문학석사 학위논문

A Study of the Explainability on Prompt-tuning through Transfer Learning

전이 학습을 통한 프롬프트 튜닝의 설명가능성 연구

2023 년 02 월

서울대학교 대학원

언어학과 언어학전공

김 은 진

A Study of the Explainability on Prompt-tuning through Transfer Learning

지도 교수 신 효 필

이 논문을 문학석사 학위논문으로 제출함
2023 년 02 월

서울대학교 대학원
언어학과 언어학전공
김 은 진

김은진의 문학석사 학위논문을 인준함
2023 년 02 월

위 원 장 _____ 이 상 아 (인)

부위원장 _____ 신 효 필 (인)

위 원 _____ 김 문 형 (인)

A Study of the Explainability on Prompt-tuning through Transfer Learning

Advising Professor, Dr. Hyopil Shin

Submitting a master's thesis of Art

February 2023

Graduate School of Humanities
Seoul National University
Linguistics Major

Eunjin Kim

Confirming the master's thesis written by
Eunjin Kim
February 2023

Chair	Sangah Lee	(Seal)
Vice Chair	Hyopil Shin	(Seal)
Examiner	Munhyong Kim	(Seal)

Abstract

Kim, Eunjin

Department of Linguistics

The Graduate School

Seoul National University

Since training continuous prompts is a parameter-efficient way to tune a Pre-trained Language Model (PLM) on a target task, recent works suggest various training methods utilizing continuous prompts. However, few studies investigate the explainability of continuous prompts, which is critical to enhancing the confidence of PLM in a real-world scenario. To deal with the problems of the unexplainable continuous prompts, this study explores the effects of Prompt-tuning v1 (Lester et al., 2021) and Prompt-tuning v2 (Liu et al., 2022) on PLM.

More precisely, we conducted the experiments using a multilingual GPT to generalize our observations both on tasks and languages. We also analyzed the results of transfer learning using continuous prompts. We first confirmed whether continuous prompts are gathered according to tasks or languages, and then analyzed how continuous prompts utilize PLM in terms of the three main architectures of GPT: the attention mechanism and the activated neurons, and the label space.

In this study, we tried to answer the following research questions: (1) Can we distinguish continuous prompts according to the encoded information about target tasks or target languages? (2) Can we find any explainable patterns in the changes in

the attention mechanism after Prompt-tuning? (3) Can we observe any explainable patterns in the activated neurons of continuous prompts through layers? (4) Can we capture that continuous prompts interact with the label space of PLM?

First, we find that continuous prompts have different space according to the encoded information about target tasks. Second, continuous prompts exploit the attention mechanism of PLM by using the attention heads that encode the content-dependent information. Third, the activated neurons have task-specific information in the deeper layers. However, the second to last layer has rather task-common neurons. Lastly, despite the low isotropy, continuous prompts make the decoding token closer to the label words. Overall, we observe consistent results after transfer learning. As a result, we conclude that continuous prompts are trained while employing the knowledge PLM obtained during pre-training to solve the target task.

Keyword : Natural Language Processing, GPT, Prompt-tuning, Continuous Prompts, Multilinguality, Explainability
Student Number : 2021-21283

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

In Natural Language Processing (NLP), one of the traditional ways to solve Natural Language Understanding (NLU) and Natural Language Generation (NLG) tasks is fine-tuning a Pre-trained Language Model (PLM) with a full labeled dataset. Since the Transformer-based PLMs, such as Transformer (Vaswani et al., 2017), BERT (Devlin et al., 2019), and GPT-2 (Radford et al., 2019), have achieved the state-of-the-art for various NLP tasks, many studies have focused on the *pre-train and fine-tune* paradigm. However, this paradigm has several inevitable drawbacks.

One of the biggest problems of fine-tuning is that its objective is different from the objective of pre-training. In language modeling, the model is pre-trained to predict the next word (e.g., for GPT-2) or the masked word (e.g., for BERT) in a sentence. On the other hand, during fine-tuning, a new added linear classifier is trained to predict the label on a target task. Also, the context embeddings fed to the linear classifier compress the input sentence so much that we cannot assure that sufficient information is provided to the model to solve the target task. Besides, it is hard to know whether the model exploits the knowledge gained after pre-training.

To deal with such problems of fine-tuning, discrete prompts were first suggested to close the gap between the objective functions of pre-training and fine-tuning. Discrete prompts consist of a natural language template and a verbalizer, where the model should predict the verbalizer corresponding to the gold label. In this way, we maintain the model structure of pre-training so that PLM can employ the knowledge obtained from pre-training. However, the prompt engineering including the template design and the verbalizer design requires a human effort.

Subsequently, continuous prompts were proposed, which are embedding parameters that can be used without the manual prompt design. P-tuning (Liu et al., 2021) showed high performances in several NLU tasks when training GPT-2 with continuous prompts. Furthermore, toward parameter-efficient learning, Lester et al. (2021) suggested Prompt-tuning v1, where only continuous prompts are trainable. After Prompt-tuning, we just need to save the trained continuous prompts which have much fewer parameters than the PLM. They showed that Prompt-tuning is useful for transfer learning because once trained continuous prompts can be re-used for other tasks. Finally, Prompt-tuning v2 was suggested by Liu et al. (2022). Deep continuous prompts are injected into every layer of PLM. With more parameters, the performances rise close to the performances of fine-tuning on various NLU tasks in the fully-supervised setting.

Despite the great promise of the *pre-train and prompt-tune* paradigm, it is problematic that the difficulty of faithfully interpreting continuous prompts in natural language could potentially lead to concealed adversarial attacks (Khashabi et al., 2022). For example, the prompt designer could hide his or her social bias in the prompts. Ultimately, it makes PLM unexplainable in that we cannot assure how continuous prompts operate in PLM, which means that we lose the controllability of PLM. Without the explainability of continuous prompts, it is hard to utilize Prompt-tuning in a real-world scenario however Prompt-tuning is efficient. Still, few studies have tried to reveal the details of the relationship between Prompt-tuning and PLM, which are essential to the explainable continuous prompts.

Meanwhile, several studies have shown that continuous prompts are one of the parameter-efficient training methods in a cross-lingual setting (Zhao and Schütze, 2021; Vu et al., 2022a), which means that continuous prompts encode the task-

relevant information in the multilingual space. Since the generalizability of tasks and languages can improve the explainability of PLM, it is important to capture the features shared in each task and language. Accordingly, in this study, we investigate how continuous prompts save and utilize the information of a multilingual PLM, mGPT (Shliazhko et al., 2022), focusing on English and Korean.

This study also aims to figure out the effects of Prompt-tuning v1 and Prompt-tuning v2 on the multilingual PLM, so that we provide fundamental directions to enhance the interpretability and explainability of PLM and Prompt-tuning. Accordingly, we analyze the changes after Prompt-tuning in terms of three major structures of GPT: the attention mechanism, the activated neurons, and the label space.

Additionally, we factorize continuous prompts breaking down into task sub-prompts and language sub-prompts so that each sub-prompt can be transferred in the cross-task and cross-language settings. We believe that the transferability of continuous prompts and the effects of transfer learning can explain which knowledge of PLM continuous prompts utilize.

To this end, we address the following research questions:

1. Can we distinguish continuous prompts according to the encoded information about target tasks or target languages?
2. Can we find any explainable patterns in the changes in the attention mechanism after Prompt-tuning?
3. Can we observe any explainable patterns in the activated neurons of continuous prompts through layers?
4. Can we capture that continuous prompts interact with the label space of PLM?

Introducing previous works about prompt-based learnings and the interpretability and transferability of continuous prompts in Chapter 2, we demonstrate the architecture of Transformer and GPT in Chapter 3. In Chapter 4, we explain the methods used to train and analyze continuous prompts, and the details of downstream tasks and datasets. In Chapter 5, we present the results and analysis of the experiments, where some patterns of the changes after Prompt-tuning are discovered. Additionally, we report the results of the ablation studies.

We first present the visualizations of continuous prompts which show that deep continuous prompts are gathered according to the target tasks in the multilingual setting. Second, we find that the attention distribution changes in some layers more than in other layers regardless of tasks and languages. Also, in Prompt-tuning v2, the changes are explainable because the most changed attention layers are composed of content-dependent heads rather than position-based heads. These results suggest that continuous prompts utilize the knowledge encoded in the attention mechanism to solve the target task.

Third, we observe a special phenomenon, where the activated neurons show task-common behavior rather than task-specific behavior in the second to last layer. Subsequently, we find that the decoding token used to predict the label word gets closer to the label word than to the non-label word through the layers, which means that continuous prompts make the embedding space of PLM adapt to the target task. Also, low isotropy and the narrow gap between the two distances at the second to last layer suggest that the activated neurons have similar skills because the desired label words are actually similar in the label space. Finally, the ablation study supports these findings. To the best of our knowledge, this study is the first to probe the changes in the PLM after Prompt-tuning.

Chapter 2. Related Works

This chapter introduces several prompt-based learning methods using continuous prompts for the Transformer-based PLMs, comparing to discrete prompts. Also, Section 2.2 discusses the studies about the interpretability and transferability of continuous prompts.

2.1. Discrete Prompts vs. Continuous Prompts

Discrete prompts, which are written in human language, were first proposed to close the gap between the objective functions of pre-training and fine-tuning. Discrete prompts are usually used in two ways: In-context learning and Pattern-Exploiting Training (PET).

For In-context learning, Radford et al. (2019) showed that GPT-2 can solve the target tasks in the few- and zero-shot settings when trained via language modeling with the task instructions. They represented input, output, and task in natural language, and set the objective function $P(\text{output} \mid \text{input}, \text{task})$. If the task is to translate French to English, the input is French text and the output is English text, and the task instruction is such as "*translate French to English"*. This allows them to train the model on various task types from classification to generation in the unsupervised settings.

GPT-3 (Brown et al., 2020), which has a similar architecture to GPT-2 but a larger size, was pre-trained with the instructions and examples of each task. With In-context learning, GPT-3 showed higher performances in the un- and semi-supervised

settings than the performances of T5 (Raffel et al., 2020) in the fully-supervised settings for some tasks. These studies suggest that a large-scale PLM can solve tasks without any parameter-updates once it is pre-trained.

Schick and Schütze (2021a) suggested a Pattern-Exploiting Training (PET), where they converted an input sentence into a cloze-style phrase with the masked token for an encoder-based model such as BERT (Devlin et al., 2019). The objective function is $P(y|x)$, where x is an input including the template and y is a verbalizer which is the single token mapped to each label. For example, if a task is to predict the sentiment of a review sentence, the input is "*Best pizza ever! It was [MASK]*". The underlined is a designed template, and the model is fine-tuned to fill the masked token with a label word like "*good*" and "*delicious*". In their further study (Schick and Schütze, 2021b), they combined PET with ALBERT (Lan et al., 2020) which is a light version of BERT. Thus, they not only improved the efficiency of training but also achieved higher performances than the performances of GPT-3 in NLU tasks. Additionally, Schick and Schütze (2021c) suggested PET for an encoder-decoder based generative PLM such as T5.

Although discrete prompts are useful in the semi-supervised settings, there are some limitations on the prompt engineering. In In-context learning, we cannot know how many examples are needed or how well the prepared examples are suitable for the target task. Similarly, in PET, the model performance depends on the template design, which requires a human to manually design the templates and verbalizers. Indeed, we still need to fine-tune the PLM and save the trained PLM, which is not efficient.

To overcome these limitations, Li and Liang (2021) proposed Prefix-tuning for a lightweight training. They prepended the continuous task-specific vectors to the

input so that their own parameters are only trained. As continuous prompts do not limit their space on the embedding of PLM, they are more expressive and can affect all layers in PLM. Using GPT-2 and BART (Lewis et al., 2020), they showed that the performances on NLG tasks are comparable to the performances of fine-tuning both in the fully- and semi-supervised settings. Also, Prefix-tuning performs well on the unseen topics. However, they concluded that how Prefix-tuning improves extrapolation is an open-question.

Liu et al. (2021) suggested P-tuning which is more flexible to task types and LM types than Prefix-tuning. They added continuous prompts and trained both PLM and those continuous prompts. Unlike Prefix-tuning, they implemented continuous prompts not only before the input but also after the input and used anchor prompt tokens to improve the performances. They reported that the performances improved both in GPT and BERT via P-tuning.

Similarly, Lester et al. (2021) proposed Prompt-tuning v1, where they used only continuous prompts by freezing the parameters of PLM. This study was the first to train only continuous prompts. They showed that the performances of Prompt-tuning are comparable to the performances of fine-tuning, where the capacity of PLM has a key role. Thus, they concluded that Prompt-tuning is a parameter-efficient way to employ the knowledge LM obtained during pre-training.

In their ablation studies on NLU tasks, they showed that the longer prompt length especially more than 20 is more useful. Also, they found that the prompt initialization methods affect the performances, where the initialization with the sampled vocabulary or class label embeddings is better than the random uniform initialization. However, the largest LM was not affected by the initialization methods.

Gu et al. (2022) performed pilot experiments to investigate the efficient and

effective ways to utilize PLM with Prompt-tuning. They reported that using continuous prompts and discrete prompts together improves the performances of Prompt-tuning on sentiment analysis tasks. However, the templates in discrete prompts affect the performances significantly, which means that a human-effort is needed to select the best discrete prompts. Additionally, the verbalizer choice affects the performance as well. They found that the words explaining the meaning of the corresponding labels are generally good choices. Meanwhile, initializing the continuous prompts with the real word of the embedding of PLM was not helpful for Prompt-tuning.

More recently, Prompt-tuning v2 was proposed by Liu et al. (2022). They tried to enhance the university of Prompt-tuning across scales and tasks by injecting continuous prompts into every layer of PLM. While Prompt-tuning v1 has low performances on the hard sequence labeling tasks, Prompt-tuning v2 improves the performances of various tasks, namely extracting question answering and named entity recognition. Accordingly, they claimed that deep continuous prompts could have a more direct impact on predicting the label.

However, most studies about prompt-based learning have been conducted in English. Especially, few studies have covered Korean. Min et al. (2021) showed that Prefix-tuning is helpful for classification tasks in Korean using ETRI-BERT[①] and Korean RoBERTa (Min et al, 2019).

Kim et al. (2021) introduced HyperCLOVA which is a Korean GPT-3 with 82B parameters. They trained GPT-3 with different sizes ranging from 137M to 82B. They found that the larger the model size is, the better the performance in In-context

① https://aiopen.etri.re.kr/bertModel

few-shot learning is, except for a few tasks. Also, they have achieved the state-of-the-art in the zero- and few-shot settings. In addition, the performances in P-tuning are better than the performances in In-context learning, which was the first discovery for the large-scale PLMs. They also discussed that the impact of discrete prompts is lower in the larger model.

Furthermore, Shin et al. (2022) investigated the effects of pre-training corpora on In-context learning for HyperCLOVA. They reported that the corpus sources have a large impact on the performance of In-context learning. Especially, the corpus was useful when its domain was relevant to the domain of the downstream task.

To summarize, the differences between discrete prompts and continuous prompts are largely three points. First, while discrete prompts are human-interpretable tokens, continuous prompts are un-interpretable pseudo tokens. Second, discrete prompts require the intervention of humans to design the templates but continuous prompts require much less human-effort. Third, continuous prompts are more parameter-efficient in that they can be trained by prepending to the frozen PLM.

Thus, we adopt Prompt-tuning v1 and Prompt-tuning v2 as training methods using continuous prompts. This is because they do not need to update the parameters of PLM and the intervention of humans in prompt design can be minimized. In other words, we can see how continuous prompts drive PLM clearly, minimizing the extra interruption that might affect training.

2.2. The Interpretability and Transferability of Continuous Prompts

Lester et al. (2021) tried to interpret continuous prompts by measuring the similarities between the embeddings of learned continuous prompts and the vocabulary of PLM. They observed that continuous prompts have 'word-like' representations, which are relevant to the domain of the target task. An example of which is the BoolQ dataset (Clark et al., 2019) of the nature/science category, where the continuous prompts are close to the words such as '*science*', '*technology*', and '*engineering*' in the embedding space.

To interpret continuous prompts in human language, Khashabi et al. (2022) investigated the Prompt Waywardness hypothesis. In this hypothesis, there exists a continuous prompt that can solve the target task while becoming close to the arbitrary discrete prompt, which is not relevant to the target task. They observed that continuous prompts satisfying the Prompt Waywardness hypothesis do exist. They also provided some explanations. First, continuous prompts cannot be projected to exactly one embedding of discrete prompts. Second, when continuous prompts are injected only into the first layer, the deeper layers have more expressivity, where the effects of Waywardness get stronger. Finally, they discussed that it is hard to discover the human-interpretable continuous prompts, which leads to the side effects in the real-world scenario, such as the concealed adversarial attacks.

Meanwhile, transfer learning using Prompt-tuning is actively researched since it is one of the effective ways to utilize large-scale PLMs. Besides, the transferability of continuous prompts between tasks and languages is also necessary to discover the

way continuous prompts encode task-relevant information.

Zhao and Schütze (2021) compared three prompt-based methods; discrete prompts, continuous prompts, and discrete prompts + continuous prompts (mixed prompts), for cross-lingual few-shot learning on NLI tasks. They designed discrete prompts and verbalizers in English first and then translated them into target languages using Google translation. For mixed prompts, they added continuous prompts in the templates. In few-shot settings in English, continuous prompts have better performances than discrete prompts for most shots and mixed prompts do not have significant improvement. In cross-lingual few-shot settings, discrete prompts have rather better performances than continuous prompts. They claimed that the code-switched templates are helpful for the cross-lingual ability of PLM. Mixed prompts are useful in specific shots (64, 128).

Vu et al. (2022b) conducted experiments on transfer learning via Prompt-tuning. They used 16 source tasks and 10 target tasks, where continuous prompts of the target tasks are initialized with the trained source continuous prompts. The source prompts improved the target tasks, and especially, the source tasks including high-level reasoning skills were useful. Also, they got the task similarity by calculating the cosine similarity between continuous prompts to predict the better source task for the target task. They found that the transferability of continuous prompts is correlated to the task similarity. Notably, the transferability was sensitive to the task type more than the domain of the dataset.

Su et al. (2022) examined the transferability of continuous prompts in the zero-shot setting. They observed that the source prompts can solve the same type of target tasks. However, the performance was low when the target task requires various linguistic skills. Also, they calculated the task similarity called ON score by using

the activation states of continuous prompts, which is a more suitable indicator of the response of PLM. They reported that the performances of transfer learning are more correlated to the ON score than to the task similarity using the cosine similarity.

Vu et al. (2022a) factorized continuous prompts decomposed into task prompts and language prompts to improve the zero-shot cross-lingual transferability of continuous prompts in summarization tasks. They trained language prompts and task prompts via unsupervised learning for each language. After training task prompts on English summarization tasks with English language prompts, they replaced the language prompts with the language prompts trained in the target language, and directly evaluated the performance in the target language. They reported that the factorized prompts prevent the catastrophic forgetting problem of the PLM.

To sum up, the explainability of continuous prompts is still an open question. Besides the interpretability of continuous prompts, the changes that continuous prompts cause to PLM are also crucial to enhance the explainability of continuous prompts. Also, to deal with the second reason Khashabi et al. (2022) mentioned, we need to investigate Prompt-tuning v2 where all layers are controlled by continuous prompts.

Thus, our study focuses on the effects of Prompt-tuning v1 and Prompt-tuning v2 on PLM through layers, so that we can figure out the operation of continuous prompts in an explainable way. While many studies have investigated the influence of fine-tuning in PLM, few studies have investigated the influence of Prompt-tuning. Again, the transferability of continuous prompts is one of the indicators to investigate how PLM reacts according to tasks and languages. Accordingly, this study aims to provide the basis for the future studies of the interpretability and explainability of continuous prompts through transfer learning.

Chapter 3. Transformer Architecture

3.1. Transformer

Vaswani et al. (2017) proposed a Transformer whose main part of modeling is the attention mechanism (Figure 1). Transformer has an encoder-decoder structure, where an input sentence is converted into a continuous vector in the encoder and an output sentence is generated in the decoder when given the continuous vectors of a previous step. The encoder and decoder are stacked in N layers, and Vaswani et al. (2017) stacked 6 layers.

The attention mechanism computes the attention score for each query token (Q) using key (K) and value (V) vectors. The softmax function is applied to compute the weights from Q and K, and then we get the final attention result by the dot products of the obtained matrix with V (Equation 1). In the multi-head attention, we concatenate all heads and get the output vector Z by multiplying them by the weight matrix W^O, so that the model can attend to other tokens in different representation subspaces at different positions (Equation 2).

$$\begin{gathered}\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V\,, \\ \text{where } Q = \text{a Query matrix}, K = \text{a Key matrix}, \\ V = \text{a Value matrix, and } d_k = \text{the dimension of a Key}\end{gathered} \tag{1}$$

$$\begin{gathered}\text{MultiHead}(Q,K,V) = \text{Concat}(head_1,\dots,head_h)W^O, \\ \text{where } head_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)\end{gathered} \tag{2}$$

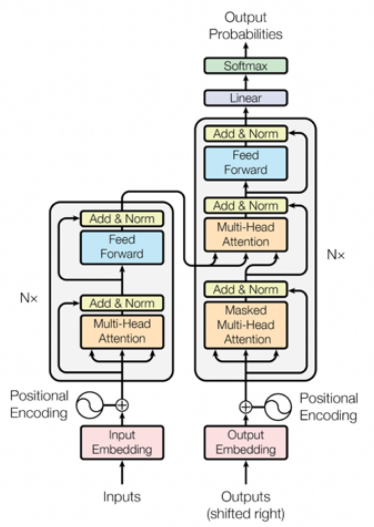

Figure 1 The Transformer architecture from Vaswani et al. (2017).

The encoder has two sub-layers, a multi-head self-attention layer, and a feed-forward neural network layer. In the self-attention mechanism, Q, K, and V are the vectors of all words from the previous layer so that we can relate all positions in the input sequence to each other while considering the context. Then, the output vector Z is fed into the position-wise feed-forward neural network (FFN) layer. The FFN layer is composed of two linear transformations connected with a ReLU activation, where the parameters are updated (Equation 3).

$$\mathrm{FFN}(x) = max(0, xW_1 + b_1)W_2 + b_2 \tag{3}$$

In addition to the two sub-layers of the encoder, the decoder has another multi-head attention layer to process the output of the encoder. The masked multi-head self-attention layer of the decoder allows the model to know only the information from the previous tokens by masking the next tokens. When the outputs of the

encoder K and V in each step are given, the decoder calculates the probability of the next token among the vocabulary of the model in the final linear layer. Then the predicted token is fed into the encoder as a new query, and the model keeps generating the tokens until the end token.

To let the model know the order of the sequence, the sinusoidal positional encodings are added to the input embeddings at the bottom of the encoder and decoder (Equation 4).

$$\begin{aligned} \mathrm{PE}_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{model}}\right) \\ \mathrm{PE}_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{model}}\right) \end{aligned} \quad (4)$$

, where pos is the position and i is the dimension of the embedding.

3.2. GPTs

As Transformer has achieved high performances on the machine translation tasks, various models adopting the Transformer mechanism are proposed to solve different NLU tasks. GPT is also a Transformer-based model, which uses only the decoder part of Transformer. GPT (Radford et al., 2018) is pre-trained via a standard language modeling whose objective function is:

$$\mathrm{L}(U) = \sum_i \log \mathrm{P}(u_i | u_{i-k}, \ldots, u_{i-1} ; \theta) \quad (5)$$

where $U = \{u_1, \ldots, u_n\}$ is a sentence and u_i is a token and k is the size of the context window, and θ is the trainable parameters of the model. The conditional probability P is modeled by the multi-layer Transformer decoders. They demonstrate that GPT can be fine-tuned on any type of tasks by predicting a label

using start tokens or end tokens.

Radford et al. (2019) proposed GPT-2 whose architecture is similar to GPT, but the scale is larger, 1.5B parameters (Figure 2). Additionally, the layer normalization was moved from the next to the attention block to the input of each block, and an additional layer normalization was added after the final attention block by removing the layer normalization after the FFN layer. Notably, they used language modeling as a fine-tuning strategy so that GPT-2 can overcome the inefficiencies of other LMs such as BERT.

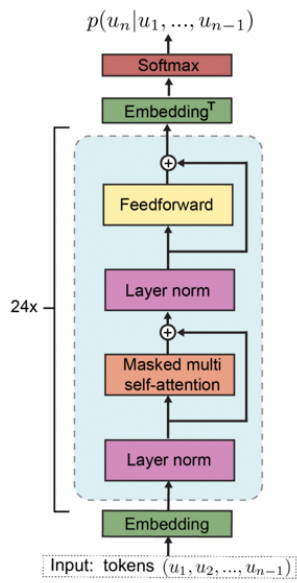

Figure 2 The GPT-2 architecture with 24 layers from Heilbron et al. (2019).

Finally, GPT-3 (Brown et al., 2020) with 175B parameters, which are 10x larger than the parameters of GPT-2, shows higher performances on various NLU tasks in few- and zero-shot learning. In addition to GPT-2, they used alternating dense and locally banded sparse attention patterns in the layers. Using In-context learning, they suggested that the large autoregressive PLM can be helpful for all tasks including

translation, question-answering, and classification without any parameter updates.

Although GPTs are efficient in the semi- and un-supervised settings, they show poor performances when trained via traditional fine-tuning strategy. To overcome such limitations of GPT, Liu et al. (2021) suggested P-tuning that leads to both higher efficiency and higher performance.

Since GPT is generative and flexible to task types, we believe that GPT has great potential, especially in the multi-lingual and multi-task environment. Also, Prompt-tuning aims to improve the parameter-efficiency which is the advantage of GPT. For these reasons, this study uses the multilingual version of GPT-3 with 1.3B parameters (Shliazhko et al., 2022).

Chapter 4. Methodology

In this chapter, we introduce the details about Prompt-tuning v1 and Prompt-tuning v2 in Section 4.1. From Section 4.2 to 4.4, we discuss the main analysis methods including the attention mechanism, the activated neurons, and the label space. Lastly, Section 4.5 introduces 6 tasks used for the experiments including datasets in English and Korean.

4.1. Prompt-tuning

4.1.1. Prompt-tuning v1 and Prompt-tuning v2

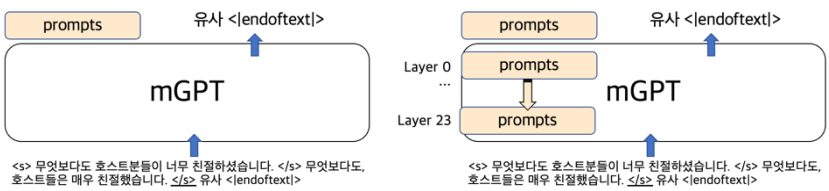

Figure 3 Prompt-tuning v1 (left) and Prompt-tuning v2 (right) for KLUE-STS. Only prompts are trainable.

Since GPT is an autoregressive model, we make the model generate the labels after the separator token </s> until the end token <|*endoftext*|> and train continuous prompts by computing the conditional probability for the label words. ② Also, we freeze mGPT and update only the parameters of the prepended prompts.

② The special tokens are the tokens used when pre-training mGPT in Shliazhko et al. (2022).

For a single type task, we feed the model the input $\{p_1, \ldots, p_k, <\text{s}>, x_1, \ldots, x_n, </\text{s}>, w_{gold}, <|\text{endoftext}|>\}$, where $P = \{p_1, \ldots, p_k\}$ is the continuous prompts with the length k, the input sentence X is $\{x_1, \ldots, x_n\}$ with the length n, and w_{gold} is the label words. Then, following Lester et al. (2021), we train the continuous prompts maximizing the probability:

$$\Pr_{\theta;\theta_P}(w_{gold}, <|\text{endoftext}|> | p_1, \ldots, p_k; <\text{s}>, x_1, \ldots, x_n, </\text{s}>) \quad (6)$$

where θ is the parameters of the model and θ_P is the parameters of the prompt embeddings. Similarly, for a pair type task, the model maximizes the probability to generate the label words after the second separator token:

$$\Pr_{\theta;\theta_P}(w_{gold}, <|\text{endoftext}|> | p_1, \ldots, p_k; <\text{s}>, x_1^1, \ldots, x_n^1, </\text{s}>, x_1^2, \ldots, x_m^2, </\text{s}>) \quad (7)$$

In this way, we can use the unified format regardless of the task type, which means that continuous prompts can be transferred between any tasks.

For Prompt-tuning v2, we prepend continuous prompts for all layers using the *past_key_value* element used for the attention mechanism. The difference with the original Prompt-tuning v2 (Liu et al., 2022) is that we inject *key* and *value* as the same one so that we get only one prompt embedding per layer which has the exact same dimension as the model. Otherwise, continuous prompts are composed of two separate embeddings. Also, we use the LM version of Prompt-tuning v2 to compare with Prompt-tuning v1 systematically.

For both Prompt-tuning v1 and Prompt-tuning v2, we randomly initialize the continuous prompt ranging from -0.5 to 0.5 since different initializing for each task,

such as using the embedding of label words, leads to fluctuating results. Also, we use the prompt length $k = 20$. As previous works suggest, the position of prompts and the length of prompts affect the performances inconsistently. In this study, we use continuous prompts in the most basic setting because we focus on the generalizability of tasks and languages on how continuous prompts operate.

4.1.2. Sub-prompts Transfer Learning

Additionally, this study investigates the transferability of Prompt-tuning by splitting continuous prompts into language prompts and task prompts. Motivated by Vu et al. (2022a), we get language prompts by post-training mGPT on the multilingual Oscar-mini dataset [③] in each language while freezing randomly initialized task prompts. We expect that continuous prompts in the multilingual model adapt to specific language during post-training by maximizing the probability:

$$\Pr_{\theta;\theta_{p^l};\theta_{p^t}}(x_i|p_1^l,\ldots,p_{10}^l\,;p_1^t,\ldots,p_{10}^t\,;x_{i-k},\ldots,x_{i-1}) \tag{8}$$

where θ_{p^l} is the only trainable parameters. We set language prompts (P^{l}) and task prompts (P^{t}) with length $k = 10$ for all tasks, initializing separately.

After training the language prompts, we trained the task prompts for each task, freezing the corresponding language prompts (Equation 9). In this way, we factorize the language sub-prompts and task sub-prompts which are available for any combination of language and task.

[③] https://huggingface.co/datasets/nthngdy/oscar-mini. Oscar dataset (Ortiz Suarez et al., 2019) is an **O**pen **S**uper-large **C**rawled **A**lmanach Co**r**pus that is based on the Common Crawl corpus(https://commoncrawl.org/).

$$\Pr_{\theta;\theta_{pl};\theta_{pt}}(t_{gold}, < |\text{endoftext}| > | \, p_1^l, \ldots, p_{10}^l \, ; p_1^t, \ldots, p_{10}^t \, ; \\ < \text{s} >, x_1, \ldots, x_n, </\text{s} >) \tag{9}$$

Preparing language prompts and task prompts, we conduct transfer learning in the cross-task and cross-language settings (Figure 4). For task transfer, the target task sub-prompts are replaced with the source task sub-prompts. While the parameters of the source language sub-prompts are fixed, the replaced task sub-prompts are trained.

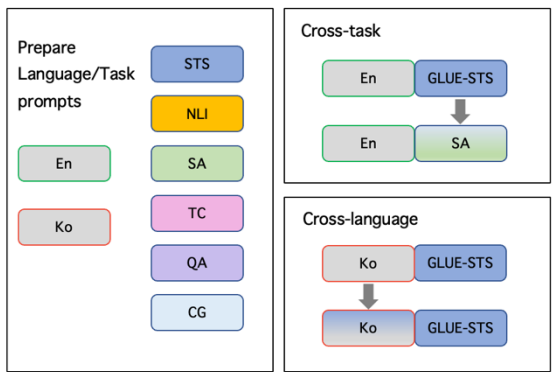

Figure 4 Transfer learning with sub-prompts. 'Cross-task' is an example for 'GLUE-STS' to 'SST2' and 'Cross-language' is an example for English to Korean ('GLUE-STS' to 'KLUE-STS').

For language transfer, we set the source task and target task as the same task in other languages. That is, when the source language is English and the target language is Korean, we transfer GLUE-STS' to KLUE-STS. Thus, the target language sub-prompts are trained while freezing the trained task sub-prompts on the source language. By training the language sub-prompts, we believe that the model learns the target language from the knowledge of the source language encoded in frozen target task sub-prompts.

Finally, using principal component analysis (PCA), we visualize continuous

prompts of each target task to see how continuous prompts are clustered. In the case of deep continuous prompts, we analyze continuous prompts for each layer.

4.2. The Attention Mechanism

In GPT, the attention mechanism works in the left-to-right direction, so most tokens give the maximum attention to their previous token. For this reason, we try to explain the attention of mGPT meaningfully.

Vig and Belinkov (2019) consider that such a tendency of the attention distribution in GPT is based on not the content but the position. To measure how attention varies over different input sequences, they suggest attention variability, which adopts the basics of the mean absolute deviation:

$$\text{Variability}_{\alpha} = \frac{\sum_{x \in X} \sum_{i=1}^{|x|} \sum_{j=1}^{i} |\alpha_{i,j}(x) - \bar{\alpha}_{i,j}|}{2 \cdot \sum_{x \in X} \sum_{i=1}^{|x|} \sum_{j=1}^{i} \alpha_{i,j}(x)} \quad (10)$$

where $\alpha_{i,j}(x)$ is the attention score x_i gives to x_j, and $\bar{\alpha}_{i,j}$ is the mean of $\alpha_{i,j}(x)$ overall sentences x in dataset X. Meanwhile, the first token of each input sequence tends to receive the maximum attention score. Thus, the first token is excluded to calculate the variability. Following Vig and Belinkov (2019), we compute the variability with the first N tokens ($N = 10$) excluding the first token for each dataset.④

④ For commonsense generation task, we use first 2 tokens without the first token because all input sequences include only 3 words.

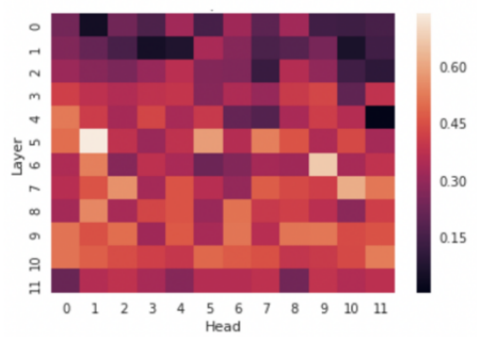

Figure 5 The heatmap of the attention variability from Vig and Belinkov (2019).

The low variability means that the attention score is given focused on the specific position over different input sequences. That is, the head with higher variability is a content-dependent head. Figure 5 shows that the attention heads in the initial layers are likely to focus on the position rather than the content in GPT-2 small.

Since we do not update the parameters of mGPT, we believe that the changes in the attention mechanism after Prompt-tuning show the effects of Prompt-tuning on mGPT. Thus, we investigate the difference in the attention distribution between the pre-trained mGPT and the prompt-tuned mGPT. We use Kullback-Leibler divergence which is commonly used to measure the difference between two probability distributions:

$$\mathrm{KLdiv}(\mathrm{PromptPLM}(X), \mathrm{PLM}(Y)) = \sum_{x \in X, y \in Y} x \log \frac{x}{y} \tag{11}$$

where X and Y are the attention weights of each head for the same input sequence, from the prompt-tuned PLM and the pre-trained PLM respectively.

To get the KL divergence per head, we post-process the attention distributions.

We first sum the attention scores of each token, and then remove the attention score of the special tokens <*s*> and </*s*> in each input. In the case of PromptPLM(X), we exclude the attention scores of prompt tokens before post-processing. Finally, we replace x and y with 1e-20 when they are 0 after passing the softmax to avoid infinity when computing KL divergence.

Lastly, we get the correlation between the attention variability and the attention changes measured with KL divergence through layers to see whether we can interpret the patterns of the changes in terms of the knowledge mGPT learned during pre-training. We hypothesize that the more changed layers are the layers with the content-dependent heads since continuous prompts make use of the information actively.

4.3. The Activated Neurons

The hidden states from the self-attention layer are fed into the FFN layers. Dai et al. (2022) regard the FFN layer as the emulation of the self-attention layers. According to them, the simple equations of each layer are:

$$\text{Self} - \text{Att}_\text{h}(X) = softmax(Q_h K_h^T) V_h \quad (14)$$

$$\text{FFN}(H) = gelu(HW_1)W_2 \quad (15)$$

where h is an attention head and H is the concatenation of the results of all attention heads. Given an input vector H as a query vector, W_1 and W_2 correspond to keys and values. They proposed a new method to detect knowledge neurons, which represent the relational fact, via the fill-in-the-blank cloze task. When the hidden states pass through the first linear layer, the knowledge neurons are

activated, and the second layer combines the corresponding memory (Figure 6). They found that the activation of the knowledge neurons is positively correlated to the knowledge expression in the prompt.

Meanwhile, the FFN layers in GPT are different from other transformer-based models, in that the linear layer is substituted by the 1-dimensional convolutional layer with GELU activation function (Figure 7). Thus, in GPT, *c_fc* layer and *c_proj* layer correspond to W_1 and W_2 in Equation 15.

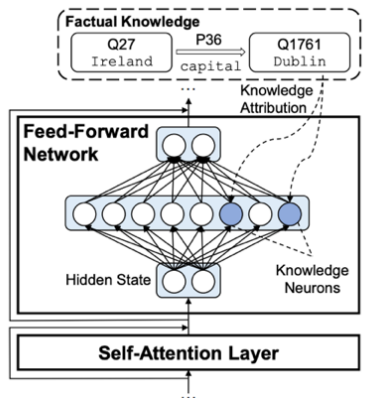

Figure 6 The knowledge neurons from Dai et al. (2022).

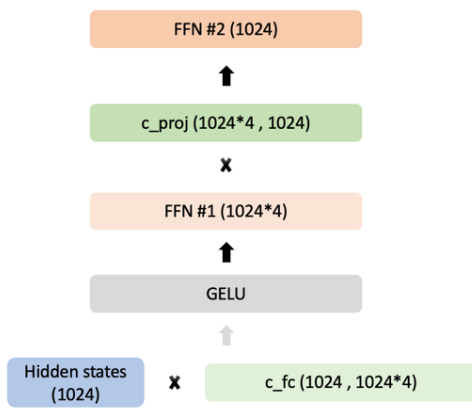

Figure 7 The simplified FFN layer structure in mGPT. The number in the bracket is the dimension.

As Prompt-tuning fills the gap between pre-training and fine-tuning, Prompt-tuning is a good way to observe the knowledge that the neurons encode in PLM. Su et al. (2022) proposed an ON score, where we can measure the response of the prompt-tuned model. Following them, this study computes the ON score using the decoding token per layer (Equation 17). Given an input sequence $\{P, </\mathrm{s}>\}$, we regard the output of the *c_proj* layer in GPT as the activation state of the continuous prompts. We compute the ON score between all possible combinations of tasks in both languages and get the average of them to see the trend line across layers.

$$\mathrm{ON}(P_l^{t_1}, P_l^{t_2}) = \frac{\mathrm{AS}(P_l^{t_1}) \cdot \mathrm{AS}(P_l^{t_2})}{||\mathrm{AS}(P_l^{t_1})||\,||\mathrm{AS}(P_l^{t_2})||}$$
$$, \text{where } \mathrm{AS}(P_l^{t_k}) \text{ is the activation state of decoding token } </\mathrm{s}> \text{ on task } t_k \text{ in layer } l \tag{17}$$

Additionally, we compute the ON score between source prompts and target prompts to compare the scores in two conditions: without and with transfer learning. For ‘without’ transfer learning, target prompts are the prompts including language sub-prompts and task sub-prompts trained on the target task from scratch. For ‘with’ transfer learning, target prompts are the prompts including language sub-prompts on the target language but task sub-prompts trained on the target task initialized from the task sub-prompts of the source task.

4.4. The Label Space

The final outputs of the l-th layer represent the embedding vectors including the context and task knowledge. Especially, we use the representations from the last

layer to decode the verbalizer, which is one of the key elements in Prompt-tuning. However, there is a problem that we cannot guarantee the best verbalizer because there always exists better choices and the label space[⑤] is not limited to the verbalizers. Recently, several studies endeavor to find the optimal label words in an automatic way (Schick et al., 2020; Gao et al., 2021). Despite these efforts to improve the verbalizer engineering problem, few studies figure out the underlying reason for the sensitivity to the verbalizer.

In this study, we try to explore the interactions between continuous prompts and the label space of PLM. First, we measure how well the embedding space represents semantics by computing the isotropy of the pre-trained mGPT and the prompt-tuned mGPT. We get the isotropy of each dataset using the cosine similarity between the N pairs of the randomly sampled words, following Rajaee et al. (2022) (Equation 18). For each input, after extracting the representations in all layers, we randomly choose a representation of one token. With the selected representations, we paired two representations among them randomly to calculate the cosine similarity.

Second, we compare the isotropy for the prompt-tuned PLM and the pre-trained PLM to investigate the effects of Prompt-tuning. If the isotropy of the embedding space where continuous prompts are implemented gets higher, Prompt-tuning drives PLM to improve semantic expressiveness (Rajaee et al., 2022).

$$\text{Isotropy}^{m}_{\text{cosine}}(W_m) = \frac{1}{N}\sum_{i=1, x_i \neq y_i}^{N} \text{cosine similarity}(x_i, y_i)$$
$$\text{, where } x_i \in X, y_i \in Y,$$
$$X \text{ and } Y \text{ are the sets of randomly sampled embeddings,}$$
$$\text{and } W_m \text{ is the embedding matrix of the model } m. \quad (18)$$

[⑤] In this study, we call the space of the verbalizers the label space.

Since we exclude the special tokens when computing the isotropy, we get the cosine similarity between the decoding token and the randomly sampled word (Equation 19). This allows us to see whether the decoding token has different representations from other tokens.

$$\begin{aligned}&\text{Isotropy}_{\text{cosine}}^{decoding\ token}(W_m)\\&= \frac{1}{N}\sum_{i=1,</s>\neq y_i}^{N} \text{cosine similarity}(</\text{s}>, y_i)\end{aligned} \tag{19}$$

Intuitively, in the embedding space, the decoding token would be closer to the corresponding label word than to the non-label word in each example. Thus, we get the cosine similarity between the decoding token and the label words across layers and compare it to the distance between the decoding token and the non-label words:

$$\text{dist}_{\text{label}} = \text{cosine similarity}\left(</\text{s}>, w_{gold}\right) \tag{20}$$

$$\text{dist}_{\text{non-label}} = \text{cosine similarity}\left(</\text{s}>, w_{non-gold}\right) \tag{21}$$

where the label candidate words = $\{w_{gold}, w_{non\_gold}\}$ for the binary classification, and $\{w_{gold}, w_{non\_gold\_1}, \ldots, w_{non\_gold\_k-1}\}$ for the multi classification with k labels. In the case of the multi-label classification tasks, we randomly select a w_{non_gold} in $\{w_{non\_gold\_1}, \ldots, w_{non\_gold\_k-1}\}$. Also, we use the first token of the label candidate words when split into sub-tokens.

4.5. Tasks

We conduct experiments on 6 tasks: Semantic Textual Similarity (STS), Natural

Language Inference (NLI), Sentiment Analysis (SA), Topic Classification (TC), Question and Answering (QA), and Commonsense Generation (CG).

STS. Semantic Textual Similarity is a task to measure the similarity score between two sentences from 1 to 5. The higher the score is, the higher the similarity is. In this study, we binarize the score into similar and dissimilar class. If the score is more than 3, it is labeled as similar and if the score is less than 3, it is labeled as dissimilar.

NLI. Natural Language Inference is a task to decide the semantic relationship of two sentences, a premise and a hypothesis, among three categories: entailment, contradiction, neutral. Given a premise first, the model should determine if a hypothesis is true or false or undetermined.

SA. Sentiment Analysis is a task to determine the sentiment expressed in a sentence. Usually, the sentence is classified into two labels: positive and negative. The datasets for SA include user reviews from various industrial domains such as movie, restaurant, and product, or comments and texts from Social Network Service and News articles. In this study, we use the movie review dataset which is one of the popular domains.

TC. Topic Classification dataset includes sentences or paragraphs from different themes. We use news topic classification task, where the topic such as IT/science and Social is annotated for each headline. Also, the topics vary for each language.

QA. For Question and Answering for reading comprehension, the model should extract the answer to a question from the given context. The answer can be a word or spans consisting of more than two words.

CG. Commonsense Generation is a task proposed to assess the model's ability

to generative commonsense reasoning. Using a set of three common concepts including an object (noun) and action (verb), the model generates a full grammatical sentence which has a coherence with an everyday scenario.

4.5.1. Evaluation Metrics

The evaluation metric depends on task types. First, for a single-label classification task SA, we use accuracy score. Especially, for STS, we report F1 score since we binarize the scores. Second, for multi-label classification tasks (NLI, TC), we use macro-F1 score.

Given a confusion matrix (Table 1), we calculate accuracy (Equation 22) and F1 scores (Equation 25 & Equation 26), where $F1_c$ is a F1 score for each class c.

		Actual Class	
		True	False
Predicted Class	True	True Positive (TP)	False Positive (FP)
	False	False Negative (FN)	True Negative (TN)

Table 1 A confusion matrix.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{22}$$

$$Precision = \frac{TP}{TP + FP} \tag{23}$$

$$Recall = \frac{TP}{TP + FN} \tag{24}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{25}$$

$$macro\ F1 = \frac{1}{|C|}\sum_{c \in C} F1_c \tag{26}$$

Third, for QA, which is a span extraction task, Exact Match (EM) and F1 score are used. If the model predicts the exact same answer span as the gold answer, EM is 1, otherwise, EM is 0. We get the total EM score by averaging the EM scores for each sample. For F1 score, individual words in the gold answer are regarded as the gold class.

Lastly, the performance of CG can be assessed by several metrics for generation tasks, such as BLEU, ROUGE, and METEOR. However, according to Lin et al. (2020), this study uses Coverage score, which is more suitable for the captioning tasks by which CG is motivated. Coverage score is the average percentage of the input concepts included in the output concepts that are lemmatized. For lemmatization, we use NLTK module[⑥] for English and Mecab module[⑦] for Korean.

4.5.2. English Dataset

GLUE-STS is a dataset of GLUE benchmark (Wang et al., 2018) for STS task. GLUE-STS consists of sentence pairs from news headlines, video and image captions, and natural language inference data. The similarity score is annotated by human ranging 1 to 5.

SNLI is a Stanford NLI corpus (Bowman et al., 2015), where the premises were

⑥ https://www.nltk.org/
⑦ https://bitbucket.org/eunjeon/mecab-ko-dic/src/master/

crawled from caption corpus Flickr30k (Young et al., 2014) and the hypotheses were written by 2,500 workers. Flickr30k contains images collected from Flickr, and 5 reference sentences were captioned by human.

SST2 (Socher et al., 2013) is the Stanford Sentiment Treebank dataset for SA including fine-grained movie reviews from rottentomato[⑧] which are introduced in Pang and Lee (2005). Each sentence was processed by the Stanford parser and annotated by human.

AGnews is constructed by Zhang et al. (2015) using AG's news article corpus[⑨]. Each headline is classified into 4 categories: World, Sports, Business, and Science/Technic.

SQuAD 2.0 is a reading comprehension question answering dataset (Rajpurkar et al., 2018) consisting of Wikipedia articles. SQuAD 2.0 is a new version of SQuAD 1.0 (Rajpurkar et al., 2016), in that they add unanswerable questions about the same paragraphs. Crowd workers were asked to craft context relevant questions to build high-quality dataset.

CommonGen is constructed to examine the ability to generate sentences with commonsense reasoning (Lin et al., 2020). They first collect concept sets from visually-grounded sentences extracted from image captioning datasets and video captioning datasets. Then crowd workers wrote the sentences using sampled frequent concept-sets. They evaluated the generated sentences using NLTK tokenizer.

⑧ https://www.rottentomatoes.com/

⑨ http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html

4.5.3. Korean Dataset

KLUE-STS is a dataset of KLUE benchmark (Park et al., 2021) for STS task. KLUE-STS contains sentences from various domains and topics, such as airbnb review corpus, policy news data, and ParaKQC data[⑩]. Unlike GLUE-STS dataset, the similarity score is annotated by human ranging 0 to 5.

KorNLI is a NLI dataset constructed by Ham et al. 2020. They translated the English NLI datasets: SNLI (Bowman et al., 2015), MNLI (Williams et al., 2018), and XNLI (Conneau et al., 2018), and the experts post-edited the translated sentences. The translation sentences of SNLI and MNLI compose training dataset and the translation of XNLI validation and test dataset.

NSMC[⑪] is a naver sentiment movie corpus crawled from naver movie review website[⑫], which is commonly used to NLU tasks in Korean. Reviews whose rating score higher than 9 are labeled as positive reviews, and reviews whose rating score from 1 to 4 are labeled as negative reviews.

KLUE-Ynat is an Yonhap News Agency dataset for Topic classification from KLUE benchmark. News headlines are classified into 7 categories: Politics, Economy, Society, Culture, World, IT/Science, Sports.

KorQuAD 1.0 (Lim et al., 2019) is a reading comprehension question answering dataset constructed from Wikipedia article corpus. KorQuAD 1.0 was constructed in the same way with SQuAD 1.0. In addition, they introduce a Syllable-level F1 score which is more suitable to Korean.

---

⑩ ParaKQC is an utterance dataset collected from user utterance at smart home devices.
⑪ https://github.com/e9t/nsmc
⑫ https://movie.naver.com/movie/point/af/list.naver

KommonGen is a Korean version dataset of CommonGen. Seo et al. (2021) used image captioning dataset from AI-HUB⑬ which is a machine translated dataset from MS COCO dataset Using Mecab tokenizer, they evaluated the model with BLEU, Meteor, Rouge, and Coverage score suggested in Lin et al. (2020).

⑬ https://aihub.or.kr/aihubdata/data/view.do?currMenu=120&topMenu=100&aihubDataSe=extrldata&dataSetSn=261

Chapter 5. Experiments

This chapter discusses the results of the experiments. Section 5.1 reports the performance results briefly, and then Section 5.2 presents the PCA of continuous prompts. The following three chapters cover the analysis of the attention mechanism, the activated neurons, and the label space. Lastly, Section 5.6 reports the results of the ablation studies.

5.1. Performance Results

5.1.1. Prompt-tuning v1 and Prompt-tuning v2

	English		Korean	
Task	V1	V2	V1	V2
STS	83.2	**84.23**	38.43	**71.59**
NLI	80.95	**86.46**	45.67	**62.8**
SA	87.15	**88.18**	84.94	**87.18**
TC	85.8	**87.27**	81.18	**84.27**
QA	61.98/47.25	**67.21/51.91**	65.41/59.62	**72.32/66.50**
CG	78.4	**82.97**	87.94	**91.33**

Table 2 The Performance results of Prompt-tuning v1 and Prompt-tuning v2.

We used two A100 GPUs with 80G memory for Prompt-tuning. The verbalizers for each task are presented in Table 5 in Appendix and the hyperparameters are presented in Table 6 in Appendix. The number of the trainable parameters for Prompt-tuning v1 is 40960 and the one for Prompt-tuning v2 is 983040.

Table 2 shows that every task improves in Prompt-tuning v2. Indeed, the larger parameters and the higher controllability in the deeper layers would lead to higher performances. On average, the scores improve by 2.85 for English and 10.57 for Korean. Especially, while the scores of most tasks rise by under 10, the scores of KLUE-STS and KorNLI rise by around 35 and 17, respectively. Since mGPT is trained on large English resources but low Korean resources, the number of tokens in Korean is much lower than the one in English. Accordingly, we surmise that deep continuous prompts are useful for low-resource language in the multilingual model.

5.1.2. Sub-Prompts Transfer Learning

For transfer learning, we used two A100 GPUs mentioned above and three Tesla V100 GPUs supported by the National IT Industry Promotion Agency (NIPA) as well. We present the performance results of sub-prompts transfer using heatmaps so that we can glimpse the transferability between the source task and the target task. The full performances are reported in Tables from 7 to 12 in Appendix.

Figure 8 shows the results of task sub-prompts transfer. We measure the transferability by subtracting the standard target score from the transferred target score. The standard target score is a score without transfer learning, using corresponding task sub-prompts. Thus, the positive values indicate the improved scores and the negative values indicate the degraded scores.

In terms of the target tasks (columns), while most performances improve in Prompt-tuning v2, some target task performances are degraded in Prompt-tuning v1. Particularly, the performances of NLI (SNLI, KorNLI), CG (CommonGen, and

KommonGen) are reduced significantly. Similarly, in terms of the source tasks (rows), most source tasks are useful for the target tasks in Prompt-tuning v2, where the target performances always improve when the source tasks are NLI, QA (SQuAD, KorQuAD), and CG both in English and Korean. These findings confirm the results from the previous work, where Vu et al. (2022b) found that the source tasks including high-level reasoning skills are useful.

Figure 8 The performance results of task sub-prompts transfer. The row is a source task and the column is a target task.

In Figure 9, the results of language sub-prompts transfer show similar patterns as the results of task sub-prompts transfer. The scores in the cells are calculated in

the same way as the scores in Figure 8. CG, QA, and NLI are useful source tasks both in English and Korean. Notably, the score of Ynat transferred from AGNews (-9.12) is degraded a lot in Prompt-tuning v2. Thus, we conclude that the source tasks including general linguistic knowledge are more beneficial than the source tasks with specific domain and specific knowledge.

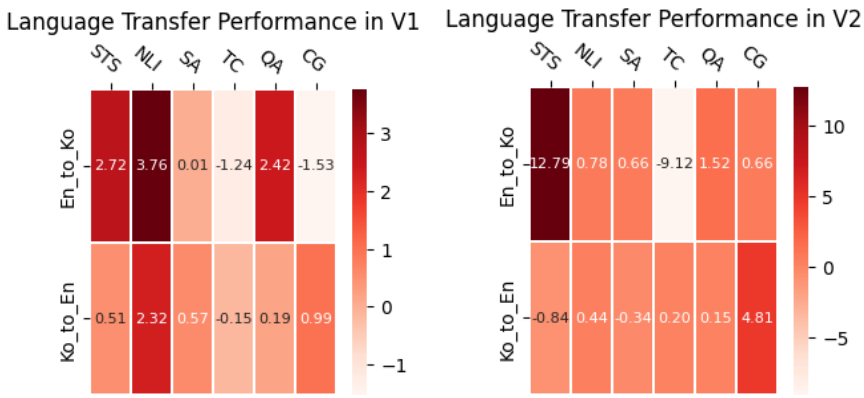

Figure 9 The performance results of language sub-prompts transfer. The row is a 'source language to target language' and the column is a task.

5.2. Analysis

5.2.1. The Visualization of Continuous prompts

Deep continuous prompts encode task-relevant information clustering by language.

We present the visualizations of continuous prompts into 2-dimension using PCA to investigate how they encode the knowledge according to task and language.

While we cannot find any meaningful clusters in Prompt-tuning v1 (Figure 10), continuous prompts are grouped in Prompt-tuning v2 through layers (Figure 11). Each point in the figures corresponds to one continuous prompt token of the corresponding task, where each task has 20 points according to the prompt length.

After passing the first layer (layer 0), continuous prompts tend to be gathered according to the target task. Although continuous prompts are dispersed again in layer 5, they keep forming clusters according to the target task. The task clusters in different languages are grouped, especially those with high cohesion in the middle layers (7~17). Additionally, starting from layer 14, we observe the separation of some clusters (NLI, STS, TC) that are not well separated in the previous layers (See Figure 31 in Appendix).

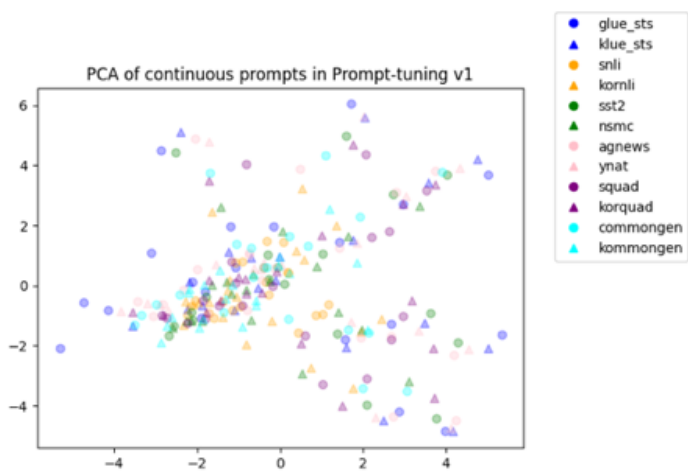

Figure 10 The PCA of continuous prompts in Prompt-tuning v1.

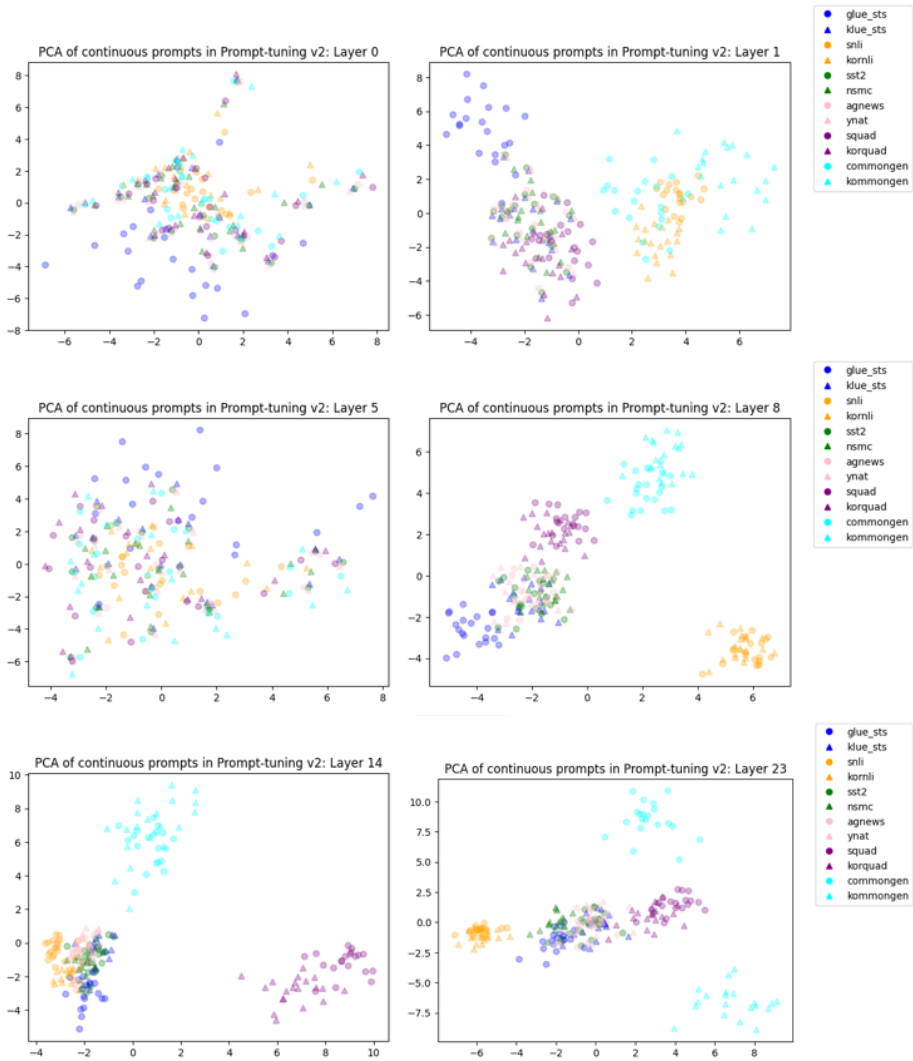

Figure 11 The PCA of continuous prompts in Prompt-tuning v2.

Deep continuous prompts head for the target task after transfer learning, which is varying between task and language sub-prompts.

Next, we analyze task and language sub-prompts to examine the role of each sub-prompt trained to the target tasks. To conserve space, we provide only the PCA results of sub-prompts transferred from QA and NLI. The grey points are the source task and source language sub-prompts, and the others are the sub-prompts trained on

the target tasks. Since we use the sub-prompt length of 10 here, each sub-prompts have 10 points. Similarly, while there are no notable clusters or movements in Prompt-tuning v1 (Figure 12 & Figure 13), there are some patterns in Prompt-tuning v2, where task sub-prompts and language sub-prompts are separated (Figure 14 & Figure 15).

For each source task, the transferred sub-prompts are clustered by the corresponding target task. Also, the clusters are similar to the previous results (Figure 11), where continuous prompts of QA, CG, and NLI especially form the isolated clusters. These results suggest that task sub-prompts move their encoded information from the source task to the target task, which is a consistent movement.

Additionally, in Prompt-tuning v1, the language-transferred sub-prompts are not separated from the task-transferred sub-prompts. However, in Prompt-tuning v2, the language-transferred sub-prompts are not blended into the task-transferred sub-prompts, which means that they do have different skills. We believe that the different role of each sub-prompt drives the source sub-prompts to better utilize PLM in transfer learning. For task transfer learning, the language sub-prompts help the task sub-prompts to adapt to the new task while keeping the information about its language. Likewise, for language transfer learning, the task sub-prompts help the language sub-prompts to learn the new language while keeping the information about the target task.

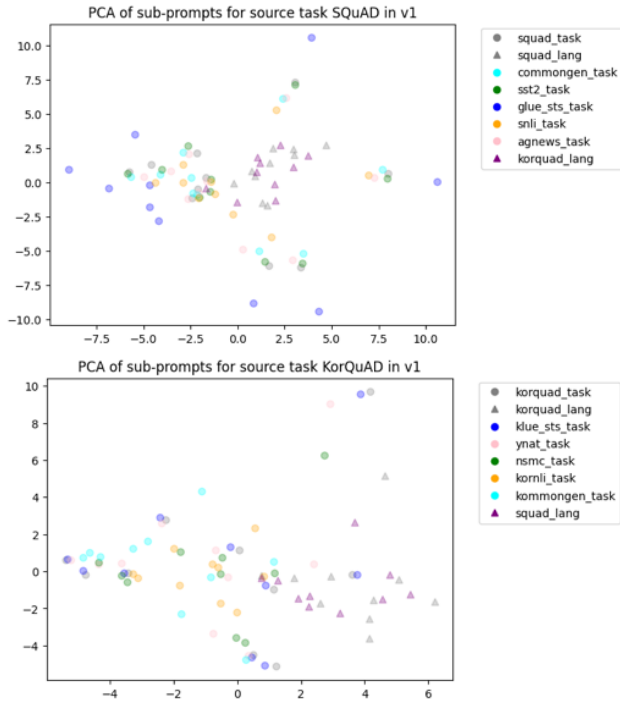

Figure 12 The PCA of continuous prompts transferred from QA task in Prompt-tuning v1.

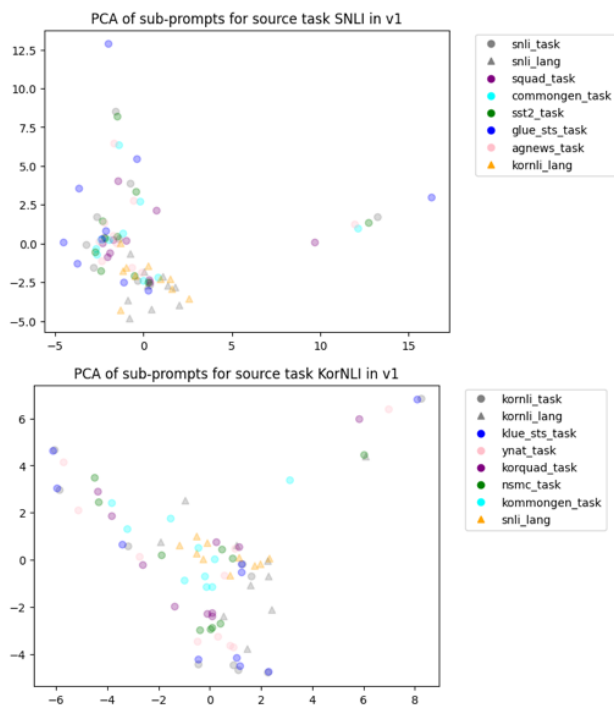

Figure 13 The PCA of continuous prompts transferred from NLI task in Prompt-tuning v1.

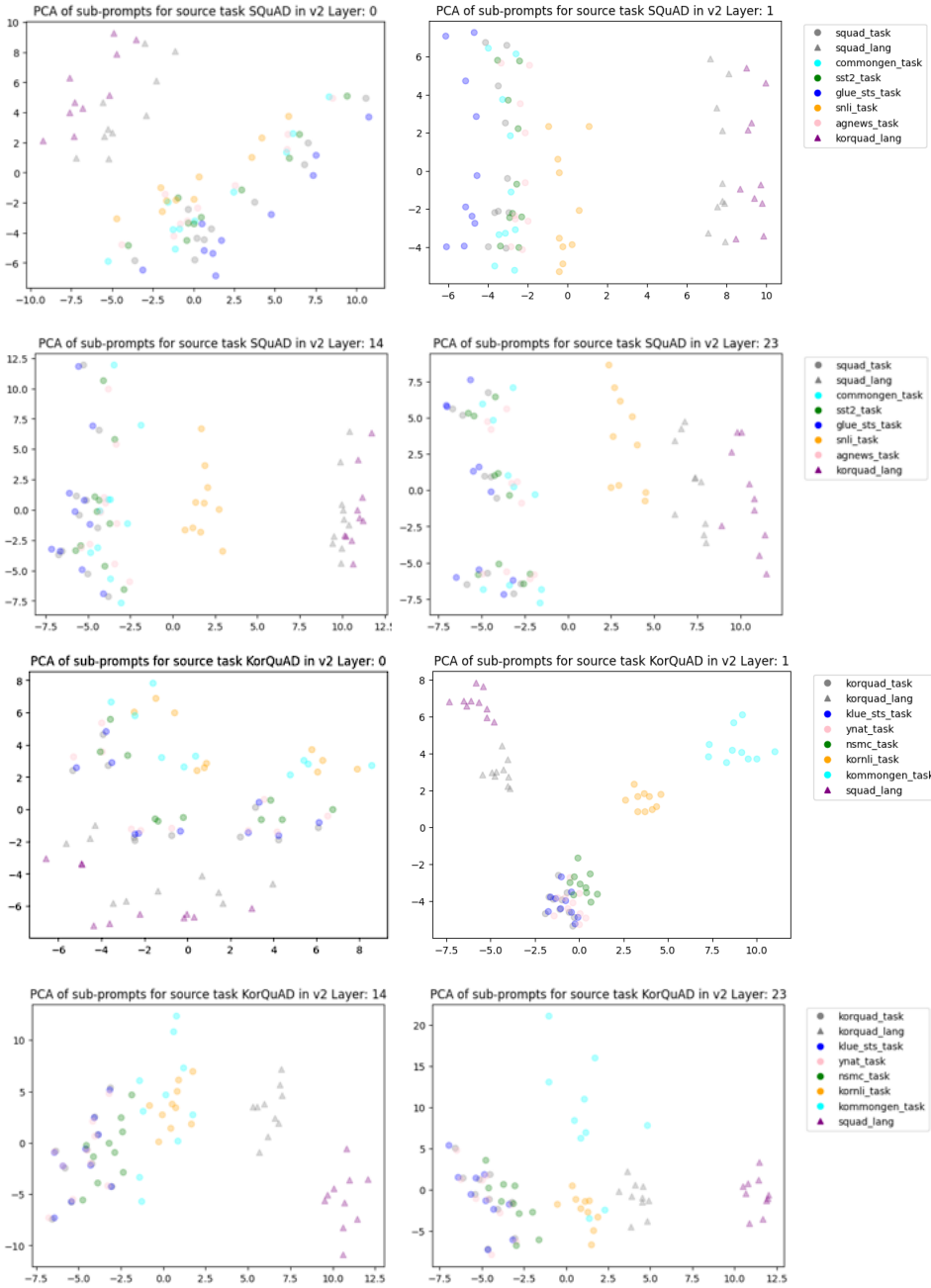

Figure 14 The PCA of continuous prompts transferred from QA task in Prompt-tuning v2.

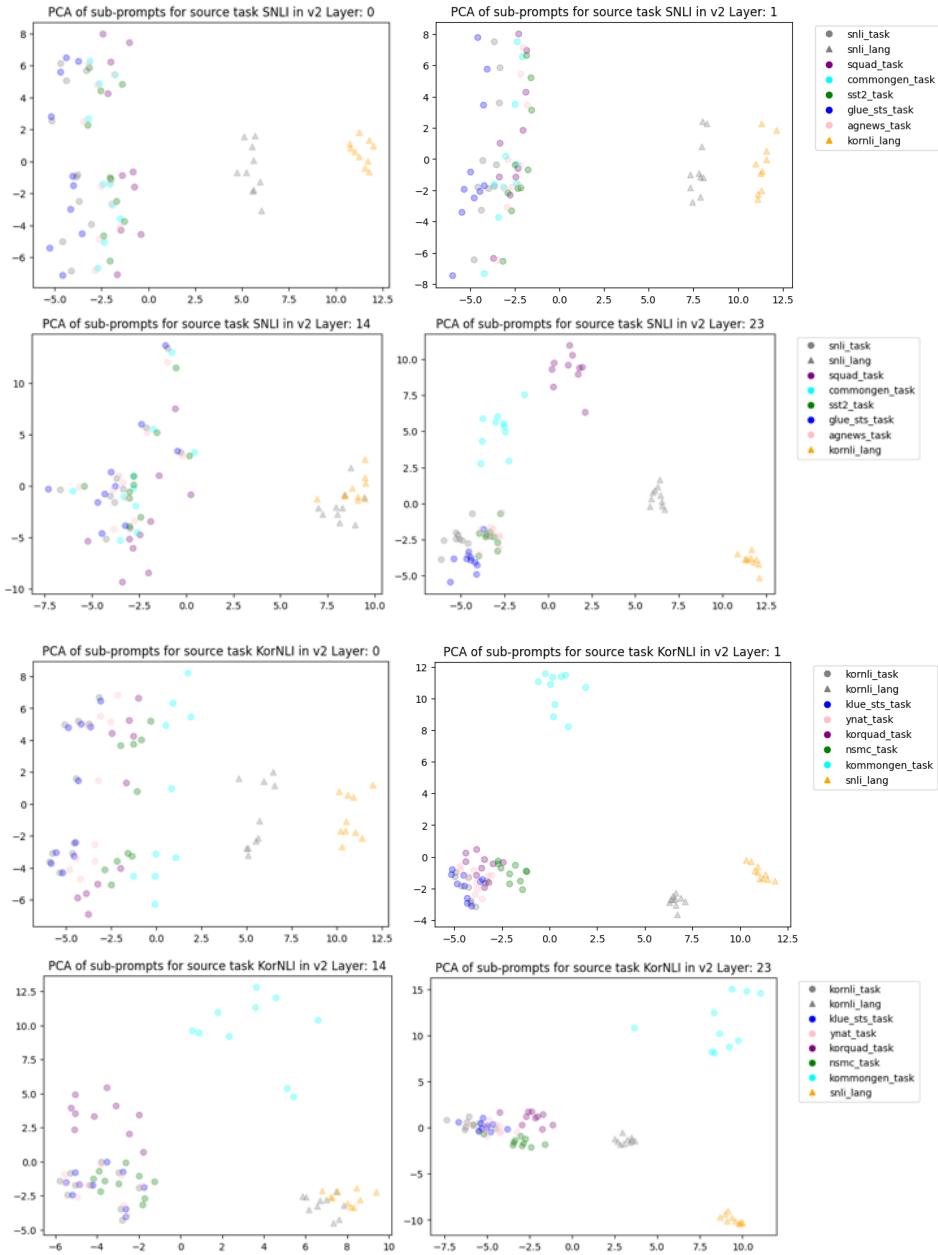

Figure 15 The PCA of continuous prompts transferred from NLI task in Prompt-tuning v2.

5.2.2. The Attention Mechanism

Deep continuous prompts employ content-dependent attention heads, while changing the attention scores in the middle-lower and upper layers significantly.

We find that the lower layers have lower variability (Figure 16), which is a consistent result with the observations in the prior work (Figure 5). The lower layers have the position-dependent heads that give the max attention scores to the previous token. Meanwhile, the attention heads in layers 7~9 have high attention variability regardless of tasks and languages, which means that these layers contain the content-dependent heads. These results show that the attention heads of mGPT have encoded context information in a robust way during pre-training (See Figure 32 in Appendix).

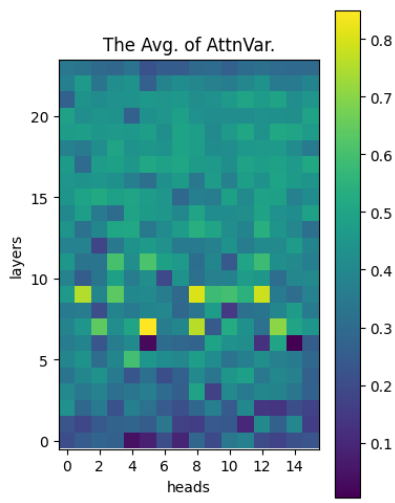

Figure 16 The average of the attention variability of all tasks in both languages.

Figure 17 displays the KL divergence results in Prompt-tuning v1 and Prompt-tuning v2 on STS (GLUE-STS, KLUE-STS) and TC (AGNews, Ynat) tasks. In Prompt-tuning v1, we observe that the attention distribution of the final layer

changes a lot, followed by the initial layers. In Prompt-tuning v2, the layers between 6 and 13 change significantly as well, which consist of the content-dependent heads. Additionally, the results are consistent with the same task rather than with the same language. We present the results of other tasks in Figure 33 and 34 in Appendix.

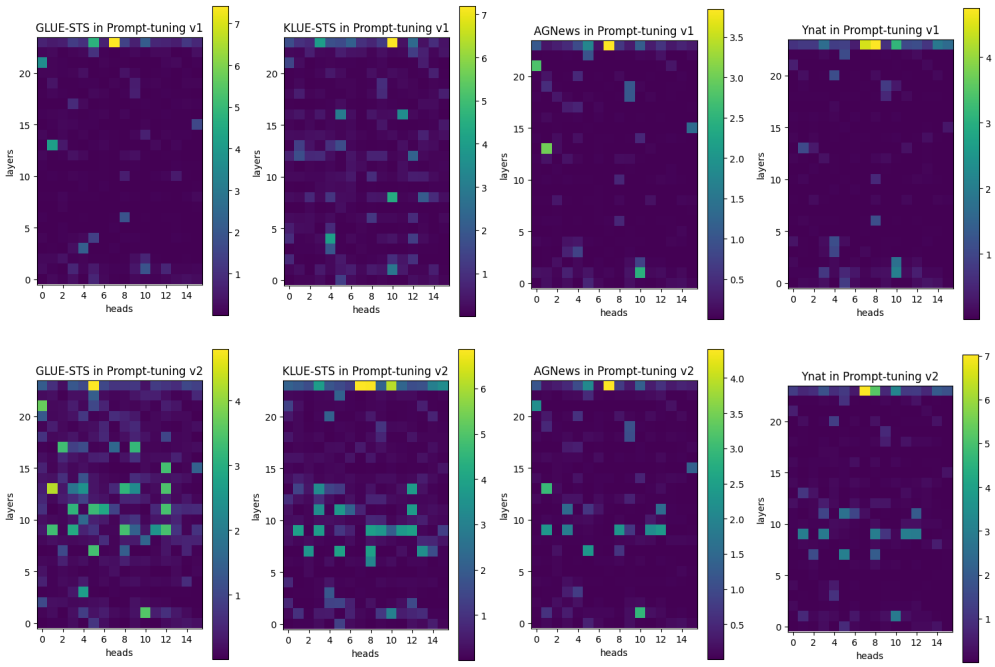

Figure 17 The KL divergence result of STS and TC task.

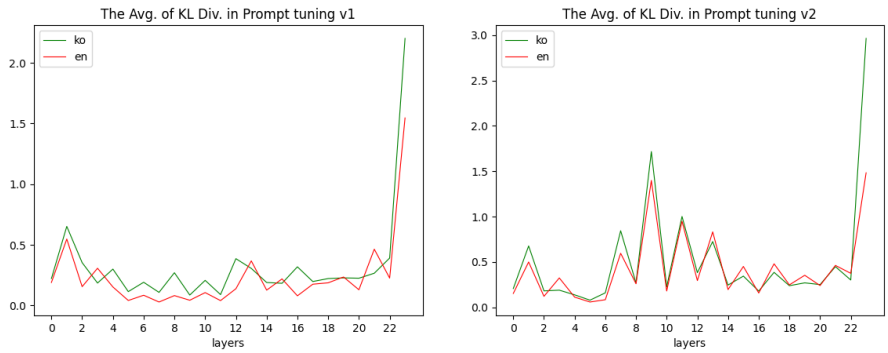

Figure 18 The average of the KL divergence per layer.

To analyze the changes by layer, we plot the average KL divergence of all tasks per language for each layer. Figure 18 shows that the middle layers from 7 to 17 change remarkably in Prompt-tuning v2 with the peaks at the odd layers {7, 9, 11, 13, 15, 17}. On the other hand, in Prompt-tuning v1, the changes in the middle layers are relatively minor. Also, the peaks appear at different layers for each language; the even layers {6, 8, 10} for both languages, the even layers {12, 16} for Korean, and the odd layers {13, 15} for English. This is because they do not have a direct impact on the deeper layers in PLM (Liu et al, 2022) since continuous prompts are injected only in the embedding layer in Prompt-tuning v1.

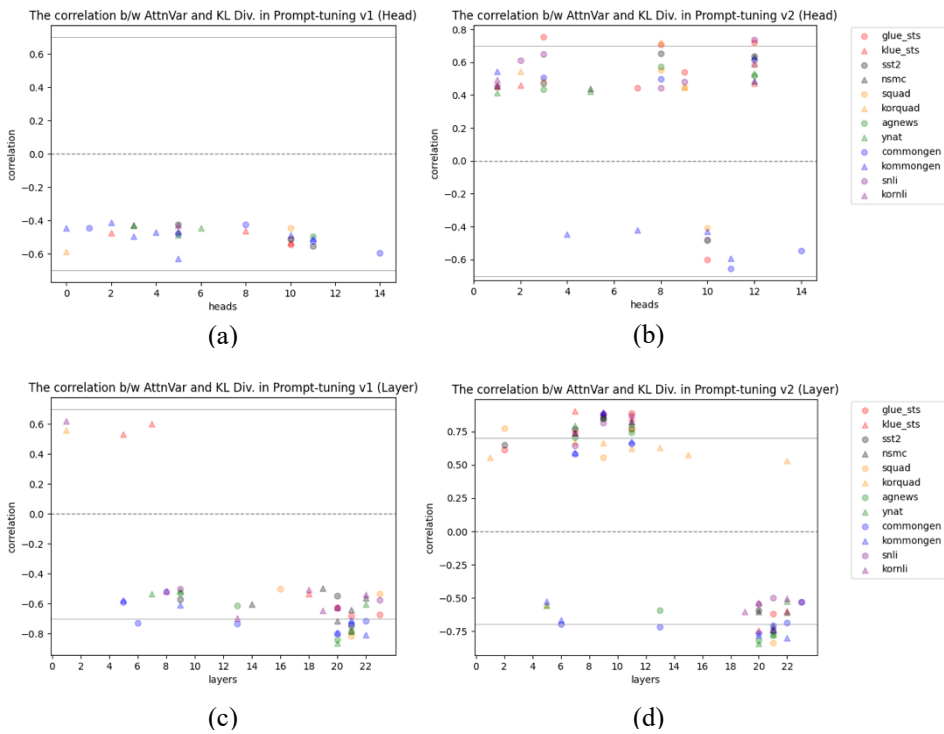

Figure 19 The correlation between the attention variability and the KL divergence.

Next, we present the results of the Pearson correlation (p-value < 0.05) between the attention variability and the KL divergence in each task. For heads in Prompt-tuning v1 (Figure 19-a), most results show negative correlations both per heads and layers, which means that the changes in the attention mechanism do not depend on the content dependency of each head in mGPT. On the other hand, most results show positive correlations in Prompt-tuning v2 (Figure 19-b), where more various tasks have meaningful relationships in more various heads. Thus, we suggest that Prompt-tuning v2 uses the information in the heads more actively than Prompt-tuning v1.

For layers (Figure 19-c & Figure 19-d), grouping the layers into 4 groups according to the depth, we sum up the observations in the attention mechanism. First, the lower layers (layer 0~5), where the position-based heads are concentrated, show a small change both in Prompt-tuning v1 and Prompt-tuning v2. Second, the middle-lower layers (layer 6~11) have different patterns in Prompt-tuning v1 and Prompt-tuning v2. Considering that the content-dependent heads are gathered in layers 7~9, the positive correlations in Prompt-tuning v2 suggest that deep continuous prompts are trained while employing the context information encoded in mGPT.

Third, the middle-upper layers (layer 12~17) do not show significant changes in Figure 18, yet, we observe that the changes in the pair-input type tasks in these layers are comparable to the changes in the middle-lower layers (Figure 20). This implies that the attention heads in the deeper layers are activated to understand the relationship between two sequences. Lastly, the upper layers (layer 18~23) have the same pattern in both Prompt-tuning v1 and Prompt-tuning v2, where the last layer shows a significant change. Simultaneously, they have negative correlations, which implies that the additional elements of PLM are involved in activating the attention head to predict the label in these layers. In the following sections, we will see the

effects of continuous prompts in terms of other structures of mGPT.

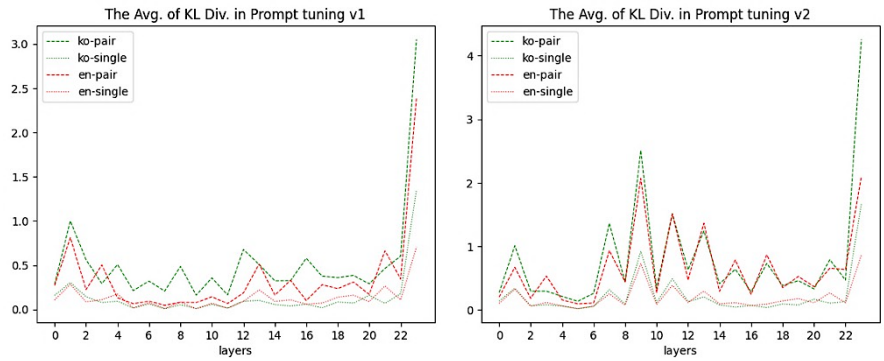

Figure 20 The average of the KL divergence per layer grouped by input type. Single-input type includes SA, TC, CG and pair-input type includes STS, NLI, QA.

Continuous prompts utilize the attention mechanisms more deeply in transfer learning.

After sub-prompts transfer learning, we find that the changes in the deeper layers get greater. Particularly, Figure 21 shows that the changes around layer 17 are notable in Prompt-tuning v2 since these layers have smaller changes without transfer learning (Figure 18). Also, in language sub-prompts transfer (Figure 22), the gap between each language is larger than the gap without transfer learning (Figure 18). In Prompt-tuning v1, the gap between the peaks at layer 2 is large and the gaps become wider after layer 14. In Prompt-tuning v2, the gap between languages in layer 7 is large and the gaps fluctuate in the middle and upper layers.

Considering that the attention changes of the same task in English and Korean are parallel in Figure 17, we believe that the language sub-prompts trained on each language cause the difference between languages after transfer learning.

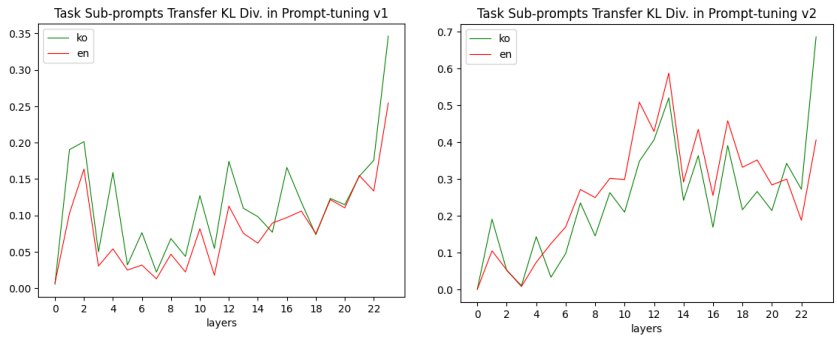

Figure 21 The average of the KL divergence per layer, after task sub-prompts transfer.

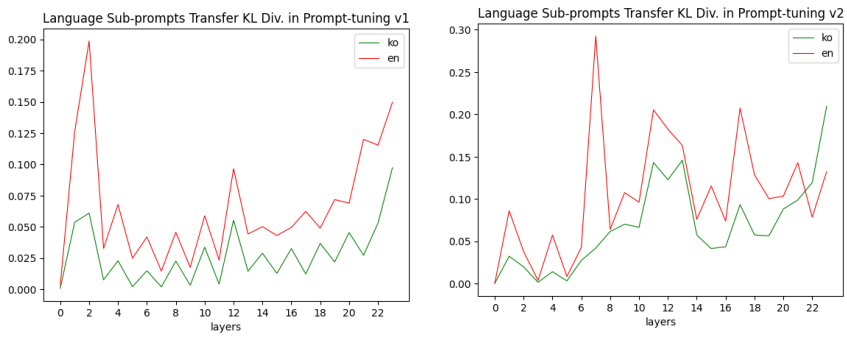

Figure 22 The average of the KL divergence per layer, after language sub-prompts transfer.

5.2.3. The Activated Neurons

The activated neurons of continuous prompts are task-specific in the deeper layers, whereas other features of PLM make the neurons common in the second to last layer.

The results of the average ON score between all combinations of tasks in each language are reported in Figure 23, where the lower the score is, the more task-specific the layer is. We find that the scores in the first layer are high and the scores

reduce rapidly in the second layer, which means that the first layer has encoded the common information regardless of task and language. Meanwhile, the results in layers from 2 to 5 have some fluctuations in Prompt-tuning v1, while there are few fluctuations in Prompt-tuning v2. The middle layers (layer 6~17) also have fluctuations, but with more narrow gaps between layers. Notably, the scores become the lowest in layer 21 in Prompt-tuning v1 and in layer 20 in Prompt-tuning v2.

While these results suggest that the deeper layers have task-specific neurons, the second to last layer shows peaks with high scores near the scores in the first layer, which has not been observed in previous works. Since most tasks have highly similar neurons in the second to last layer, we hypothesize that other features of PLM affect the neurons when solving the target tasks. We also present the ON scores between all tasks in Figure 35 and Figure 36 in Appendix.

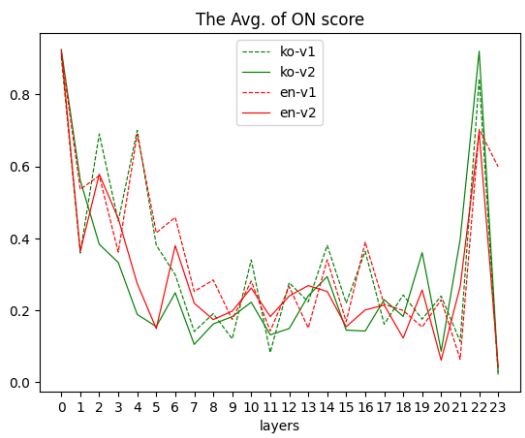

Figure 23 The average ON score between every combination between all tasks.

The task-specific neurons are activated consistently after transfer learning.

After sub-prompts transfer learning, we get the ON score between source tasks and target tasks, and between source tasks and target tasks that are initialized by the

source task. We expect that the score keeps or gets higher with transfer learning because source sub-prompts would learn the knowledge about the target task. To be specific, the kept score means that consistent neurons are activated with and without transfer learning. The higher score indicates that more similar neurons are activated when solving the target tasks. We observe both in the results of transfer learning.

For task sub-prompts transfer, in Figure 24, the scatters are labeled according to the source tasks in the legend and the texts for each scatter are the target tasks. For instance, the yellow point with 'glue_sts' in Figure 24-a has the ON score between SQuAD and GLUE-STS as the x-axis and the ON score between SQuAD and GLUE-STS, which is transferred from SQuAD, as the y-axis.

Figure 24 shows two groups of clusters. The orange circles are clusters with similar ON scores between tasks with and without transfer learning. This cluster suggests that the patterns of the activated neurons between the source tasks and target tasks are consistent when the sub-prompts of the target task are initialized from the sub-prompts of the source task. Also, the blue circles are clusters with higher ON scores between tasks with transfer learning than without transfer learning, which means that the target task has more commonly activated neurons with the source task after transfer learning.

For language sub-prompts transfer, in Figure 25, the scores are maintained (orange box) or raised (blue box) in most cases. Additionally, we compare the scores between languages. In Prompt-tuning v1, the scores between STS, TC, and NLI are similar between languages, and in Prompt-tuning v2, the scores between CG and the ones between QA are similar between languages.

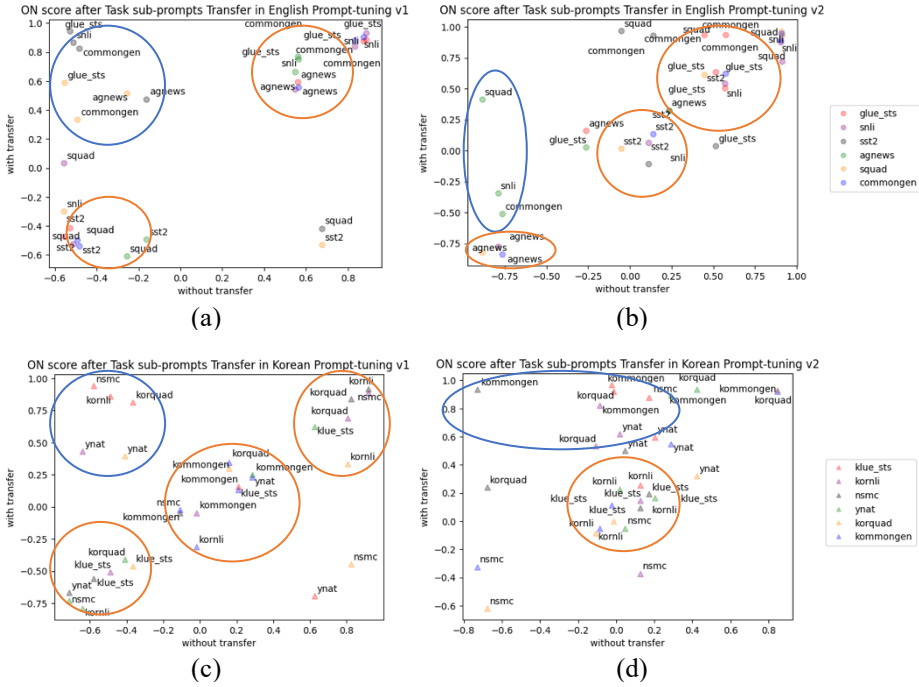

Figure 24 The results of ON score in the last layer without and with task sub-prompts transfer.

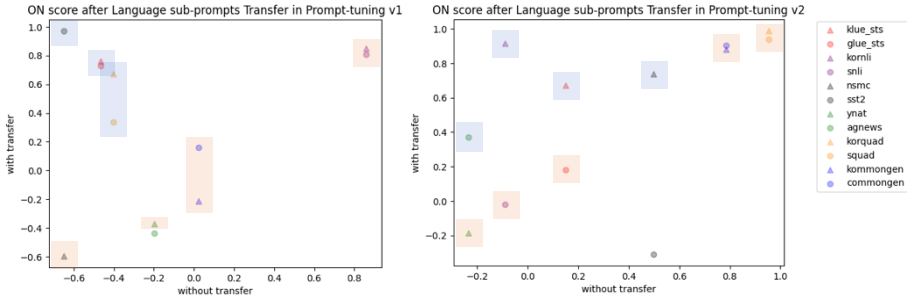

Figure 25 The results of ON score in the last layer without and with language sub-prompts transfer.

5.2.4. The Label Space

Prompt-tuning does not improve the isotropy of PLM.

We first present the cosine-based isotropy in the PLM and the prompt-tuned PLM in Figure 26. We measure the isotropy by calculating the cosine similarities between randomly sampled token pairs for each dataset (Equation 18) and getting the average of them. Again, the lower the cosine similarity is, the higher the isotropy is. Additionally, we get the cosine similarity between the decoding token and the randomly sampled token (Equation 19) for each dataset and get the average as well.

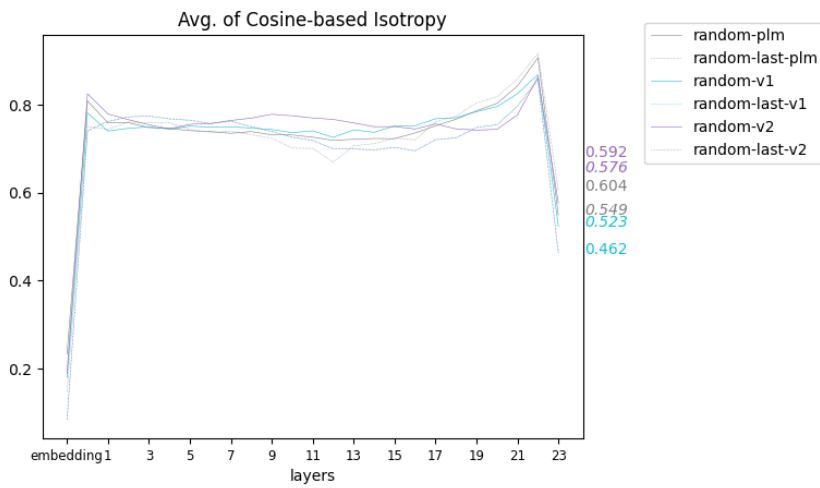

Figure 26 The average cosine-based isotropy. 'random' is $\text{Isotropy}^{m}_{\text{cosine}}$ and 'random-last' is $\text{Isotropy}^{\text{decoding token}}_{\text{cosine}}$.

In the first embedding layer, the cosine similarity is low both in the pre-trained PLM (grey lines) and the prompt-tuned PLM (blue lines and purple lines), which means that words have discriminative representations. However, the cosine similarity increases near 0.8 and keeps high, and then begins increasing around layer

15. Reaching a peak at layer 22, it decreases at the last layer. Also, the decoding token does not show different patterns from other tokens. Thus, even though the decoding token has a specific role, its semantic representation is not discriminative from other common tokens. Finally, despite the trivial gaps between the pre-trained PLM and the prompt-tuned PLM, the isotropy is not improved after Prompt-tuning, which means that the embedding space of mGPT is anisotropic.

Continuous prompts interact with the representation space of PLM through layers.

To repeat, the low isotropy leads to the high similarity between semantically non-related random words. Then the lowest isotropy at layer 22 is one of the explanations to the special phenomenon we observed in the previous section, where the similarities between any tasks increase at layer 22. Indeed, the model decodes the label word using the hidden state of the last layer (layer 23) which contains the information passing the second to last layer (layer 22). Thus, the observed high similarity in the activated neurons at the second to last layer (Figure 23) could be because the labels each model has to decode are actually similar in the representation space.

To test our intuitions, we present the cosine similarities in the label space. Figure 27 illustrates that the cosine similarities of the decoding token </*s*> with the gold label words ($\text{dist}_{\text{label}}$) get higher than the ones with the non-gold label words ($\text{dist}_{\text{non-label}}$) in the deeper layers. Particularly, the second to last layer has the trough, where the gaps between two similarities narrow down. We believe that the anisotropic embedding space of mGPT could lead the task-common neurons in the second to last layer because the desired label words have similar representations in

the anisotropic space. Thus, we conclude that these observations are one of the possible explanations as to why continuous prompts are hard to interpret.

Therefore, we claim that continuous prompts interact with PLM since the embedding spaces and the activated neurons have explainable relations. In addition, while the gap between $\text{dist}_{\text{label}}$ and $\text{dist}_{\text{non-label}}$ is not significant in the PLM, the gap increases after layer 11 in the prompt-tuned model. Accordingly, continuous prompts make use of the task-specific knowledge in the embedding space of the PLM.

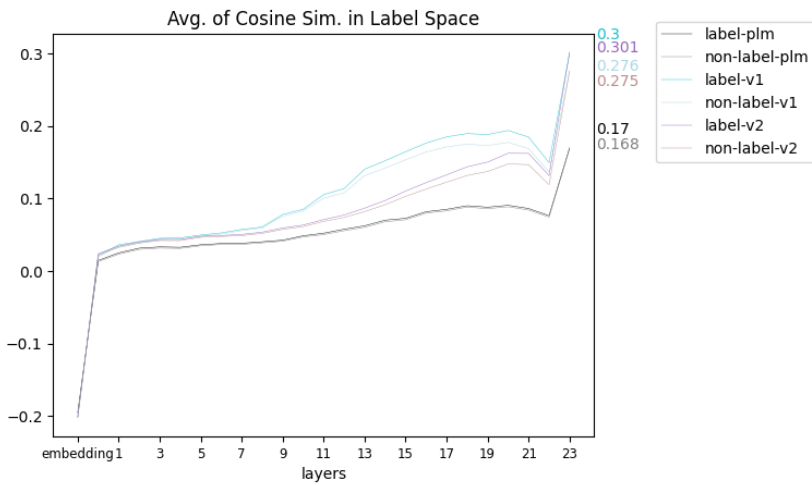

Figure 27 The average of the cosine similarities in the label space.

Prompt-tuning makes the decoding token closer to the label words of target tasks than to the label words of source tasks after transfer learning.

Next, we get the average of the cosine similarities in label space of all combinations of tasks for transfer learning. In Figure 28, the dotted lines, which have a 'pre-' label, are calculated from the prompt-tuned PLM on the source task. The results show that the similarity between the decoding token and 'label-' is higher than the similarity between the decoding token and 'pre-label-', especially after layer

7. Moreover, the gaps between ‘pre-label-’ and ‘pre-non-label-’ are narrow than the gaps between ‘label-’ and ‘non-label-’, which means that the decoding token makes its representation adapt to the target task during transfer learning.

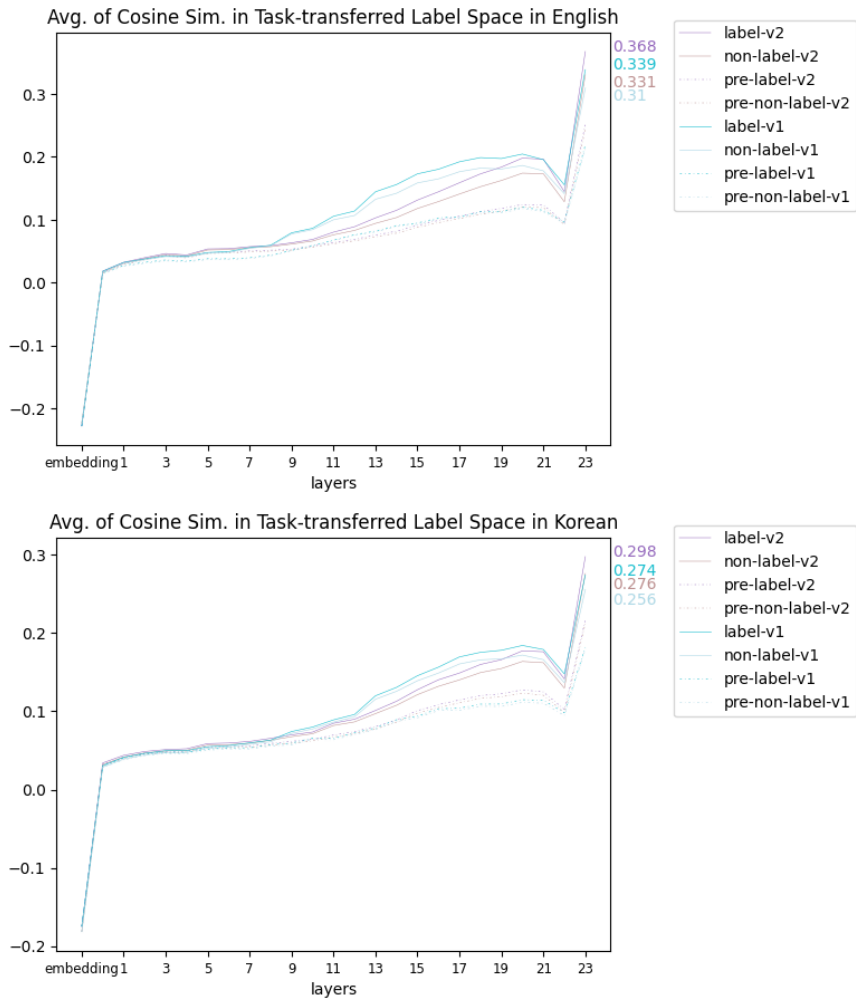

Figure 28 The average of the cosine similarities in the task-transferred label space.

In the case of language transfer learning (Figure 29), the gaps between the ‘pre-’ labeled lines are similar to the gaps between the labeled without ‘pre-’ lines, which means that the decoding token conserves the information from the source task.

Simultaneously, the similarity between the decoding token and the transferred label word is higher. Notably, the gap between the dotted lines and the solid lines widens from the lower layers. These results imply that the decoding token makes its representation adapt to the target language in a different way from the task transfer learning.

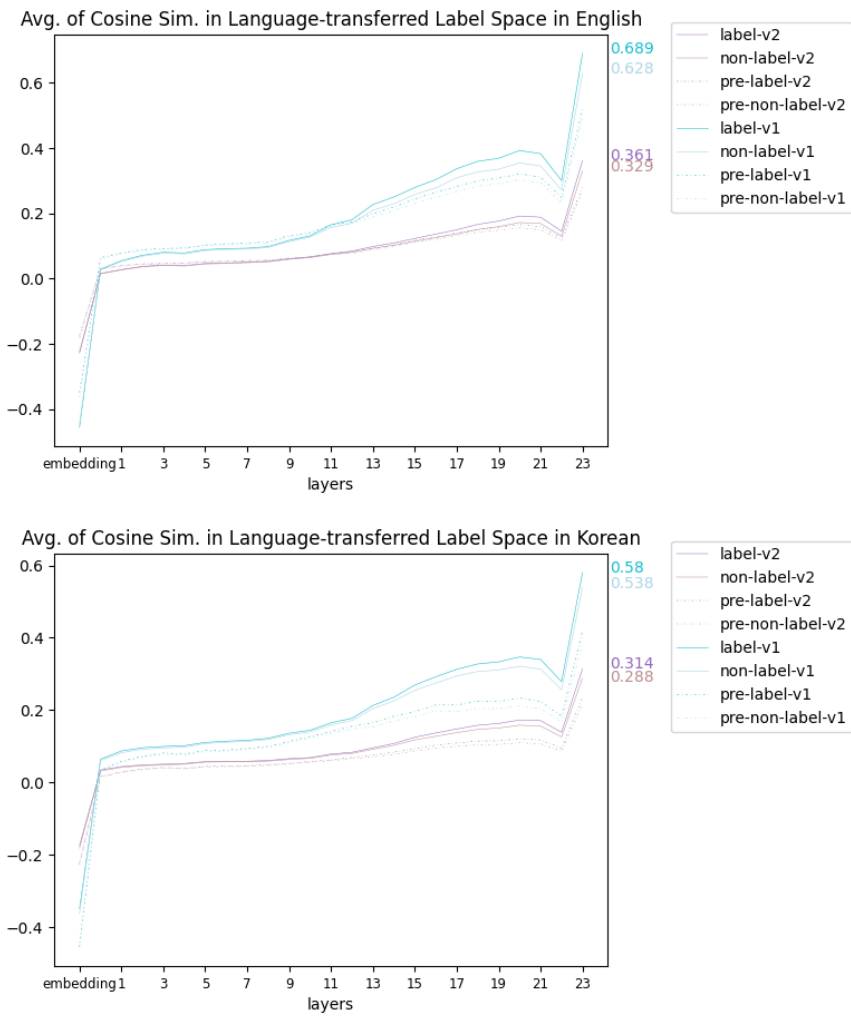

Figure 29 The average of the cosine similarities in the language-transferred label space.

5.3. Ablation Studies

5.3.1. Zero-shot Cross-lingual Results

We measure the zero-shot cross-lingual performance without and with the factorization of sub-prompts. Without factorizing, we use the prompts of the source language to evaluate the corresponding task in the target language. With factorizing, we use the source language sub-prompts instead of the target language sub-prompts.

To this end, the model generates the label words in the source language. If the target task is KLUE-STS, the task sub-prompts of GLUE-STS are prepended and the model generates the label words in English. For QA tasks, we let the model generate without any other controls. Since the label words should be the same for classification tasks, we exclude TC (AGNews, Ynat) here. Also, we exclude the generation task, CG (CommonGen, KommonGen).

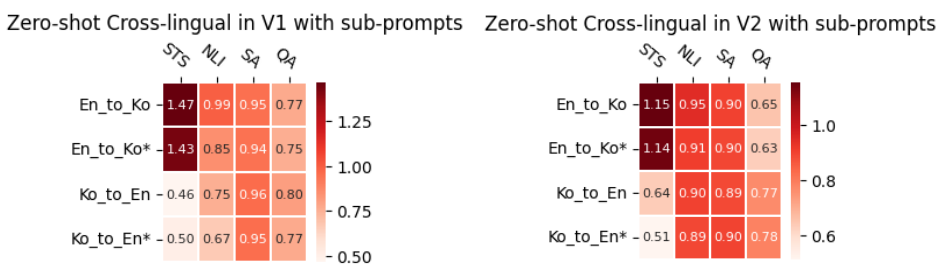

Figure 30 The relative zero-shot cross-lingual performance in Prompt-tuning v1 and Prompt-tuning v2. The rows with '*' are the scores from the target language sub-prompts and the rows without '*' are the scores from the source language sub-prompts.

Figure 30 illustrates the results with the factorized sub-prompts. We report the relative zero-shot performance (zero-shot cross-lingual performance / original

performance). For the scores in the rows named with '*', we use the target language sub-prompts and the ones named without '*', the source language sub-prompts. One example of this is that when we evaluate KLUE-STS using task sub-prompts trained on GLUE-STS, we prepend the Korean sub-prompts for the former case and the English sub-prompts for the latter case.

For STS, English is helpful for Korean, and even the zero-shot cross-lingual scores are higher than the original full-shot scores. On the contrary, Korean is not as useful as English since the score drops about 50% in Prompt-tuning v1 and about 30% in Prompt-tuning v2. Similarly, for NLI, only English helps solve Korean NLI in Prompt-tuning v1. Meanwhile, in Prompt-tuning v2, Korean is also helpful for solving English NLI. This is because, since KLUE-STS and KorNLI show relatively low original performances, their task sub-prompts do not have enough knowledge to solve the tasks. For SA, the scores are highly comparable to the original full-shot scores in both cross-lingual settings. For QA, the scores drop about 20% on average, which means that the extraction task is more sensitive to language than the classification task.

In Section 5.2.1, we observe that the language sub-prompts have separate spaces from the task sub-prompts in Prompt-tuning v2, which means that they encode different information. Thus, we expect that the factorized sub-prompts are practical since the model can utilize the source language information from the language sub-prompts. However, factorized prompts are not always the best choice. In most cases, the scores without factorizing are higher than the ones with factorizing, or the differences between them are trivial.

These results suggest that each sub-prompt employs the information from the prepended frozen sub-prompts during updating parameters. We conclude that even

if each target language sub-prompts encode language-specific information, the relationship between the source task sub-prompts and the source language sub-prompts is powerful. Thus, more systematic factorizing would be required to gain profit in the zero-shot cross-lingual setting.

5.3.2. Deep continuous prompts compression

In this section, we investigate whether deep continuous prompts can better utilize the specific layers, where the effects of Prompt-tuning are notable. To this end, we feed continuous prompts to the layers, where changes and roles are clear.

In Section 5.2.2, we find that the changes in the attention distribution show a zigzag trend line. The peaks are detected in layers {1, 7, 9, 11, 13, 15, 17, 21} and the troughs are detected in layers {2, 8, 10, 12, 14, 16, 18, 20, 22}, both in English and Korean. We hypothesize that continuous prompts are more beneficial for the layers at the peaks than the layers at the troughs.

Furthermore, looking at the results of the activated neurons in Section 5.2.3, the last layer (23) is a task-specific layer, and the second to last layer shows a special phenomenon, where the neurons of all tasks are similar to each other. Also, the neurons in layer 20 have the lowest ON score, which means that the neurons are activated depending on tasks.

Motivated by these observations, we group the layers into two categories: peak and trough. The peak group includes layers {0, 1, 7, 9, 11, 13, 15, 17, 20, 21, 22, 23}, and the trough group includes layers {0, 2, 8, 10, 12, 14, 16, 18, 20, 21, 22, 23}. We set the first layer {0} and the last four layers {20, 21, 22, 23} in common, in

consideration that the former is the input layer, and the latter ones are task-specific layers. Each group includes half of the number of layers of mGPT, which means that they have half the parameters of continuous prompts in Prompt-tuning v2.

This method is similar to Liu et al. (2022). They conducted the ablation study of Prompt-tuning v2 by adding continuous prompts to certain layers grouping into two groups; the first four layers and the last four layers. On the contrary, this study tries to classify the layers in an explainable way. Also, we believe that the alternating dense and sparse attention mechanism of mGPT prevents the performances from a significant drop when excluding some layers.

	English		Korean	
Task	Peak	Trough	Peak	Trough
STS	**84.17**	83.74	60.32	**63.26**
NLI	**85.35**	84.85	**61.97**	58.18
SA	**90.02**	89.79	**86.73**	86.25
TC	**86.66**	86.46	**83.75**	82.66
QA	**64.40/49.29**	61.82/47.11	**68.82/62.41**	66.57/60.42
CG	**82.56**	81.62	**90.43**	90.21

Table 3 The performance results of prompt compression with peak-layers and trough-layers.

Table 3 shows the performances of each group on all tasks. Compared to the results of vanilla Prompt-tuning v2 (see Table 2), most scores drop but raise in comparison to the results of Prompt-tuning v1. Notably, the scores of both groups on SST2 raise compared to Prompt-tuning v2. Except for KLUE-STS, the peak groups show higher performances than the trough groups. Thus, we conclude that continuous prompts in the peak groups employ the knowledge of mGPT better.

Chapter 6. Conclusion

In this study, we investigated how continuous prompts and deep continuous prompts encode task-relevant knowledge and employ the knowledge from the PLM. Using mGPT, we conducted the experiments on various tasks, including classification and generation in each language.

First, we found that deep continuous prompts show task-specific representations in the deeper layers, while continuous prompts in Prompt-tuning v1 do not. Also, deep continuous prompts have shared task-relevant information within languages. Second, we observed that the changes in the attention mechanism after Prompt-tuning v2 can be explained in terms of the attention variability. The higher the attention variability is where the more significant the changes are. Simultaneously, the last four layers show negative correlations between the attention variability and the changes in the attention distribution after both Prompt-tuning v1 and Prompt-tuning v2. Thus, we concluded that these layers play another key role to solve the target task.

Third, the response of the model suggests that the deeper layers have more task-specific neurons. Unlike previous studies, we reported a special phenomenon in the second to last layer, where most of all tasks have unexpected common neurons. Also, the transferred neurons are consistent with tasks and languages. Lastly, we confirmed that the decoding token becomes closer to the label words after Prompt-tuning. Additionally, with the label space, we offered explanations as to the special phenomenon in the activated neurons. Since the desired label words are actually similar between tasks in the anisotropic space, the activated neurons in the second to

last layer show task-common behavior. We hope that this study provides a guide to the explainable continuous prompts and the explainable PLM.

This study, however, has some limitations. Although we explore the multilingual space of PLM, our findings are limited to English and Korean. We choose Korean as a non-English language because Korean is understudied in Prompt-tuning and has different linguistic properties from English, such as typology. Also, even though our experiments have a fixed seed (42), other trials with other seed numbers are required, since different seed numbers can lead to fluctuating results. Additionally, we fail to analyze the observations relating to the performances. We encourage further studies to include more various languages and more systemic experiments.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel HerbertVoss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In Advances in Neural Information Processing Systems, volume 33, pages 1877–1901. Curran Associates, Inc. Christopher Clark.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. XNLI: Evaluating Cross-lingual Sentence Representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2475–2485, Brussels, Belgium. Association for Computational Linguistics.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge Neurons in Pretrained Transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making Pre-trained Language Models Better Few-shot Learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Yuxian Gu, Xu Han, Zhiyuan Liu, and Minlie Huang. 2022. PPT: Pre-trained Prompt Tuning for Few-shot Learning. In *Proceedings of the 60th Annual Meeting of the Association for*

Computational Linguistics (Volume 1: Long Papers), pages 8410–8423, Dublin, Ireland. Association for Computational Linguistics.

Micha Heilbron, Benedikt Ehinger, Peter Hagoort, and Floris P. de Lange. Tracking Naturalistic Linguistic Predictions with Deep Neural Language Models. 2019. *Conference on Cognitive Computational Neuroscience, 2019*. arXiv: 1909.04400.

Daniel Khashabi, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, and Yejin Choi. 2022. Prompt Waywardness: The Curious Case of Discretized Interpretation of Continuous Prompts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3631–3643, Seattle, United States. Association for Computational Linguistics.

Boseop Kim, HyoungSeok Kim, Sang-Woo Lee, Gichang Lee, Donghyun Kwak, Jeon Dong Hyeon, Sunghyun Park, Sungju Kim, Seonhoon Kim, Dongpil Seo, Heungsub Lee, Minyoung Jeong, Sungjae Lee, Minsub Kim, Suk Hyun Ko, Seokhun Kim, Taeyong Park, Jinuk Kim, Soyoung Kang, et al.. 2021. What Changes Can Large-scale Language Models Bring? Intensive Study on HyperCLOVA: Billions-scale Korean Generative Pretrained Transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3405–3424, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In International Conference on Learning Representations.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The Power of Scale for Parameter-Efficient Prompt Tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. 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*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-Tuning: Optimizing Continuous Prompts for Generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural*

Language Processing (Volume 1: Long Papers), pages 4582–4597, Online. Association for Computational Linguistics.

Seungyoung Lim, Myungji Kim, and Jooyoul Lee. 2019. Korquad1.0: Korean qa dataset for machine reading comprehension. arXiv preprint arXiv:1909.07005.

Bill Yuchen Lin, Wangchunshu Zhou, Ming Shen, Pei Zhou, Chandra Bhagavatula, Yejin Choi, and Xiang Ren. 2020. CommonGen: A Constrained Text Generation Challenge for Generative Commonsense Reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1823–1840, Online. Association for Computational Linguistics.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021. Gpt understands, too. arXiv:2103.10385.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-Tuning: Prompt Tuning Can Be Comparable to Fine-tuning Across Scales and Tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Jinwoo Min, Seung-Hoon Na, Jong-Hoon Shin, Young-Kil Kim. 2019. RoBERTa for Korean Natural Language Processing: Named Entity Recognition, Sentiment Analysis, Dependency Parsing. In: *Proceedings of the KIISE Korea Software Congress*, pages 407-409.

Jinwoo Min, Seung-Hoon Na, Dongwook Shin, Seon-Hoon Kim, Inho Kang. 2021. Prefix-tuning for Korean Natural language processing. In: *Annual Conference on Human and Language Technology*. Human and Language Technology, pages 622-624.

Bo Pang and Lillian Lee. 2005. Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Sungjoon Park, Jihyung Moon, Sungdong Kim, Won Ik Cho, Jiyoon Han, Jangwon Park, Chisung Song, Junseong Kim, Yongsook Song, Taehwan Oh, et al. 2021. Klue: Korean language understanding evaluation. arXiv preprint arXiv:2105.09680.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. OpenAI Blog, 1(8):9.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with

a unified text-totext transformer. Journal of Machine Learning Research, 21(140):1–67.

Sara Rajaee and Mohammad Taher Pilehvar. 2022. An Isotropy Analysis in the Multilingual BERT Embedding Space. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1309–1316, Dublin, Ireland. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know What You Don't Know: Unanswerable Questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Timo Schick, Helmut Schmid, and Hinrich Schutze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, pages 5569–5578. International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021a. Exploiting Cloze-Questions for Few-Shot Text Classification and Natural Language Inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021c. Few-Shot Text Generation with Natural Language Instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jaehyung Seo, Chanjun Park, Hyeonseok Moon, Sugyeong Eo, Myunghoon Kang, Seounghoon Lee, and Heuiseok Lim. 2021. KommonGen: A Dataset for Korean Generative Commonsense Reasoning Evaluation. In *Proceedings of the 33th Annual Conference on Human & Cognitive Language Technology*.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, and Nako Sung. 2022. On the Effect of Pretraining Corpora on In-context Learning by a Large-scale

Language Model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5168–5186, Seattle, United States. Association for Computational Linguistics.

Oleh Shliazhko, Alena Fenogenova, Maria Tikhonova, Vladislav Mikhailov, Anastasia Kozlova, and Tatiana Shavrina. 2022. mgpt: Few-shot learners go multilingual. arXiv preprint arXiv:2204.07580.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Yusheng Su, Xiaozhi Wang, Yujia Qin, Chi-Min Chan, Yankai Lin, Huadong Wang, Kaiyue Wen, Zhiyuan Liu, Peng Li, Juanzi Li, Lei Hou, Maosong Sun, and Jie Zhou. 2022. On Transferability of Prompt Tuning for Natural Language Processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3949–3969, Seattle, United States. Association for Computational Linguistics.

Pedro Javier Ortiz Suárez, Laurent Romary, and Benoît Sagot. 2020. A Monolingual Approach to Contextualized Word Embeddings for Mid-Resource Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1703–1714, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in neural information processing systems, pages 5998–6008.

Jesse Vig and Yonatan Belinkov. 2019. Analyzing the Structure of Attention in a Transformer Language Model. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 63–76, Florence, Italy. Association for Computational Linguistics.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. 2022a. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. arXiv preprint arXiv:2205.12647.

Tu Vu, Brian Lester, Noah Constant, Rami Al-Rfou', and Daniel Cer. 2022b. SPoT: Better Frozen Model Adaptation through Soft Prompt Transfer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5039–5059, Dublin, Ireland. Association for Computational Linguistics.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman.

2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. 2022. Finding Skill Neurons in Pre-trained Transformer-based Language Models. arXiv preprint at arXiv:2211.07349.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. 2014. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. 2015. In *NeurIPS*, pages 649–657.

Appendix

Task	English Verbalizer (train/validation/test)
STS	similar (2994/629), different (2755/871)
NLI	entailment (183414/3329/3368), contradiction (183185/3278/3237), neutral (182762/3235/3219)
SA	positive (37569/444), negative (29780/428)
TC	business, scitech, sports, world (30000/1900)

Table 4 The verbalizers (label words) and the number of each example for English classification datasets. If test dataset is not provided, we use validation dataset instead.

Task	Korean Verbalizer (train/validation/test)
STS	유사 (5602/220), 상이 (6066/299)
NLI	함의 (183382/523/1670), 모순 (183382/524/1670), 중립 (183382/523/1669)
SA	긍정 (74825/25171), 부정 (75710/24826)
TC	정치 (7379/722), 경제 (6118/1348), 생활문화 (5751/1369), 사회 (5133/3701), IT과학 (5235/554), 세계 (8320/835), 스포츠 (7742/578)

Table 5 The verbalizers (label words) and the number of each example for Korean classification datasets. If test dataset is not provided, we use validation dataset instead.

Task	Batch size	Epochs	Max length
STS	16	10	180
NLI	32	16	150
SA	32	20	256
TC	64/32	10	80
QA	16	12	400
CG	16	20	60

Table 6 The hyperparameters of each task. For batch size in task TC, 64 is for English dataset and 32 is for Korean dataset.

	GLUE-STS	SNLI	SST2	AGNews	SQuAD	CommonGen
GLUE-STS	81.1	76.97	88.41	85.37	58.98/44.06	78.22
SNLI	83.11	76.5	88.76	85.47	59.65/45.14	79.81
SST2	82.99	74.14	88.76	85.88	58.91/44.38	76.5
AGNews	83.05	71.63	89.44	85.2	57.83/42.74	77.63
SQuAD	84.13	75.91	89.33	85.59	58.03/43.57	80.53
CommonGen	83.69	75.36	89.1	86.01	58.76/44.02	79.56

Table 7 Cross-task performance in English Prompt-tuning v1. The columns are the source tasks and the rows are the target tasks. The grey cells are the baseline scores.

	GLUE-STS	SNLI	SST2	AGNews	SQuAD	CommonGen
GLUE-STS	84.38	85.94	90.59	87.71	66.43/51.16	83.65
SNLI	86.04	85.57	92.08	87.87	66.79/51.78	80.71
SST2	85.55	85.78	90.02	87.74	66.16/50.97	78.09
AGNews	84.67	85.71	91.16	87.31	66.29/51.02	81.75
SQuAD	84.64	86.16	91.28	87.8	66.18/51.04	80.5
CommonGen	84.41	86.12	91.28	88	66.29/51.07	78.86

Table 8 Cross-task performance in English Prompt-tuning v2. The columns are the source tasks and the rows are the target tasks. The grey cells are the baseline scores.

	KLUE-STS	KorNLI	NSMC	Ynat	KorQuAD	KommonGen
KLUE-STS	41.15	37.09	83.67	78.01	60.87/54.57	83.26
KorNLI	64.4	47.12	83.96	78.93	63.15/56.87	82.31
NSMC	37.11	47.88	83.7	79.88	63.35/57.23	81.6
Ynat	54.68	52.9	83.92	79.67	58.99/53.13	82.61
KorQuAD	64.39	53.96	84.33	82.29	61.31/55.12	82.98
KommonGen	44.92	40.01	84.17	80.18	63.55/56.99	82.9

Table 9 Cross-task performance in Korean Prompt-tuning v1. The grey cells are the baseline scores.

	KLUE-STS	KorNLI	NSMC	Ynat	KorQuAD	KommonGen
KLUE-STS	59.24	62.37	86.71	84.3	71.16/65.17	90.89
KorNLI	78.98	62.48	87.26	84.39	71.96/66.24	90.81
NSMC	68.37	63.27	86.73	84.58	71.49/65.53	90.65
Ynat	57.74	63.74	86.84	83.59	71.53/65.58	90.94
KorQuAD	75.7	63.08	86.96	85.11	70.74/64.65	90.66
KommonGen	75.87	62.98	87.48	85.24	72.53/66.74	90.48

Table 10 Cross-task performance in Korean Prompt-tuning v2. The grey cells are the baseline scores.

	STS	NLI	ST	TC	QA	CG
En_to_ko	43.87	50.88	83.71	78.43	63.85/57.55	81.37
Ko_to_en	81.61	78.82	89.33	85.05	58.50/43.75	80.55

Table 11 Cross-lingual performance in Prompt-tuning v1. ‘En_to_ko’ is English to Korean and ‘Ko_to_en’ is Korean to English.

	STS	NLI	ST	TC	QA	CG
En_to_ko	72.02	63.26	87.39	74.47	72.01/66.17	91.13
Ko_to_en	83.55	86	89.67	87.51	66.09/51.19	83.67

Table 12 Cross-lingual performance in Prompt-tuning v2. ‘En_to_ko’ is English to Korean and ‘Ko_to_en’ is Korean to English.

	STS	NLI	ST	QA
En_to_ko	37.68	57.49	85.32	34.93/47.84
En_to_ko*	40.90	50.93	84.51	33.50/46.25
Ko_to_en	60.42	46.84	79.55	42.29/49.17
Ko_to_en*	59	40.17	78.91	41.71/48.27

Table 13 Zero-shot cross-lingual performance in Prompt-tuning v1 with the factorized sub-prompts. ‘En_to_ko’ is English to Korean and ‘Ko_to_en’ is Korean to English. The rows with ‘*’ are the scores using the target language sub-prompts and the rows without ‘*’ are the scores using the source language sub-prompts.

	STS	NLI	ST	QA
En_to_ko	68.18	59.10	78.10	42.34/49.79
En_to_ko*	67.34	56.77	77.62	40.95/48.60
Ko_to_en	53.86	77.16	80.04	39.10/52.03
Ko_to_en*	43.44	76.23	80.84	40.04/52.92

Table 14 Zero-shot cross-lingual performance in Prompt-tuning v2 with the factorized sub-prompts. ‘En_to_ko’ is English to Korean and ‘Ko_to_en’ is Korean to English. The rows with ‘*’ are the scores using the target language sub-prompts and the rows without ‘*’ are the scores using the source language sub-prompts.

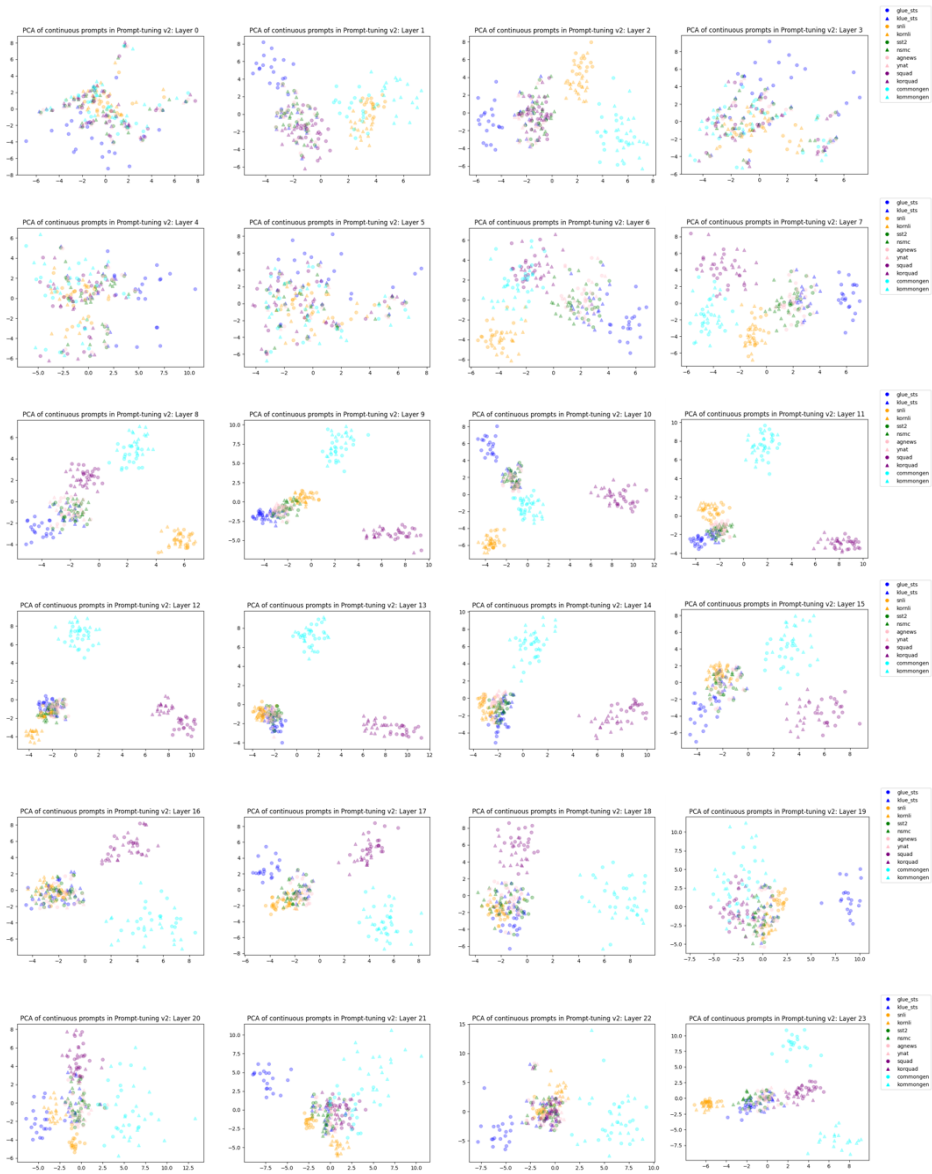

Figure 31 The PCA result in Prompt-tuning v2 over all layers.

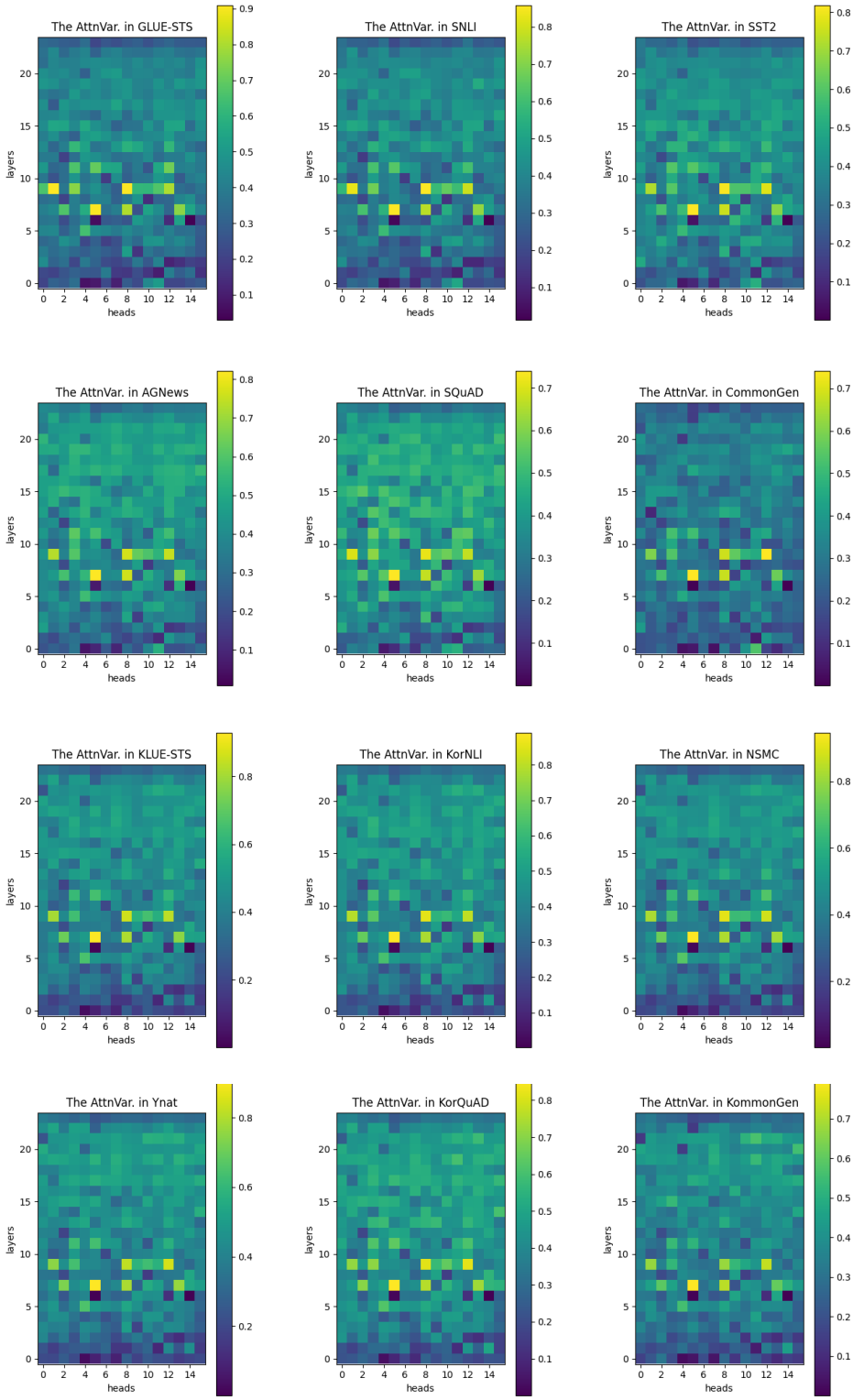

Figure 32 The attention variability for all tasks.

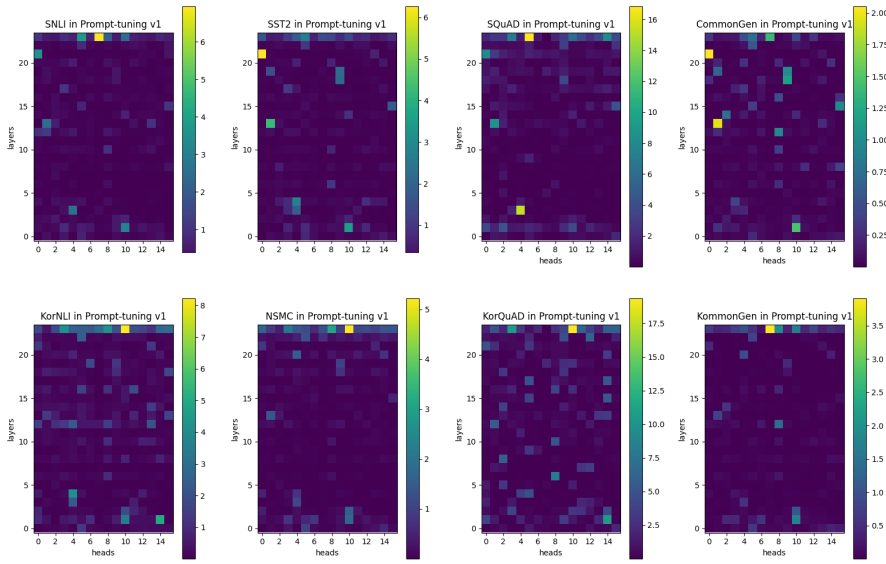

Figure 33 The KL divergence in Prompt-tuning v1.

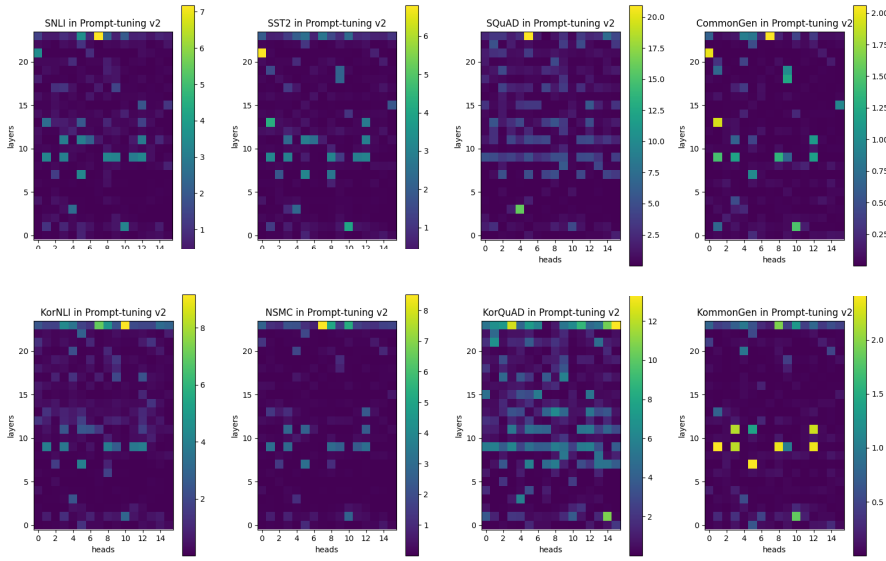

Figure 34 The KL divergence in Prompt-tuning v2.

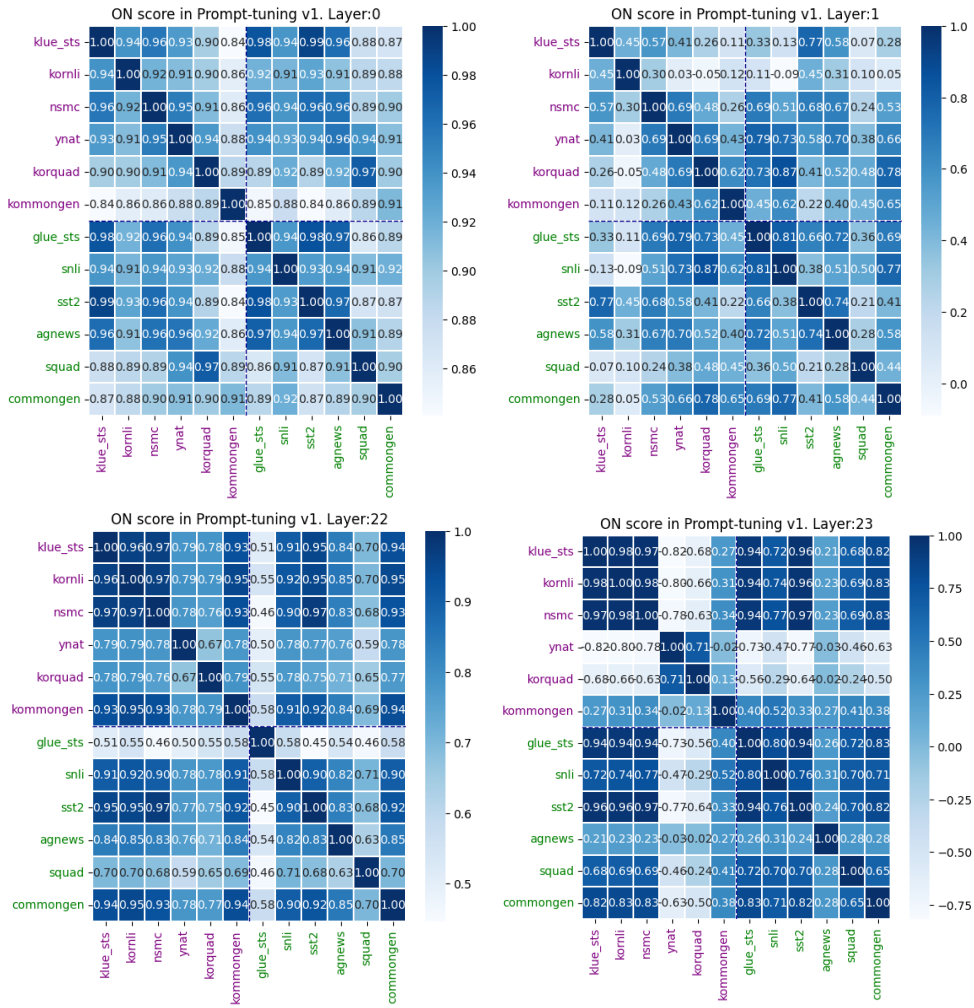

Figure 35 ON score in Prompt-tuning v1 between all tasks.

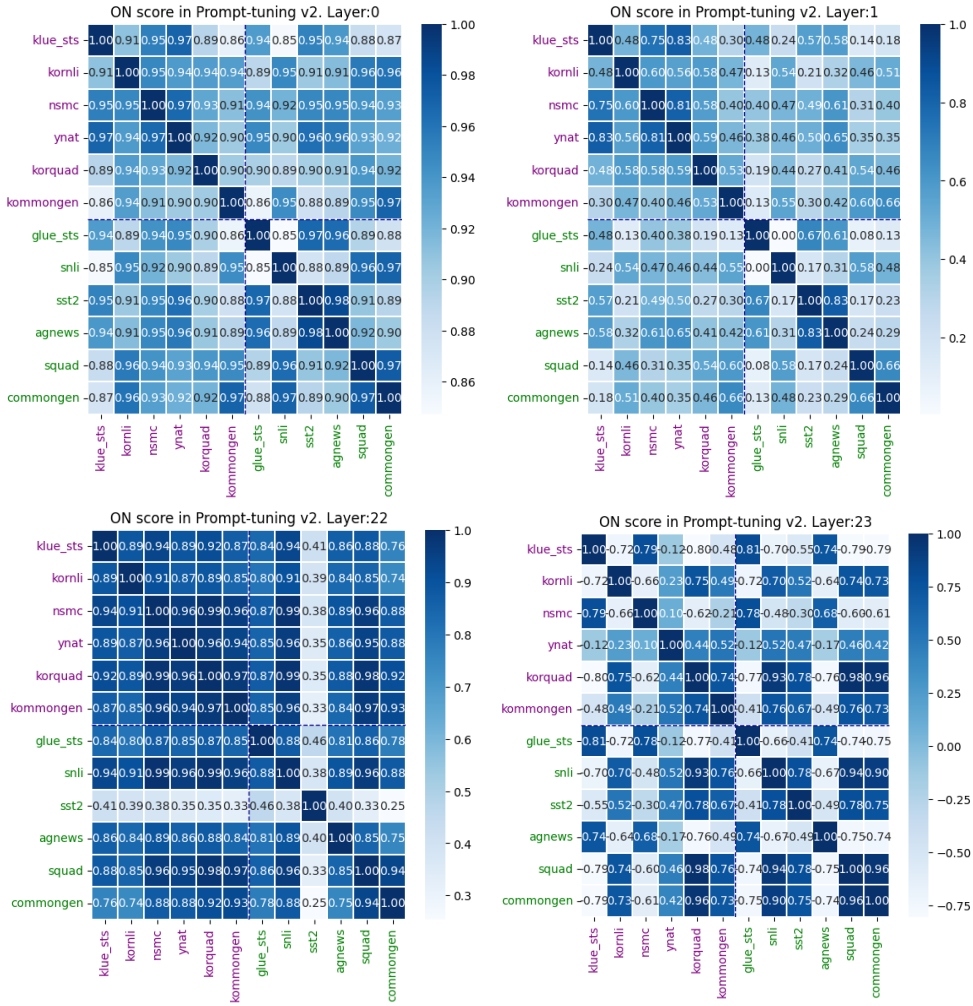

Figure 36 ON score in Prompt-tuning v2 between all tasks.

국문 초록

사전학습 언어 모델을 효과적으로 원하는 태스크에 사용할 수 있는 대표적인 파라미터 효율적인 방법 중 하나는 continuous prompts를 훈련시키는 것이다. 이에 따라 최근에 많은 연구들이 continuous prompts를 활용한 학습 방법을 제안하였다. 그러나, continuous prompts의 설명가능성이 사전학습 언어 모델의 신뢰성을 높이기 위한 중요한 요소임에도 불구하고 이와 관련된 연구는 매우 적다. 따라서 본 연구에서는 continuous prompts의 설명가능성에 대한 문제를 해결하기 위해 Prompt-tuning v1 (Lester et al., 2021)와 Prompt-tuning v2 (Liu et al., 2022)이 사전학습 언어 모델에 미치는 영향을 살펴본다.

이를 위해 본 연구에서는 다국어 언어 모델 GPT를 사용하여 실험하여 태스크와 언어에 대해 일반화 가능성을 모색해보았다. 또한, 본 연구에서는 continuous prompts를 활용한 전이 학습도 진행하였다. 먼저 continuous prompts가 태스크와 언어에 따라 벡터 공간에서 모이는지 조사하고자 하였다. 이어서 continuous prompts가 사전학습 언어 모델을 어떻게 활용하는지를 GPT의 세 가지 주된 구조인 어텐션 메커니즘, 활성화 뉴런, 레이블 공간에 초점을 맞추어 보고자 하였다.

이를 위해 본 연구에서는 다음과 같은 연구 질문을 설정하였다: (1) continuous prompts를 태스크나 언어에 따라 구별할 수 있을까? (2) 프롬프트 튜닝 이후 어텐션 메커니즘의 변화에서 설명가능한 패턴을 발견할 수 있을까? (3) continuous prompts의 활성화 뉴런에서 레이어에 걸친 설명가능한 패턴을 발견할 수 있을까? (4) 사전학습 언어 모델의 레이블 공간과 continuous prompts의 상호작용을 포착할 수 있을까?

첫번째로 continuous prompts가 태스크에 대해 학습한 정보에 따라 구분되는 공간을 가지고 있음을 관찰하였다. 두번째로 continuous prompts가 특히 문맥 의존적인 어텐션 헤드를 활용하면서 사전학습 언어 모델의 어텐션 메커니즘을 활용하고 있음을 관찰하였다. 세번째로 활성화 뉴런은 더 깊은 레이어에서 태스크 특징적인 정보를 담고 있었다. 그런데, 마지막에서 두번째 레이어에서는 오히려 태스크에 공통적인 행동을 보였다. 마지막으로, 사전학습 언어 모델의 등방성이 낮음에도 불구하고 continuous prompts는 임베딩 공간 상에서 디코딩 토큰을 레이블이 아닌 단어보다 레이블인 단어와 더 가깝게 만들고 있었다. 전체적으로 본 관찰 결과들은 전이 학습 이후에도 일관적으로 나타났다.

결과적으로, 본 연구에서는 continuous prompts가 사전학습 언어 모델이 사전학습을 하면서 얻은 지식을 태스크를 해결하는 데 사용하고 있음을 관찰하였다.