poetic style when translating poems (5). The same went for the semantics and sensory details when dealing with lyrics. The last problem was the task of retaining rhythmical accuracy within the translation of said lyrics. This was important because we wanted to use the translated lyrics for singing according to the original melody scores.

As a solution to the problems stated above, we propose the 'stacked fine-tuning' method. By fine-tuning a pre-trained model with a dataset from the lyrics domain and then once more with a dataset containing rhythmical properties, we can teach the model to translate rhythmically accurate lyrics. We hypothesized that by using our stacked fine-tuning method, we could train a model that met our objectives without using as much data as we would have needed. To test the effectiveness of our method, we trained models through single fine-tuning or stacked fine-tuning and compared the translation results of the models. We could observe that the models trained with stacked fine-tuning learned both the lyrical and rhythmic characteristics. In conclusion, by sequentially applying fine-tuning to a pre-trained model, we could teach the model multiple characteristics from multiple different domain datasets.

RESULTS

When we first tried to translate a famous Korean nursery rhyme, “Airplane” (비행기), with Google Translate and Naver Papago, we noticed that they failed to capture the proper form and meaning of the original lyrics (**Table 1**). We addressed this problem with the use of a semantically translated lyrics dataset. The purpose of said data was to teach the model the lyrical characteristics. However, there was a possibility that the translated lyrics dataset did not provide adequate data for learning translation itself, so we added some regular translation data to let the model learn translation before learning the specific characteristics for translation of lyrics. In addition to that, when we mapped the translated lyrics by Google Translate and Naver Papago on the music score, the mapping was not smooth and natural (**Figure 1**). Thus, additional fine-tuning with the rhythmically translated parallel data was also necessary to let the model learn how to constrain the translated lyrics into the appropriate rhythm.

To see the effect of stacked fine-tuning, we trained four different translation models: Wiki, Lyrics, Rhythm, and Rhythm+. For the training, we used parallel corpora extracted from public Wikipedia articles (wiki dataset), from K-pop lyrics and their translations (lyrics dataset), and from Korean nursery rhymes and their English translations matched with their original melody (rhythm dataset) (6–9). We fine-tuned the pre-trained multi-lingual model, mBART25, using combinations of our parallel data (10, 11). We fine-tuned the Wiki model with the wiki dataset only, the Lyrics model with the lyrics dataset only, and then we stacked fine-tuned the Rhythm model with the lyrics and rhythm datasets, and finally, we stacked fine-tuned the Rhythm+ model with all wiki, lyrics, and rhythm datasets. Using these models, we translated two famous Korean nursery rhymes: “Airplane” (비행기) and “Sunset” (노을).

Looking at the translations of “Airplane,” the two models fine-tuned only with one parallel data set, Wiki and Lyrics models, generated relatively shorter translations compared to the results of stacked fine-tuned models, Rhythm and Rhythm+ models (**Table 2**). We also noticed that the results of Wiki and Lyrics models resembled the translations of

Table 1. Translations of “Airplane” by a human, Google Translate, and Naver Papago.

Original (Korean)	떴다 떴다 비행기 날아라 날아라, 높이 높이 날아라 우리 비행기.
Human	Rising rising aeroplane fly away fly away, Higher higher fly away our aeroplane
Google Translate	Float, fly, fly, fly, fly, Fly high, our plane.
Naver Papago	Fly, fly, fly, fly, fly, fly Fly high, fly high, our plane.

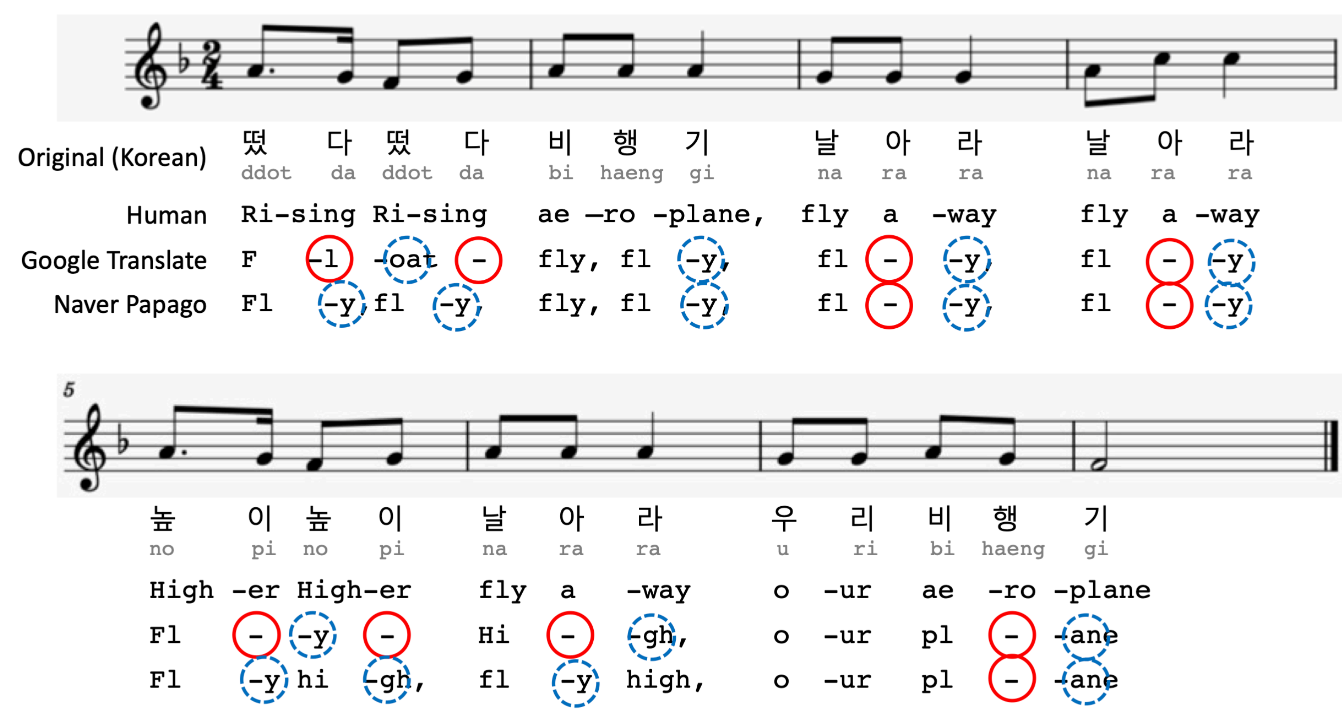

Figure 1. The score of the nursery rhyme “Airplane” with lyrics translated by a human, Google Translate, and Naver Papago. The translated lyrics are mapped to the original music score to best match the rhythm of the original song. A solid red circle indicates an abnormal stretch of a word by adding additional syllables, and a dashed blue circle indicates an abnormal stretch of a word by pausing.

Google Translate and Naver Papago. On the other hand, the translations of Rhythm and Rhythm+ models were more verbose than others (**Table 2**).

In detail, regarding the effects of fine-tuning with the lyrics dataset compared to the wiki dataset, the Lyrics model accurately translated the repetitive characteristics of “Airplane,” while the Wiki model failed to do so. This is shown through the phrases, “떴다 떴다”, “날아라 날아라”, and “높이 높이” in the original lyrics. The human translation emphasized these repetitive characteristics through phrases such as “Rising, rising”, “Fly away, fly away”, and “Higher, higher” (**Table 2**). Similarly, the Lyrics model outputs a translation more closely resembling the human translation, using repetitive phrases such as “Fly high, fly high, fly high” or “High, high, high”, while the Wiki model did not show any lyrical characteristics whatsoever (**Table 2**).

The lyrics translated by the Rhythm model also preserved the repetitive characteristics (**Table 2**). In addition to that, the translated lyrics could be more naturally mapped to the original melody (**Figure 2**). The Lyrics model required nine pauses and five additional syllables to correctly map to the original melody, while the Rhythm model required one pause and two additional syllables—a major improvement (**Figure 2**). When the lyrics were translated by the Rhythm+ model, we saw improvements in its rhythmic aspect even from that of the Rhythm model. The Rhythm+ model did not require any pauses or additional syllables to be perfectly mapped onto the melody (**Figure 2**).

Table 2. Translation of “Airplane” by fine-tuned translation models: Wiki, Lyrics, Rhythm, and Rhythm+.

Original (Korean)	떴다 떴다 비행기 날아라 날아라, 높이 높이 날아라 우리 비행기.
Human	Rising rising aeroplane fly away fly away, Higher higher fly away our aeroplane
Wiki	Swing up and flying, Swing up the airplane
Lyrics	Fly high, fly high, fly high, Let’s fly high, high, high
Rhythm	It's up it's up it's up airplane it's flying on it Let's fly up high and high up in the sky
Rhythm+	It's rising up it's rising airplane and flying up Let's fly up high and fly high up on our airplane

In the case of “Sunset,” the Wiki model failed to translate the last line of the original lyrics ‘빨갛게 노을이 타고 있어요’ and even the ‘있어요’ was omitted from the translated results. Additionally, each line of translation was short and lacked descriptive expressions (**Table 3**). The translation results of the Lyrics model were quite similar to the human translation, but the word ‘rising’ in the last line was an unexpected translation, which likely came from ‘타다’ in the original lyrics, a word which can mean either ‘burn’ or ‘ride’. The former is the intended meaning, but the Lyrics model seemed to take the latter meaning (**Table 3**).

Looking at the rhythmical aspect of the translation results, we compared Lyrics, Rhythm, and Rhythm+ models. When we mapped the translated lyrics to the original melody, the Rhythm model yielded a result that was much more rhythmically accurate compared to the Lyrics model

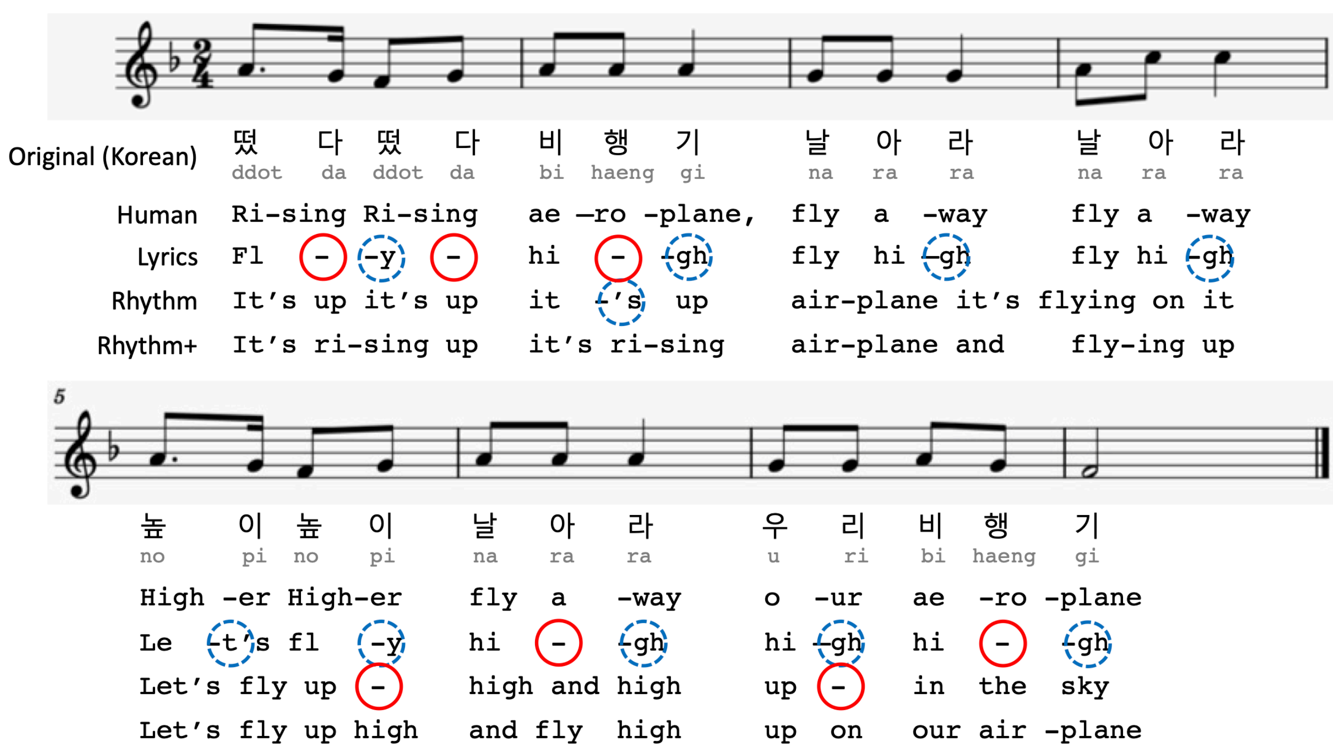

Figure 2. The score of the nursery rhyme “Airplane” with lyrics translated by Lyrics, Rhythm, and Rhythm+ models. The translated lyrics are mapped to the original music score to best match the rhythm of the original song. A solid red circle indicates an abnormal stretch of a word by adding additional syllables, and a dashed blue circle indicates an abnormal stretch of a word by pausing.

(**Figure 3**). While the Lyrics model required three pauses and nine additional syllables, the Rhythm model needed only five additional syllables and four pauses to match the melody (**Figure 3**). Compared to the results of "Airplane," the improvement seems marginal. However, considering that the lyrics of "Sunset" do not contain much repetition and instead contain several descriptive expressions, decreasing the translation by four additional syllables can be considered a major improvement. Mapping the translated lyrics of Rhythm+ to the melody, we could see almost no improvement compared to the result of Rhythm model: three pauses and six additional syllables (**Figure 3**).

Regarding the accuracy of the translation results, the Rhythm model loses a bit of its lyrical and/or semantic accuracy compared to the translations of the Lyrics model. An example is in the first line, where the Lyrics model accurately translated '들판' to 'field', while the Rhythm model translated it to 'woods' (**Table 3**). Furthermore, in the third line, while the Lyrics model correctly translated '색동옷' to 'colorful clothes,' the Rhythm model translated it to 'green cloths' (**Table 3**).

In the third and fourth lines, the translation of "Sunset" from the Rhythm+ model bears a greater resemblance to the Lyrics model in both composition and semantics (**Table 3**). Compared to the Rhythm model, the Rhythm+ model translated the third line to "On the autumn hill dressed in colored cloth," which resembled the translation of the Lyrics model more closely. Similarly, in the fourth line, the Rhythm+ model output "There's a red sunset in the sky," while the Rhythm model output "There's a red sky in the sunset." Again, the Rhythm+ model is more like the Lyrics model, which outputs "The red sunset is rising" (**Table 3**). However, when we looked at the second line, we can see that the Rhythm+ model completely fails, as it returns the phrase, "Mommy's worries fade away into the sunset" (**Table 3**).

The stacked fine-tuning method fulfilled our expectations of allowing the model to learn multiple characteristics from each domain-specific dataset by sequentially fine-tuning the pre-trained model with the datasets. In other words, this method allows us to teach the pre-trained model multiple characteristics from multiple data domains without having to find a specific dataset that belongs to every single domain that we are attempting to train.

DISCUSSION

As we used stacked fine-tuning using the lyrics and rhythm datasets on the pre-trained model, we could see the different effects that each fine-tuning stack had by observing the translation results of the Lyrics model and the Rhythm model. Looking at the translations of "Airplane," the Lyrics model accurately translated the repetitive characteristics such as "Fly high, fly high, fly high," but could not keep the rhythmic characteristics as the translation needed nine pauses and five additional syllables. However, the Rhythm model translated the lyrics to be mapped to the original melody only with one pause and two additional syllables also keeping the repetitive characteristics. This indicated that the Rhythm model learned the ability of keeping the rhythmic characteristics in addition to the ability of translating the repetitive characteristics of the Lyrics model.

Table 3. Translation of "Sunset" by various translation models: Wiki, Lyrics, Rhythm, and Rhythm+.

Original (Korean)	바람이 머물다 간 들판에, 모락 모락 피어나는 저녁 연기. 색동옷 갈아입은 가을 언덕에, 빨갛게 노을이 타고 있어요.
Human (Literal)	In the field where the wind stayed, Rising puffs of smoke from the dinner stove. On the hill wearing vibrant clothing of autumn leaves, The setting sun burns red.
Wiki	The wind stays in the field, Decay in the evening. Sweeping on the hills, 빨갛게 노을이 타고.
Lyrics	On the field where the wind stays, The evening smoke is blooming in all directions. On the autumn hill with colorful clothes, The red sunset is rising.
Rhythm	In the woods where the wind has stopped All around me is a spreading cloud of smoke Dressed in green cloth on the hill of autumn There's a red sky in the sunset
Rhythm+	In the woods where the wind has stopped Mommy's worries fade away into the sunset on the autumn hill dressed in colored cloth There's a red sunset in the sky

However, teaching the model a new characteristic seemed to result in the model losing a bit of accuracy in previously learned characteristics. In other words, teaching the model rhythmic characteristics resulted in the model's semantic and lyrical accuracy decreasing. When we evaluated the translation results of "Sunset," we could see that the Rhythm model lost a bit of its lyrical and/or semantic accuracy learned from the lyrics dataset, which can be seen through inaccurate word selections ('woods' and 'green cloth') and awkward expressions. A possible cause for this could be because the Rhythm model was attempting to change the words to fit the melody more accurately. However, it could also be because the Rhythm model was overfitted to the rhythm dataset. Since the rhythm dataset was too small and needed to be applied numerous times (720 epochs), it resulted in overfitting and decreased the overall accuracy of the model.

As the Rhythm model was not as semantically accurate, we tried reinforcing the translation's accuracy by starting with the wiki dataset, which has a larger vocabulary size. We hypothesized that, since stacked fine-tuning has proven to be effective in teaching the model multiple characteristics, we might be able to teach the model more expansive vocabulary by adding a fine-tuning stack in front of the Rhythm model's fine-tuning stacks. The resulting model was the Rhythm+ model which was more expressive than the Rhythm model. Looking at the translation results from the Rhythm+ and Rhythm models, we could see that both models included rhythmical qualities in their translations. Furthermore, when it comes to semantics or lyrical qualities, we could see that the Rhythm+ model gave better results than the Rhythm model, save for one phrase in the second line, "Mommy's worries

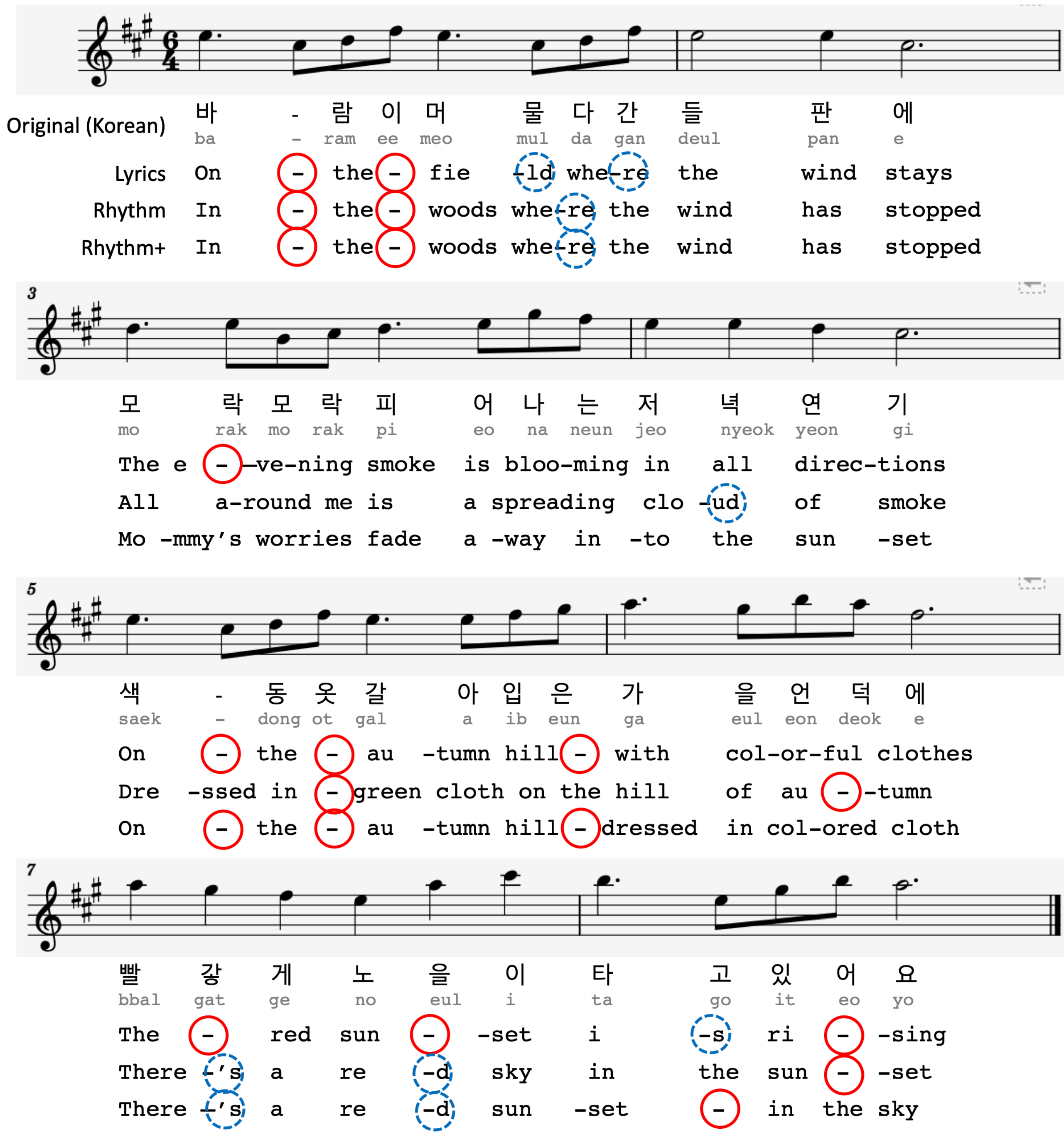

Figure 3. The score of the nursery rhyme "Sunset" with lyrics translated by Lyrics, Rhythm, and Rhythm+ models. The translated lyrics are mapped to the original music score to best match the rhythm of the original song. A solid red circle indicates an abnormal stretch of a word by adding additional syllables, and a dashed blue circle indicates an abnormal stretch of a word by pausing.

fade away in the sunset." This is because when we trained the Rhythm+ model, we trained with the rhythm dataset about 40 times more than with the lyrics dataset (800 epochs with the rhythm dataset versus 20 epochs with the lyrics dataset) due to the dataset's small size. This caused the model to be overfitted to the rhythm dataset, returning the phrase, "Mommy's worries fade away," which is a phrase that belongs to one of the nursery rhymes in the rhythm dataset.

Overall, we could see the effectiveness of stacking fine-tuning to teach the model multiple characteristics. This means that it will become much easier to train a highly specific model even when we cannot find a proper dataset containing all the necessary characteristics if we use the stacked fine-tuning method. However, when we stack fine-tuning, we must consider the loss of previously learned characteristics (12). Thus, we suggest applying the "freezing" technique to preserve previously learned characteristics (12, 13). Adding one layer for every fine-tuning stack while freezing the rest could negate the loss of previously learned characteristics. Moreover, as we tested only with the nursery rhymes, we suggest testing our model with a greater variety of songs to see the changes in performances according to the genre, lengths, or subjects of the songs.

The result of this study suggests that it is possible to teach a single NMT model multiple characteristics one by one through the stacked fine-tuning method introduced in this paper. This makes designing specialized language models a much more approachable task and possibly much less time-consuming since it both reduces the amount of necessary data and broadens the range of acceptable data for the task.

MATERIALS AND METHODS

There are many models and training methods for NMT, including the Bidirectional and Auto-Regressive Transformers (BART) and the Bidirectional Encoder Representations from Transformers (BERT) models (14–16). Among these pre-trained models, the BART model is a denoising autoencoder for pretraining a sequence-to-sequence model combining Bidirectional (encoder) and Auto-Regressive (decoder) Transformers (14). It is trained by corrupting text with an arbitrary noising function and learning a model to reconstruct the original text (14). The representations produced by BART can be used for text generation but also work well for comprehension tasks with minor modifications (14). Applying BART to large-scale monolingual corpora across many languages, a multilingual sequence-to-sequence denoising autoencoder, mBART, was proposed (10). mBART is trained once for all languages, providing a set of parameters that can be fine-tuned for any of the language pairs in both supervised and unsupervised settings, without task-specific or language-specific modifications or initialization schemes (10).

For this study, we chose the mBART25 model which was pre-trained using 25 monolingual languages extracted from the Common Crawl (CC25) corpus (10). In addition to that, the model could be fine-tuned without the model modifications and supported multiple languages, including our target languages, Korean and English (5). We downloaded the mBART25 model from the publicly available fairseq GitHub page (11).

For the Korean-English language pair, we collected three different parallel datasets: wiki, lyrics, and rhythm. For the wiki dataset, we used the WikiMatrix dataset which was the extraction of 135 million parallel sentences of 1,620 different language pairs in 85 different languages from public Wikipedia articles (6). We extracted the Korean-English pairs with threshold above 1.04 from the WikiMatrix corpora as previous literature suggested (6). As a result, 301,900 lines of pairs were used for the training (wiki dataset).

As the lyrics parallel dataset for NMT is not available, especially for the Korean-English translations, we collected the data from the LyricsTranslate.com website (7). 168,264 lines of Korean-English translations from 3,558 Korean songs were collected (lyrics dataset). Note that while the number of lines was quite numerous, the number of words was not as great since the length of each line was relatively short.

For the case of rhythmically accurate parallel lyrics dataset, especially for Korean-English translation, it was extremely difficult to come across a dataset that maintains the original rhythm without messing with the semantics. However, we discovered several nursery rhymes which were translated to be sung along in each language (8, 9). We collected 1,045 lines of Korean-English translations from 110 Korean nursery rhymes (rhythm dataset).

All three parallel datasets were tokenized using the sentence-piece model which was used in the training of mBART25. The byte-pair-encoding algorithm was used for subword segmentation. For the fine-tuning, we used an open-source toolkit named Fairseq with the common parameters, 0.3 dropout, 0.2 label smoothing, 2,500 warm-up updates, 0.00005 of the maximum learning rate, and 40,000 maximum training updates (17). For the update steps of each dataset, we fine-tuned the models with every 800 steps for the wiki and lyrics datasets and 80 steps for the rhythm dataset and compared their translations to choose the best resulting update steps.

Wiki model

To build the Wiki model, we fine-tuned the mBART25 with the wiki dataset. This model was made for testing the lyrics translation by the non-lyric translator. We used the checkpoint with 24,000 update steps in epoch 4.

$$\textit{Wiki model} = \textit{mBART25} + \textit{wiki dataset}_{\textit{24000 steps}}$$

Lyrics model

We also fine-tuned the mBART25 only with the lyrics dataset and created the Lyrics model. This model could accurately carry over lyrical characteristics in its translations. Nevertheless, this model still lacked the rhythmic characteristics of the song. The checkpoint we used was built with update steps of 24,000 in epoch 20.

$$\textit{Lyrics model} = \textit{mBART25} + \textit{lyrics dataset}_{\textit{24000 steps}}$$

Rhythm model

We further fine-tuned the Lyrics model with the rhythm dataset to learn the rhythmic characteristics: Rhythm model. This was the stacked fine-tuned model, and we further fine-tuned the Lyrics model 720 steps in epoch 104 with the rhythm dataset.

$$\textit{Rhythm model} = \textit{mBART25} + \textit{lyrics dataset}_{\textit{24000 steps}} + \textit{rhythm dataset}_{\textit{720 steps}}$$

Rhythm+ model

Starting from the Wiki model, we further trained with the lyrics and rhythm datasets. With the lyrics dataset, we fine-tuned 24,000 steps, and with the rhythm dataset, we further fine-tuned 800 steps in epoch 115.

$$Rhythm+\ model = mBART25 + wiki\ dataset_{24000\ steps} + lyrics\ dataset_{24000\ steps} + rhythm\ dataset_{800\ steps}$$

Because automated metrics cannot reliably measure the rhythmical or lyrical characteristics of the translation, we applied human-based evaluation using the following two Korean nursery rhymes: "Airplane" (비행기) and "Sunset" (노을) (5). For evaluation of the model's rhythmic translation capabilities, we matched the entire translated lyrics of each song to their score and counted the number of abnormal stretches through additional syllables or pauses. The less abnormal stretches mean more accurate translations.

Received: July 22, 2022
Accepted: April 21, 2023
Published: May 1, 2023

REFERENCES

1. "Google Translate." *Google.* translate.google.com. Accessed 3 Jul. 2022.
2. "Papago." *Naver.* papago.naver.com. Accessed 3 Jul. 2022.
3. Genzel, Dmitriy, *et al.* ""Poetic" Statistical Machine Translation: Rhyme and Meter." *EMNLP*, 2010, pp. 158-166.
4. Ghazvininejad, Marjan, *et al.* "Neural Poetry Translation." *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Association for Computational Linguistics, Jun. 2018, pp. 67-71. doi:10.18653/v1/N18-2011.
5. Chakrabarty, Tuhin, *et al.* "Don't Go Far Off: An Empirical Study on Neural Poetry Translation." *arXiv preprint arXiv:2109.02972v2*, Sep. 2021. doi:10.48550/arXiv.2109.02972.
6. Schwenk, Holger, *et al.* "WikiMatrix: Mining 135M Parallel Sentences in 1620 Language Pairs from Wikipedia." *arXiv preprint arXiv:1907.05791v2,* Jul. 2019.
7. "Lyrics Translations." *Lyrics Translations*. lyricstranslate.com. Accessed 3 Jul. 2023.
8. "우리 동요를 영어 가사로 봉황님의 글." *Suksuk.co.kr.* www.suksuk.co.kr/momboard/read.php?table=BEB_002&number=38845. Accessed 9 Jul. 2022
9. *Story of sunny*. blog.daum.net/koreanair777/3053930. Accessed 9 Jul. 2022.
10. Liu, Yinhan, *et al.* "Multilingual Denoising Pre-training for Neural Machine Translation." *Transactions of the Association for Computational Linguistics,* vol. 8, Nov. 2020, pp. 726-742, doi:10.1162/tacl_a_00343.
11. "fairseq/examples/mbart/README.md at facebook research/fairseq." *GitHub*. github.com/facebookresearch/fairseq/tree/main/examples/mbart. Accessed 16 Jul. 2022
12. Howard, Jeremy, and Sebastian Ruder. "Universal Language Model Fine-tuning for Text Classification." *arXiv preprint arXiv:1801.06146v5,* May 2018.
13. Felbo, Bjarke, *et al.* "Using Millions of Emoji Occurrences to Learn Any-domain Representations for Detecting Sentiment, Emotion and Sarcasm." *arXiv preprint arXiv:1708.00524v2,* Oct. 2017.
14. Lewis, Mike, *et al*. "Bart: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, Jul. 2019, pp.7871-7880.
15. Devlin, Jacob, *et al*. "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding." *arXiv preprint arXiv:1810.04805v2*, May 2019.
16. Vaswani, Ashish, *et al*. "Attention Is All You Need." *Advances in Neural Information Processing Systems,* vol. 30, 2017.
17. Ott, Myle, *et al.* "fairseq: A Fast, Extensible Toolkit for Sequence Modeling." *arXiv preprint arXiv:1904.01038v1*, Apr. 2019.