Korean-English Neural Machine Translation with Language Style Control

Stanford CS224N Custom Project

Jiwon Jeong
Department of Electrical Engineering
Stanford University
jeongjw@stanford.edu

Hyejin Lee
Department of Bioengineering
Stanford University
hjinlee@stanford.edu

Youjin Song
Department of Electrical Engineering
Stanford University
ujinsong@stanford.edu

Abstract

In this project, we developed a Neural Machine Translation (NMT) model capable of generating translated sentences with specified formality levels. We modified and fine-tuned a multilingual translation model, incorporating an additional input parameter for formality level in the Korean-English setting. Notably, even without adding a loss term for formality, our model demonstrated the capability to produce sentences that maintain the original meaning while varying in formality. Additionally, we explored various weight tuning methods and the potential for controlling sentiment through our framework. Our results indicate significant improvements in both semantic preservation and formality alignment compared to baseline models. This work highlights the feasibility and effectiveness of direct style control in NMT, paving the way for more nuanced and contextually appropriate translations.

1 Key Information to include

- Mentor: Moussa Doumbouya
- External Collaborators (if you have any): N/A
- Sharing project: N/A
- Team contributions: All three members contributed equally. Jiwon worked on dataset and evaluation. Hyejin and Youjin worked on fine-tuning the LLM model using two different approaches – Encoder output Extension and Categorical Token Fine-tuning, respectively.

2 Introduction

Recent large language models (LLMs) have demonstrated remarkable accuracy in translating sentences between languages while preserving their original meaning. However, in practice, merely focusing on meaning preservation may not suffice, as different language styles are employed depending on the context and audience. For instance, the level of formality in sentences varies based on the social setting and the interlocutors. Recognizing this need, our project aims to develop a neural translation model capable of style control for Korean and English. Specifically, our model translates Korean sentences into English while adhering to a specified formality level, thereby ensuring consistency with the source sentences' meaning.

The task of controlling the style in translations presents several challenges. One significant hurdle is the lack of parallel corpora with source and target sentences that exhibit different styles, particularly

for low-resource languages. Traditional approaches often require extensive datasets with varied style annotations, which are not readily available. Additionally, incorporating new loss functions to manage stylistic variations introduces substantial computational complexity and resource consumption, which we sought to avoid.

Our approach diverges from existing methods by leveraging the inherent formality variations present in pre-existing Korean-English parallel corpora. By avoiding the need for additional stylistic loss terms during training, we circumvent the associated computational overhead. Instead, we experiment with providing the neural machine translation (NMT) model with the target sentence's formality level during the decoding phase. This strategy enables the model to learn and apply the desired formality without extensive data augmentation or complex loss function integration. Additionally, we explored the possibility of applying the same approach to different styles, such as sentiment.

We employed a multilingual machine translation model as our foundation model and modified its structure in two distinct ways to process a target style and incorporate it into the translation output. The first approach, Encoder output Extension, involves extending the encoder output's embedding dimension to include style information directly. The second approach, Categorical Token Fine-Tuning, introduces style tokens that guide the model's output style. These methods were designed to explore the feasibility of direct style control in NMT and assess their impact on translation quality.

In summary, this project contributes to the field of style-controlled machine translation by demonstrating the feasibility of controlling translation styles without the need for additional stylistic loss functions or augmented data. This work paves the way for more nuanced and contextually appropriate translations, addressing a critical gap in the current capabilities of neural machine translation models.

3 Related Work

Controlling styles in machine translation has been explored through various methods. Wang et al. (2023) utilized prompts to control different styles, including modern and early English, honorific style Korean, and modern and classical Chinese. These approaches highlight the flexibility of prompt-based control in achieving stylistic outcomes in translation tasks.

Formality control has been extensively studied, often through post-editing or re-ranking translations. Zhang et al. (2022) and Vincent et al. (2022) proposed methods that involve post-processing steps to adjust the formality levels, relying on external formality scoring systems and re-ranking mechanisms to refine the output. While effective, these can introduce additional computational overhead and complexity.

A major challenge in formality control is the need for parallel corpora with different formality levels, particularly for low-resource languages Tyagi et al. (2023). To address this, some researchers have generated synthetic data for training. Rippeth et al. (2022) generated synthetic parallel corpora by converting the formality level of sentences, which were then used to train the translation models. Although useful, this approach depends heavily on the quality of the synthetic data, which may not always accurately reflect natural language variations. Iterative dual knowledge transfer frameworks have also been explored (Wu et al., 2021). These frameworks iteratively refine translation and style transfer tasks, improving the model's ability to handle formality variations without extensive parallel corpora.

In addition to these methods, some researchers have explored using Minimum Risk Training (MRT) to incorporate stylistic loss terms directly into the training process He et al. (2020). However, MRT introduces substantial computational complexity and resource consumption, making it less practical for large-scale applications.

Overall, while significant progress has been made in controlling styles in machine translation, the reliance on extensive parallel corpora, synthetic data, and computationally intensive training methods highlights the need for more efficient and scalable solutions. Our work contributes to this field by demonstrating that effective formality control can be achieved without additional stylistic loss functions or augmented data, paving the way for more practical and accessible style-controllable translation models.

4 Approach

4.1 Foundation model: facebook nllb-200-distilled-600M

We began with a multilingual NMT model, the facebook/nllb-200-distilled-600M (Costa-jussà et al., 2022). This model is capable of translating between multiple languages, including low-resource languages, and offers a favorable balance of performance and model size compared to other models of similar capability. It is a sequence-to-sequence multilingual machine translation model based on the Transformer encoder-decoder architecture and serves as the foundation model for our fine-tuning efforts described in the rest of the report.

4.2 Baseline model

Given the absence of an open-sourced, style-controllable Korean-to-English NMT model, we used our foundation model – a naive translator – in conjunction with a formality style transfer model (Damodaran, 2022) as the baseline for our formality-controllable NMT experiments. For our additional style input, sentiment, we explored several sentiment style transfer models (Shen et al., 2017; He et al., 2020; Yi et al., 2021; Wang et al., 2019; Li et al., 2018). However, none of these models produced satisfactory qualitative results for reversing the sentiment of general sentences. Consequently, we compared the performance of the foundational model without any sentiment style transfer against our proposed methods

4.3 Formality Level Control Without Stylistic Loss and Data Augmentation

Creating a machine translation (MT) model capable of style control requires addressing two main challenges. First, while some studies have introduced additional stylistic loss terms to generate translations that align with the desired style, directly backpropagating these losses through the model is often infeasible. To address this, researchers have utilized Minimum Risk Training (MRT). Although effective, MRT introduces substantial computational complexity and resource consumption, which we sought to avoid. Second, some methods require source-target parallel datasets with multiple style variants for each source sentence to facilitate training. However, most language pairs, particularly low-resource languages, lack such datasets. Instead, our dataset exhibited mismatches in formality levels between source and target sentences. We hypothesized that this property could be beneficial, as it might help the model learn to generate sentences with different formality levels from the source sentences.

Consequently, we experimented with providing the NMT model with the target sentence's formality level during the decoding phase. This approach leverages the natural formality variations present in the existing Korean-English parallel corpora, enabling the model to adjust the formality of the output without the need for additional stylistic loss terms or augmented data.

4.4 Our Approaches

APPROACH 1: Encoder output Extension

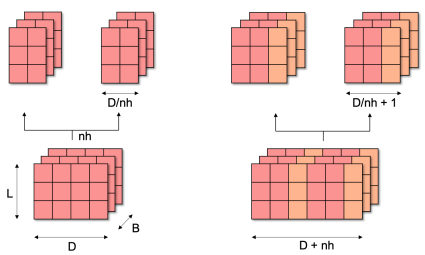

Figure 1: Encoder output extension for the multi-head attention mechanism. The left figure illustrates the initial few steps of the original mechanism, while the right figure depicts the extended version. B, D, L, and nh stands for batch size, embedding dimension, sequence length and number of attention heads, respectively.

We introduced an additional input parameter (style level) into the output of the encoder stack. Following the encoding process, we expanded the embedding dimension of both the encoder output and the decoder hidden state. Specifically, the embedding dimension was extended for each attention head, as illustrated in Figure 1. For the encoder output, this extended dimension was filled with a continuous style level, while for the decoder hidden state, it was filled with zero. Upon loading the pre-trained weights, the existing parameters were assigned to their corresponding positions, and the additional parameters were either initialized to zero or sampled from a normal distribution $\mathcal{N}(0, 0.25)$, and truncated to the range [-0.5, 0.5].

APPROACH 2: Categorical Token Fine-Tuning

We found that the pretrained translation model, facebook/nllb-200-distilled-600M (Costa-jussà et al., 2022), is capable of translating any pair of languages simply by switching the language token. By positioning the target language token at the beginning of the target sentence, i.e., `[Language_token] X [EOS_token]` with `X` being the tokenized source or target text, it functions as a BOS (beginning of sentence) token that also indicates the target language for the generated sentence. During the training phase, the model learns the function of this target language token as a language signal through the teacher forcing nature of the transformer. During inference, the first token is forced to the language token of the target language.

This was surprising, as a token at the front significantly affects the generation as a whole by influencing subsequent tokens. Given its strong signal, we hypothesized that it might also be able to control the style of the sentence, in our case, formality, by using a 'style token' without adjusting any model structure. Starting with model weights that already performed well for Korean to English translation, we inserted the style token(e.g. formality or sentiment token) immediately after the target language token, i.e., `[Language_token] [Style_token] X [EOS_token]`. During training, the model is expected to learn from the style token signals through teacher forcing. After sufficient training, the model should be able to output translated sentences with the desired level and/or class of language style during inference.

5 Experiments

5.1 Data

We used a dataset from 'The Open AI Dataset Project (AI-Hub, S.Korea)' which are widely used for KO-EN and EN-KO NMT tasks. First dataset is Korean Public AI Hub Parallel Corpora (Park et al., 2021) of 1.6M sentences from several areas including news, Korean culture, colloquial style, and conversational style. Since our objective is to generate the sentences spanning a wide range of formality, we aimed to use the sentences from the areas whose formality score distributions are not skewed. Therefore, we omitted the areas such as news whose sentences are mostly formal. The method we used to measure the formality score is demonstrated on Section 5.2. The training dataset consists of 400K sentences in colloquial style and 90K sentences in conversational style.

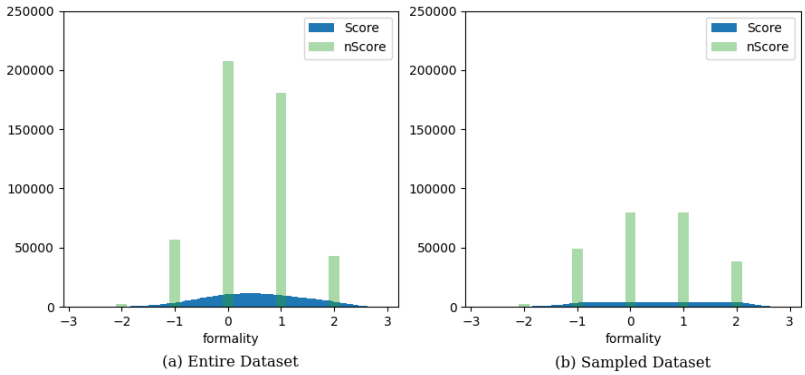

Figure 2: Formality score distribution of the entire dataset and sampled dataset. Score is a raw output of the formality regressor, and nScore is Score discretized into 5 levels.

Figure 2 (a) shows the formality score distribution of the entire training dataset of 490K sentences. To make the distribution more uniform, we first divided the score range into 100 bins. For each bin having more than 10K sentences, we randomly sampled 10K sentences. Figure 2 (b) illustrates that the sampled dataset contains less sentences having moderate formality score. We used this sampled dataset for training.

For the test dataset, we used 10K sentences in conversational style since the difference in formality is usually demonstrated well in conversations. For formality scores which will be given to the model, we utilized two types of approaches. First, since the baseline model can only generate formal or informal sentences, we introduced extreme formality scores so that we can compare the performance of our model to that of the baseline model. For APPROACH 1(embedding extension model), target formality score -2 and 2 were given for each sentence in the test dataset, and target formality token 1 and 5 were given for APPROACH 2(categorical token fine-tuning model). Next, to validate if the model can generate the sentences with different levels of formality, we applied random formality scores. For each sentence in the test dataset, we randomly assigned two integer formality scores in range of -3 to 3 and two formality tokens within 1, 2, 3, 4 and 5. Since two consecutive scores or tokens may not display enough difference in formality even in the sense of human judgement, we constrained two scores or two tokens to have a difference greater than or equal to 2.

For sentiment control as the extension of our approach, we added sentiment score for each sentence in the training set measured by the method illustrated on Section 5.2, and assigned two random scores within -1 (negative), 0 (neutral), and 1 (positive) for each sentence in the test set.

5.2 Evaluation method

Our model is evaluated across three dimensions: meaning preservation, formality, and sentiment. First, we defined three scores corresponding to each dimension as described below. Given the pairs consisted of the source sentences from the test dataset and their translated versions generated by the model, we calculated those scores for each pair and used the average score for each dimension to demonstrate the performance of the model. For meaning preservation and formality, we referred to the best practices for automatic evaluation in style transfer (Briakou et al., 2021) which strongly correlates with human judgements.

We used chrF (Popović, 2015) to calculate the similarity between the source sentence and the model output in terms of semantics. This metric is based on the character *n*-gram F-score, and it is known for having good correlations with human judgements on both system-level and segment-level.

To compute the formality of a sentence, we utilized a model that fine-tuned *XLM-R* (Conneau et al., 2020) on English formality ratings (Pavlick and Tetreault, 2016) with the Adam optimizer, a batch size of 32, and a learning rate set to 5e-5 for 5 epochs (Briakou et al., 2021). This formality regressor can measure the formality of a given sentence with a score between -3 (informal) and 3 (formal). For baseline model and APPROACH 1, we directly compared the target formality score and the formality score of model output by calculating the difference between two. However, since APPROACH 2 utilizes discretized tokens for formality score, we assigned integer formality score between -2 to 2 to each formality token and computed the difference with the translated sentence's formality score.

Regarding the sentiment dimension, we made use of TimeLMs (Loureiro et al., 2022) fine-tuned for sentiment analysis with the TweetEval (Barbieri et al., 2020) benchmark. We defined a sentiment score by subtracting the probability of a sentence being negative from that of a sentence being positive, resulting in a real value between -1 (negative) and 1 (positive). Similar to formality transfer evaluation, we calculated the difference between target sentiment score and the model output's sentiment score.

5.3 Experimental details

As the foundation model, we used the facebook nllb-200-distilled-600M model, following the NLLB (Costa-jussà et al., 2022) team's configuration. The learning rate and weight decay were set to 10^{-4} and 10^{-3}, respectively, with a batch size of 16. Our loss converged quickly due to the pretrained weights. However, a small loss and a slower convergence rate did not necessarily indicate that our model was sufficiently trained; while the output sentences began to make sense, they still exhibited subtle semantic differences even when the loss was very small. After at least 10,000 iterations, we

monitored and manually stopped training based on translation quality and alignment with the target formality level. Training took about 12 hours on a Titan RTX.

5.3.1 Efficient Initialization and Weight tuning

We started by controlling the continuous formality level (from least formal to most formal) using the data mentioned in Section 5.1. For APPROACH 1, we applied the formality level ranging from -3(informal) to 3(formal) to the output of the transformer's encoder stack, specifically to the expanded dimension of the weights. For APPROACH 2, we inserted one of five formality style tokens immediately after the language token, i.e., `[Language_token] [FormalityLevel_token] X [EOS_token]`, to provide a formality signal to the model.

For both approaches, we compared convergence speed and model performance after at least 20,000 iterations for different ranges of freezing weights and initializing methods in the transformer. For the first approach of extending weights, we experimented with two settings for each initialization method (all zero or randomly sampled from a normal distribution) and freezing range (not freezing and freezing encoder weights only). For the second approach of categorical token fine-tuning, we experimented with 3 ranges of freezing weights: training all weights, freezing only the encoder, and freezing both the encoder and decoder. All experiments started from the pre-trained weights of the foundation model. Further details on the token vocabulary expansion process and the rationale for weight tuning are explained in Appendix B.

5.3.2 Exploring Other Language Style Transfer

After determining the most efficient way of fine-tuning the model, we explored whether different language style transfers (such as sentiment) could work well. We utilized the sentiment analysis technique described in Section 5.2. In APPROACH 1, we used a continous sentiment level ranging from -1 (negative) to 1 (postive) and in APPROACH 2, we discretized sentiment into three levels: negative, neutral, and positive, and added these three style tokens to our tokenizer and transformer embedding vocabulary. We also tried to utilize both formality and sentiment styles, by extending the embedding twice, and forcing style token as `[Language_token] [FormalityLevel_token] [SentimentLevel_token] X [EOS_token]`, respectively. We evaluated the model by measuring the distance between the target sentiment score and the output sentiment score.

5.4 Results

5.4.1 Efficient Initialization and Weight tuning

We evaluated the model performance using two types of test sets, each with different target formality score settings: "extreme" and "random" (as in Section 5.1). The models generated two translated sentences from a single source text using the two target formality levels and calculated the formality distance from the target formality score. We also measured semantic similarity to ensure the translator functions well while modifying the formality level. The quantitative results are shown in Table 1.

		baseline	APPROACH 1				APPROACH 2		
			no freeze zero init	no freeze rand init	freeze enc zero init	freeze enc rand init	no freeze	freeze enc	freeze enc&dec
extreme	Semantic ↑	0.353	0.405	0.416	0.413	0.414	0.436	0.430	0.463
	Formality ↓	1.405	1.054	1.095	1.081	**1.050**	1.388	**1.291**	1.823
random	Semantic ↑	–	0.404	0.414	0.412	0.412	0.448	0.449	0.465
	Formality ↓	–	1.001	1.029	1.032	**0.973**	1.056	**1.009**	1.370

Table 1: Table above compares quantitative results of our formality style transfer approaches. "Semantic" indicates how well the semantic meaning of the source sentence is preserved, measured by the chrF score (higher is better). "Formality" indicates alignment with the target formality level, measured by the difference in formality scores (lower is better).

Compared to the baseline model, both of our methods excelled in semantic meaning preservation and formality style transfer. The level of meaning preservation remained similar across different weight tuning methods, while formality alignment showed more significant differences. This might be because we started with a well-performing language translator model, so most of our training

iterations focused on training the extended style parameters (APPROACH 1) or the new style token (APPROACH 2) rather than semantic language understanding.

Evaluating with extreme formality levels was generally more challenging than with random formality levels. This could be because it is inherently more difficult to make sentences very formal or very informal, and also due to a comparable lack of data for those extreme cases in our training dataset.

For both of our approaches, freezing only the encoder outperformed the other methods, even surpassing the performance of training all the weights. For APPROACH 2, training only the shared weights (freezing both the encoder and decoder) performed poorly, showing worse formality alignment than the baseline. Our rationale for freezing only the encoder was based on the expectation that the encoder output encapsulates the semantic information of the source sentence, while the decoder is responsible for generating a sentence in the target style using this encoded information and the style token. Given our goal to preserve semantic meaning in this style transfer task, freezing the encoder proved to be highly efficient. By starting with a model that already excels at extracting semantic meaning from the source sentence, freezing the encoder allowed us to fully leverage this capability.

Additionally, for APPROACH 1, randomly initializing the extended parameters performed slightly better than zero initializing them. We suspect that randomly initializing the parameters introduced diverse starting points for the parameters, which led to faster and more effective learning. Using a normal distribution with zero mean and a small standard deviation helped avoid instability during training, thus improving the overall performance.

5.4.2 Exploring Other Language Style Transfer

As our approach of adding style information at the decoding phase seems to be effective, we decided to explore other language styles and potentially incorporate two styles simultaneously. We used formality as the first style, and selected sentiment for the second as it is one of the widely used styles for text style transfer tasks and various open-sourced sentiment analysis tools are available. Compared to the baseline model, which resulted in a sentiment distance of 0.774, our model with APPROACH 1 and APPROACH 2 achieved slightly better scores of 0.758 and 0.763, respectively. When both formality and sentiment styles were utilized, the sentiment distance and formality distances were 0.759, 1.002 and 0.761, 0.996 for APPROACH 1 and APPROACH 2, respectively.

While it appears that style transfer for two styles is feasible, sentiment style transfer poses challenges. This is primarily because most sentiment pairs inevitably differ in semantics, for example, "I like this food" versus "I don't like this food." This setup may have introduced contradictions during training since the translator fundamentally needs to convey the same meaning as the source sentences. If there exists a style with a sufficient dataset or classifier that exhibits differences only in style while preserving meaning, our method could perform better. Importantly, our method has opened up the possibility of style transfer for multiple styles simultaneously, without modifying the model structure or compromising training efficiency.

6 Analysis

Following Figure 3 provides qualitative results for both of our approaches. Given a single sentence, we varied the formality level from very informal to very formal. For APPROACH 1, this ranged from -3 to 3 on a real number scale, and for APPROACH 2, we used 5 formality tokens. Despite these changes, the semantic meaning of the sentence is highly preserved.

Additionally, we wanted to assess how effectively our approaches signal the desired formality level compared to the model before fine-tuning. Our foundation model, naive machine translator, produces sentences with varying formality levels primarily by detecting the formality of the source sentences and attempting to generate translated sentences that match this level. Note that its ability to discern formality is minimal, showing differences only when the formality disparity is significant.

As shown in Figure 3, we generated sentences with varying formality levels for two input sentences that only differed in formality. Although their meanings are exactly the same, the formality levels are distinctly different, evident to any Korean speaker, even beginners. The foundation model produces different sentences based on these formality levels. It is noteworthy that our fine-tuned model still produces sentences ranging from less formal to more formal, although it remains slightly biased toward the formality tone of the input.

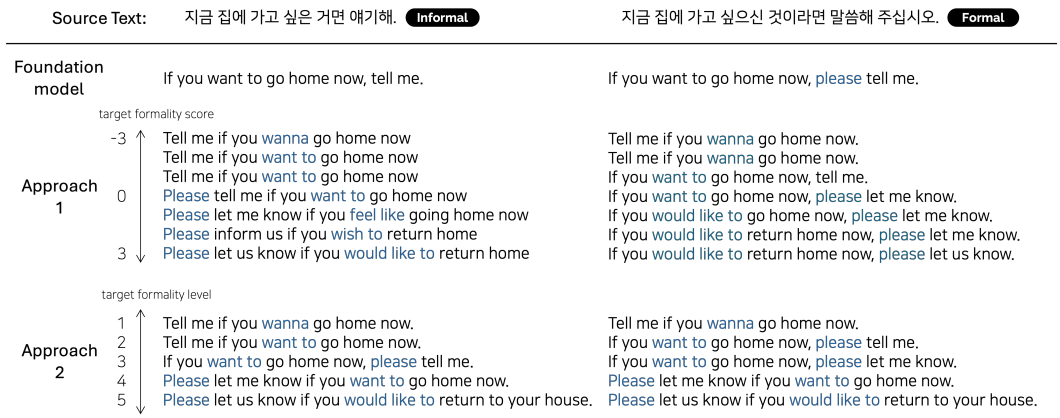

Figure 3: Comparison of sentence generation with varying formality levels. The foundation model produces very minor differences in sentences (with and without 'please') reflecting the formality level of the inputs. Our fine-tuned model, while slightly biased toward the formality tone of the input, successfully generates sentences across a range of formality levels from less formal to more formal. The vocabulary choice for verbs (e.g., 'want' vs. 'wanna') and nouns (e.g., 'go' vs. 'return') varies, as well as expressions of politeness and formality, such as 'please,' across different formality levels.

Although our model successfully generates sentences with different formality levels, its performance is better when generating formal tones rather than informal ones. This is likely due to the bias in our dataset: sentences with a formal tone significantly outnumber those with an informal tone.

There are also some fundamental challenges in the language style transfer task in general. Since the 'style' and 'content' of a language cannot be completely separated, even for humans, it is difficult to label or classify the style of a sentence accurately. Formality style was comparatively easier to manage because it often has a direct counterpart in being formal or informal with exactly the same meaning. However, it was more challenging for sentiment or emotion transfer. It was nontrivial to find pairs of sentences that varied only in sentiment or emotion while conveying exactly the same meaning.

7 Conclusion

In conclusion, our research have focused on nuanced control of stylistic attributes at translation from Korean to English. One of our significant contributions is that we've opened up the possibility of efficient language style transfer in translation. With over 7,000 languages globally, there are linguistic styles unique to specific languages, some of which may be explicitly present in one language while absent in others. Through our method, anyone possessing a dataset that pairs target sentences with specific style levels or types can seamlessly incorporate these stylistic nuances.

To enhance the efficiency of the pipeline, we have experimented with various initialization methods and weight-freezing techniques. Furthermore, we have achieved this feat without introducing additional loss functions or adopting additional data augmentation techniques for style transfer, relying solely on the cross entropy loss inherent in the transformer. This success owes much to our strategic selection of parallel translation datasets, which ensure the preservation of semantic meaning between source and target texts without necessarily aligning with formalities.

Our model did not explicitly introduced new loss function for formality. If we could find a way to integrate such loss models without the need for tokenizing processes or effectively propagate such losses during training, our training could be even more efficient. We also wish to mention that our training dataset had only one reference translation for each sentence, as in Niu et al. (2017)'s work. If we can make use of more reference translations with various formality and sentiment levels for each source sentence, our model performance will be improved further. Additionally, the unavailability of open-source Korean formality measuring models hindered our ability to experiment with English-Korean translation with formality control. Acquiring such a classifier or dataset would allow us to easily apply it to our model and further enhance its capabilities.

8 Ethics Statement

Our model has a risk of generating output that contains toxic language, since our dataset used for fine-tuning included slangs. Based on the fact that slangs are typically classified as informal, the model may have been trained to use toxic vocabularies if the target formality is set extremely informal. This issue is also observed from the output of our foundation NMT model, as it translates the abusive language from source sentence without any filtering. One way to mitigate this risk is applying toxicity mitigation methods such as the one proposed by Bhan et al. (2024) to our model's output. Furthermore, we can ensure that the model avoids generating specific words by implementing a logit processor at the end of our transformer, such as `NoBadWordsLogitsProcessor`. This processor ensures that specified sequences are never selected by setting the probability of such tokens to negative infinity.

The generation of false translations is another possible concern of our model. Although our foundation model generally maintains the semantics of the source sentence well, our model often modifies the meaning if the target formality is set to extreme. Unlike other tasks using a single language, misinformation can be especially problematic in translation tasks since the users typically lack an understanding of the target language. As we interpreted this observation was due to the lack of training data having extreme formality scores, utilizing more of such data would help the model not to alter the meaning of the sentence.

References

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa-Anke, and Leonardo Neves. 2020. TweetEval:Unified Benchmark and Comparative Evaluation for Tweet Classification. In *Proceedings of Findings of EMNLP*.

Milan Bhan, Jean-Noel Vittaut, Nina Achache, Victor Legrand, Nicolas Chesneau, Annabelle Blangero, Juliette Murris, and Marie-Jeanne Lesot. 2024. Mitigating text toxicity with counterfactual generation.

Eleftheria Briakou, Sweta Agrawal, Joel Tetreault, and Marine Carpuat. 2021. Evaluating the evaluation metrics for style transfer: A case study in multilingual formality transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1321–1336, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Prithiviraj Damodaran. 2022. Styleformer. `https://github.com/PrithivirajDamodaran/Styleformer`.

Junxian He, Xinyi Wang, Graham Neubig, and Taylor Berg-Kirkpatrick. 2020. A probabilistic formulation of unsupervised text style transfer.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. In *North American Association for Computational Linguistics (NAACL)*.

Daniel Loureiro, Francesco Barbieri, Leonardo Neves, Luis Espinosa Anke, and Jose Camacho-collados. 2022. TimeLMs: Diachronic language models from Twitter. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 251–260, Dublin, Ireland. Association for Computational Linguistics.

Xing Niu, Marianna Martindale, and Marine Carpuat. 2017. A study of style in machine translation: Controlling the formality of machine translation output. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2814–2819, Copenhagen, Denmark. Association for Computational Linguistics.

Chanjun Park, Midan Shim, Sugyeong Eo, Seolhwa Lee, Jaehyung Seo, Hyeonseok Moon, and Heuiseok Lim. 2021. Empirical analysis of korean public ai hub parallel corpora and in-depth analysis using liwc.

Ellie Pavlick and Joel Tetreault. 2016. An empirical analysis of formality in online communication. *Transactions of the Association for Computational Linguistics*, 4:61–74.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Elijah Rippeth, Sweta Agrawal, and Marine Carpuat. 2022. Controlling translation formality using pre-trained multilingual language models.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment.

Harshita Tyagi, Prashasta Jung, and Hyowon Lee. 2023. Machine translation to control formality features in the target language.

Sebastian T. Vincent, Loïc Barrault, and Carolina Scarton. 2022. Controlling formality in low-resource nmt with domain adaptation and re-ranking: Slt-cdt-uos at iwslt2022.

Ke Wang, Hang Hua, and Xiaojun Wan. 2019. Controllable unsupervised text attribute transfer via editing entangled latent representation. In *NeurIPS*.

Yifan Wang, Zewei Sun, Shanbo Cheng, Weiguo Zheng, and Mingxuan Wang. 2023. Controlling styles in neural machine translation with activation prompt. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2606–2620, Toronto, Canada. Association for Computational Linguistics.

Xuanxuan Wu, Jian Liu, Xinjie Li, Jinan Xu, Yufeng Chen, Yujie Zhang, and Hui Huang. 2021. Improving stylized neural machine translation with iterative dual knowledge transfer. In *International Joint Conference on Artificial Intelligence*.

Xiaoyuan Yi, Zhenghao Liu, Wenhao Li, and Maosong Sun. 2021. Text style transfer via learning style instance supported latent space. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3081–3087, Yokohama, Japan.

Daniel Zhang, Jiang Yu, Pragati Verma, Ashwinkumar Ganesan, and Sarah Campbell. 2022. Improving machine translation formality control with weakly-labelled data augmentation and post editing strategies. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 351–360, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

A Encoder output Extension and Parameter Initialization

To incorporate style information by extending the embedding dimension of the encoder hidden state, we focused on the multi-head attention algorithm. In multi-head attention, multiple attention heads calculate attention scores for divided hidden states. If we naively extend the hidden state dimension by 16 (the number of attention heads) at the end, it would result in the last attention head only processing the style information. To avoid this, we extended the hidden state dimension (1024) to 1040 by increasing the hidden state embedding dimension for each attention head.

Consequently, the decoder parameters were also extended. Parameters with one dimension (i.e., norm, bias) that have a shape of [1024] were extended to [1040], and 2D tensors with shapes [4096, 1024], [1024, 1024], and [1024, 4096] were extended to [4096, 1040], [1040, 1040], and [1040, 4096], respectively. For initialization, each attention head's first 64 parameters were set to the pre-trained values, and the last 1 was initialized to 0 or sampled from a truncated normal distribution. This allowed our model to perfectly retain its translation capacity when initialized to 0, and partially retained translation performance when randomly initialized.

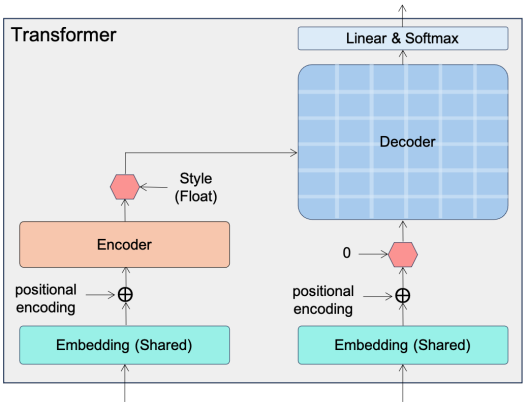

Figure 4: Embedding dimensions of both the encoder output and decoder hidden state were extended. The encoder output was extended with the style, represented as a float, and the decoder hidden state was zero-padded. They then went through the adjusted decoder layer to get the final output.

Sampling from a truncated normal distribution with zero mean and a small standard deviation was crucial. When we experimented with random numbers or a standard deviation of 1, the model completely lost its ability to translate, and the training became very unstable. Even after 60,000 iterations, the model produced outputs like "the the the the the the the". By using a small standard deviation, we ensured that the model retained some of its pre-trained capabilities, making the training process more stable and effective.

One small consideration is that we extended the dimensions from 1024 to 1040, which is not a power of 2. In neural networks, dimensions that are powers of 2 are often preferred for their computational efficiency. Changing 1024 to 1040 might affect resource efficiency. Future work could explore methods to extend hidden state dimensions in a way that minimally impacts resource efficiency.

B Token Vocabulary Expansion and Efficient Weight Tuning

Expanding the token vocabulary and forcing the token to be positioned at the front requires an understanding of the entire pipeline. Our base language translation pipeline consists of two main modules: the tokenizer and the transformer. The tokenizer converts natural language into a sequence of tokens at the beginning of the process so that the model can process it as numerical vectors, and it converts the translated tokens back into natural language at the final stage. The transformer learns to generate tokenized output in the target language given tokenized inputs in the source language by training shared embeddings (also called 'shared weights' since they are shared for both the encoder and decoder) and non-shared weights of the encoder and decoder.

The number of vocabularies in the tokenizer and shared embedding must match, as the transformer expands a given language token (a single number) into a high-dimensional vector (1024-D in our case) at the shared embedding layer. Adding any number of tokens (we discretized formality into 5 levels and added 5 special tokens corresponding to each level) requires expanding the vocabulary size of both the tokenizer and the embedding layer of the transformer. Expanding the vocabulary size of the tokenizer is straightforward, as it only needs to assign a unique ID to each added vocabulary. For expanding the vocabulary size of the embedding layer, we ensured that the embedding vectors for pre-existing vocabularies remained the same while initializing vectors for the added vocabularies. In summary, we added 5 style tokens to the tokenizer and expanded the embedding vocabulary size in the transformer while keeping other weights fixed.

The shared embedding must not be frozen since we initialized the added embedding of style token.

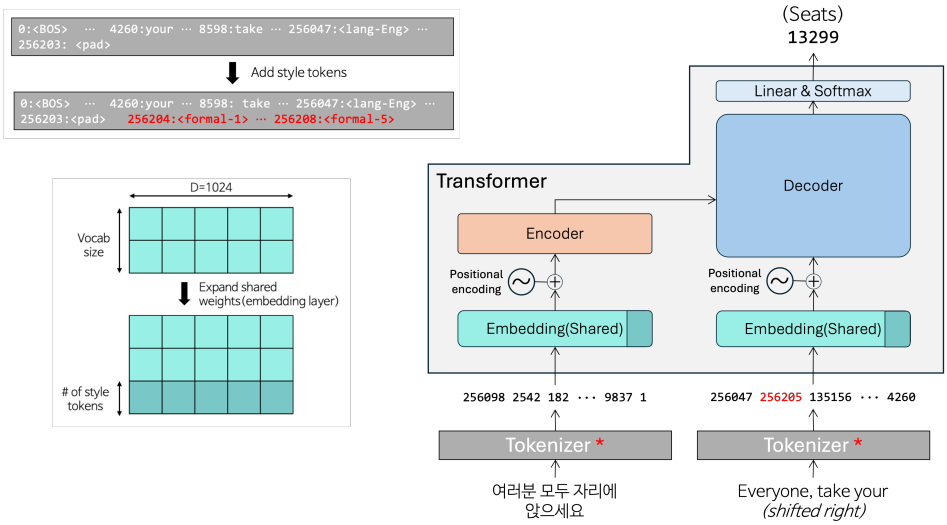

Figure 5: Token vocabulary expansion process and our machine translation pipeline

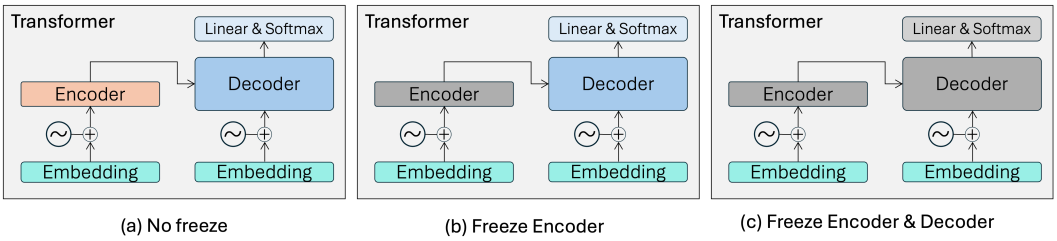

Figure 6: We tried different ranges of freezing weights: training all weights without any freezing, freezing only the encoder, and freezing both the encoder and decoder (with only the shared embedding weights being trained)