JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 29 No. 4, pp. 9-21, April 2024
https://doi.org/10.9708/jksci.2024.29.04.009

Context-Based Prompt Selection Methodology to Enhance Performance in Prompt-Based Learning

Lib Kim*, Namgyu Kim*

*Graduate Student, Graduate School of Business IT, Kookmin University, Seoul, Korea
*Professor, Graduate School of Business IT, Kookmin University, Seoul, Korea

[Abstract]

Deep learning has been developing rapidly in recent years, with many researchers working to utilize large language models in various domains. However, there are practical difficulties that developing and utilizing language models require massive data and high-performance computing resources. Therefore, in-context learning, which utilizes prompts to learn efficiently, has been introduced, but there needs to be clear criteria for effective prompts for learning. In this study, we propose a methodology for enhancing prompt-based learning performance by improving the PET technique, which is one of the contextual learning methods, to select PVPs that are similar to the context of existing data. To evaluate the performance of the proposed methodology, we conducted experiments with 30,100 restaurant review datasets collected from Yelp, an online business review platform. We found that the proposed methodology outperforms traditional PET in all aspects of accuracy, stability, and learning efficiency.

▸ **Key words**: Pre-trained language model, Large language model, In-context learning, Prompt-based learning, PET

[요 약]

최근 딥러닝 분야가 빠르게 발전하는 가운데, 다양한 영역에서 거대 언어 모델을 활용하기 위한 많은 연구들이 진행되고 있다. 하지만 언어 모델의 개발 및 활용을 위해서는 방대한 데이터와 고성능 자원이 필요하다는 현실적인 어려움이 존재한다. 이에 따라 프롬프트를 활용하여 언어 모델을 효율적으로 학습할 수 있는 문맥 내 학습이 등장하였지만, 학습에 효과적인 프롬프트가 무엇인지에 대한 명확한 기준은 구체적으로 제시되지 않았다. 이에 본 연구에서는 문맥 내 학습 방법 중 하나인 PET 기법을 활용하여 기존 데이터의 문맥과 유사한 PVP를 선정하고, 이를 통해 생성한 프롬프트를 학습하여 모델의 성능을 향상시킬 수 있는 프롬프트 기반 학습 성능 향상 방법론을 제안한다. 제안 방법론의 성능 평가를 위해 온라인 비즈니스 리뷰 플랫폼인 Yelp에서 수집된 레스토랑 리뷰 데이터 30,100개로 실험을 수행한 결과, 제안 방법론이 기존의 PET 방법론에 비해 정확도와 안정성, 그리고 학습 효율성의 모든 측면에서 우수한 성능을 보임을 확인하였다.

▸ **주제어**: 사전학습 언어 모델, 거대 언어 모델, 문맥 내 학습, 프롬프트 기반 학습, PET

• First Author: Lib Kim, Corresponding Author: Namgyu Kim
*Lib Kim (fpdkfflq9231@kookmin.ac.kr), Graduate School of Business IT, Kookmin University
*Namgyu Kim (ngkim@kookmin.ac.kr), Graduate School of Business IT, Kookmin University
• Received: 2024. 03. 11, Revised: 2024. 04. 04, Accepted: 2024. 04. 04.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

최근 딥러닝(Deep Learning) 분야가 빠르게 발전하는 가운데, 거대 언어 모델(Large Language Model)을 활용하여 다양한 영역에 걸친 수많은 과제를 해결하기 위한 많은 연구들이 진행되고 있다. 거대 언어 모델이란 기존의 자연어 처리(Natural Language Processing) 분야에서 주로 활용되던 사전학습 언어 모델(Pretrained Language Model)의 확장된 개념으로서, 수십억 개에서 수천억 개의 파라미터로 구성되어 대규모의 데이터를 학습한 초대형 언어 모델이다. 2020년 OpenAI의 실험을 통해 언어 모델의 성능이 모델 크기, 데이터의 양, 그리고 훈련에 사용되는 컴퓨팅 자원과 비례한다는 상관관계가 있음[1]이 밝혀진 이후, GPT-3[2]를 시작으로 Google의 PaLM[3], Meta의 LLaMA[4] 등과 같은 다양한 모델이 등장하면서 딥러닝에서 거대 언어 모델의 영향력이 점차 증가하고 있다. 거대 언어 모델은 광범위한 대규모 언어 데이터를 학습함으로써, 문장의 구조, 문법 및 의미를 매우 정확하게 이해하고 생성한다. 이러한 성능에 힘입어 거대 언어 모델은 텍스트 생성 및 감정 분석과 같은 범용 자연어 처리 과제는 물론, 의료, 금융, 과학 등 여러 특화 분야에서 광범위하게 활용되면서 그 우수성을 입증하고 있다.

하지만 적용 상황에 맞춰 거대 언어 모델을 효과적으로 활용하는 것에는 현실적인 어려움이 존재한다. 첫째, 매우 큰 모델의 크기로 인해 과적합(Overfitting) 또는 과소적합(Underfitting)이 일어날 수 있다. 거대 언어 모델은 수천억 개의 파라미터로 구성되어 있는데, 이러한 모델의 크기는 오히려 학습 과정에서 데이터의 양이나 설정된 하이퍼파라미터에 따라 문제의 원인이 되기도 한다. 둘째, 학습 과정에서 필요로 하는 매우 방대한 양의 데이터를 수집 및 처리하기 위해서는 상당한 양의 자원과 시간이 소모된다. 마지막으로 모델을 다루기 위해서 고성능 컴퓨팅 자원이 필요하다. 거대 언어 모델은 이전의 사전학습 언어 모델보다 훨씬 더 높은 수준의 계산 능력과 방대한 메모리를 요구하므로, 대규모 데이터를 통해 거대한 크기의 모델을 학습하기 위해는 고비용 및 고성능의 자원을 확보해야 한다는 현실적인 어려움이 존재한다.

이러한 문제에 대한 해결책으로 문맥 내 학습(In-Context Learning)이라는 새로운 방법이 등장하였다. GPT-2[5]에서 처음 제안된 이 학습 방법은, 프롬프트 엔지니어링(Prompt Engineering)을 사용하여 사용자의 목적에 맞게 모델을 효과적으로 활용하는 것을 목표로 한다. 문맥 내 학습은 사용자가 하고자 하는 작업에 대한 설명(프롬프트)과 그에 대한 몇 개의 예시로 입력값을 구성하고, 모델이 새로운 입력값의 문맥적인 의미를 이해하여 그에 따른 작업을 수행하도록 한다. 이 방법은 거대 언어 모델이 이미 방대한 데이터로 사전학습이 되었음을 전제로, 모델을 추가적으로 학습하는 것이 아닌 기존 지식을 효과적으로 활용하기 위해 고안되었다. 이러한 이유로 역전파 학습을 통해 모델의 가중치를 갱신하는 과정을 진행하지 않는다. 따라서 문맥 내 학습은 기존의 학습과 달리 추가적인 대규모의 데이터를 필요로 하거나 모델의 가중치 수정을 위한 방대한 작업을 요구하지 않기 때문에, 고성능 컴퓨팅 자원 없이 효율적인 학습이 가능하다는 특징을 갖고 있다.

이러한 접근법은 프롬프트와 함께 주어지는 예시의 수를 기준으로 zero-shot, one-shot 및 few-shot 학습으로 구분되며, 각 분석 목적에 따라 이들 중 적절한 방식을 활용함으로써 모델의 성능을 효과적으로 향상시킬 수 있다. 하지만 이러한 여러 방식에서도 모델이 소수의 예시만 활용하여 사용자의 의도를 정확히 파악하기는 여전히 어렵다는 한계가 드러나면서, 프롬프트를 효과적으로 활용하는 방안에 대한 관심이 높아지게 되었다. 즉 작업에 맞는 적절한 프롬프트를 선정하여 문맥 내 학습을 보다 효율적으로 수행하기 위한 다양한 방법이 모색되고 있으며, 이러한 접근 방식을 프롬프트 기반 학습(Prompt-Based Learning)이라고 한다.

프롬프트 기반 학습에 관한 연구는 프롬프트의 표현 형식을 제안하는 방향으로 특히 많은 시도가 이루어지고 있다. 대표적인 방법으로는 문장 또는 텍스트에서 특정 부분을 공백으로 남겨놓고 해당 공백에 들어갈 맞는 단어를 예측하는 Cloze-style[6], 그리고 특정한 접두사를 붙여 모델에게 어떤 종류의 정보를 생성하거나 요청하는지 알려주는 Prefix-style[7] 등이 있다, 이들 방법은 프롬프트의 구성뿐만 아니라 형식에 대한 변화로도 모델의 성능 향상에 기여할 수 있음을 증명하였다.

하지만 이러한 다양한 시도에도 불구하고, 어떠한 특성을 갖는 프롬프트가 모델의 학습에 효과적인지를 판단하기 위한 명확한 기준은 구체적으로 제시되지 않았다. 따라서 프롬프트 별 학습 성능 평가는, 여러 상이한 프롬프트를 입력으로 학습을 진행한 이후 각 학습의 최종 결괏값 혹은 손실(loss) 값을 확인하는 비효율적인 방식으로 이루어져 왔다. 이러한 불편함을 줄이기 위해 AutoPrompt[8], LM-BFF[9], 그리고 P-tuning[10] 등 모델에 적합한 프롬프트를 자동으로 생성하기 위한 연구들이 진행되었지만, 이러한 방식은 학습에 효과적인 프롬프트 생성을 위해 개

별 모델의 학습이 추가적으로 필요하다는 또 다른 한계를 야기하였다. 이는 한정된 자원과 데이터에의 의존도를 줄이기 위해 제안된 프롬프트 기반 학습의 취지에 어긋난다는 모순을 발생시켰다.

이러한 복합적인 문제들을 해결하기 위해, 최근 고안된 PET(Pattern-Exploiting Training)[11]는 여러 개의 Cloze-style 프롬프트를 무작위로 선정하여 학습을 수행한 후 이를 앙상블(Ensemble)로 통합하는 프롬프트 기반 학습을 제안하였다(Fig. 1). 이 방법은 여러 개의 프롬프트 후보 중 품질이 낮은 프롬프트가 우연히 선정되어 전체 모델의 성능이 저하되는 위험을 줄이는 것에는 기여하였지만, 앙상블에 참여하는 프롬프트 선정에 대한 명확한 기준을 제시하지 않았다는 점에서 여전히 모호성을 갖는다는 한계가 있다.

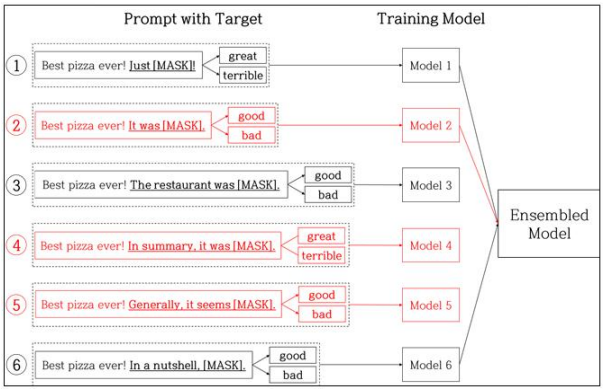

Fig. 1. Ensemble Method of PET

Fig. 1은 PET 기법의 앙상블 방식을 개략적으로 나타낸다. 즉 예측 대상이 되는 목표 값(Target)을 마스킹(Masking)한 뒤, 이를 포함한 다양한 프롬프트를 생성하고 비교하여 앙상블 모델을 생성한다. 본 예에서는 여러 후보 프롬프트의 실제 학습 성능을 확인해 보았을 때, 성능이 좋지 않은 프롬프트가 붉은색의 2번, 4번, 그리고 5번이라고 가정한다. 기존의 PET는 학습에 효과적이지 않은 프롬프트가 우연히 선정되어 모델의 품질을 저하하는 최악의 상황을 방지하는 방안으로, 하나가 아닌 여러 개의 프롬프트를 동시에 선택하고 앙상블하는 방법을 채택하였다. 하지만 이렇게 여러 개를 선택하는 과정은 단순히 무작위로 진행되기 때문에, 특정 프롬프트가 전반적인 앙상블의 성능에 부정적인 영향을 끼칠 가능성이 여전히 존재한다. 예를 들어 본 예에서 우연히 2번, 4번, 그리고 5번의 프롬프트가 앙상블 구성 프롬프트로 선정되고 1번, 3번, 그리고 6번 프롬프트가 제외된다면 저품질 프롬프트로 구성된 앙상블 모델의 성능도 역시 높지 않게 나타날 것으로 예상할 수 있다.

이에 본 연구에서는 기존의 PET 기법의 성능을 향상시키기 위해, 기존의 무작위 방식이 아닌 프롬프트의 문맥을 기준으로 학습 후보를 선정하는 프롬프트 기반 학습 방법론을 제안하고자 한다. 구체적으로 본 연구에서는 기존 학습 데이터와 프롬프트 간의 유사성에 기반을 두어 최종 앙상블의 후보 모델을 선정하는 방안을 제시하고, 제안 방법론을 통해 PET의 성능을 향상시킬 수 있음을 확인하고자 한다.

본 논문의 이후 구성은 다음과 같다. 우선 다음 2장에서는 본 연구와 관련된 선행 연구를 소개하고, 3장에서는 본 연구에서 제안하는 문맥 중심 프롬프트 선정을 통한 프롬프트 기반 학습 방법론을 소개한다. 4장에서는 제안 방법론과 전통적인 PET 방법론과의 성능을 비교하고, 마지막 5장에서는 본 연구의 기여와 한계를 정리한다.

II. Preliminaries

1. Pre-trained language model

사전학습 언어 모델이란 인간이 지식을 습득하는 방식처럼 방대한 텍스트 데이터에서 언어의 문법 및 의미를 학습하는 모델이다. 사전학습 언어 모델을 활용하여 생성, 분류, 그리고 요약 등과 같이 다양한 작업을 범용적으로 수행하기 위해서는, 각 작업에 맞는 방식으로 미세조정(Fine-tuning)[12]이 이루어져야 한다. 사전학습 언어 모델은 미리 대규모 데이터로 학습되어 기존의 언어 모델보다 더 많은 범용 지식을 습득하였기 때문에, 미세조정 시에는 비교적 적은 데이터만 사용하더라도 작업을 수행하는 데 있어 우수한 성능을 보일 수 있는 장점이 있다. 대표적인 사전학습 언어 모델로는 BERT(Bidrectional Encoder Representations from Transformers)[13]와 GPT(Generative Pre-trained Transformer)[14] 등이 있다.

BERT는 자연어 처리 분야에서 빼놓을 수 없는 대표적인 사전학습 언어 모델이다. Google에서 개발한 이 언어 모델은 Transformer[15]의 Encoder 구조를 활용하였으며, 문장의 앞뒤 단어 모두를 고려하여 문맥을 이해한다. 또한 BERT는 문장 내 일정 비율의 단어를 무작위로 선택 및 마스킹한 뒤 주변 단어들의 맥락을 바탕으로 해당 단어를 예측하는 MLM(Masked Language Model) 학습을 통해 기존 다른 언어 모델보다 문맥을 잘 이해하며, 특히 텍스트 분류, 질의응답, 그리고 감정 분석과 같은 다양한 자연어 관련 작업에서 높은 성능을 발휘하고 있다. 이로 인

해 BERT는 사전학습 언어 모델의 대표적인 모델로 인식되고 있으며, 이후의 언어 모델 개발에도 많은 영향을 끼치고 있다.

BERT에서부터 파생된 모델로는 학습 계산 방식을 개선하여 메모리 효율성을 높인 ALBERT[16], 동적 마스킹(Dynamic Masking) 학습법을 활용하여 성능을 개선한 RoBERTa[17], 그리고 새로운 RTD(Replaced Token Detection) 학습법으로 학습의 효율성을 높인 ELECTRA[18] 등이 있다. 또한 생물학 자료를 통해 사전학습된 BioBERT[19], 금융 관련 뉴스와 보고서로 사전학습된 FinBERT[20], 그리고 법률 문서로 사전학습된 LegalBERT[21] 등 사전학습 모델을 추가로 학습시켜 각 도메인에 특화된 사전학습 모델을 생성하기도 한다. 이처럼 BERT 기반 모델은 다양한 방법으로 개선이 이루어지고 있다.

또 다른 대표적인 사전학습 모델은 GPT이다. OpenAI에서 개발한 이 모델은 BERT와 다르게 Transformer의 Decoder 구조를 기반으로 구성되어, 문장을 생성하는 데에 효과적인 사전학습 모델이다. GPT는 자동 회귀 언어 모델(Autoregressive Language Model)로, 문장 생성 성능의 향상을 위해 문맥 내의 이전 텍스트 의미를 바탕으로 다음 단어를 예측하는 훈련을 수행한다. 이를 통해 GPT는 문맥에 맞는 자연스러운 문장들을 생성할 수 있는 것으로 알려져 있다. 또한 GPT는 방대한 양의 다양한 텍스트 데이터로 사전학습되기 때문에, 언어의 광범위한 측면을 이해할 수 있다는 장점을 갖는다. 이러한 특징들을 통해 GPT는 다양한 자연어 분석 작업에 활용되었으며, 이후 GPT-2, GPT-3, 그리고 GPT-4[22] 등 대규모의 데이터와 새로운 학습법으로 꾸준히 개선되고 있다.

2. Large Language Model

거대 언어 모델(Large Language Model)은 기존의 언어 모델(Language Model)을 확장한 개념이다. 모델의 성능 향상이 모델 크기, 학습 데이터의 양, 그리고 컴퓨팅 자원의 양에 비례하여 이루어짐이 증명되면서, GPT-3, Palm, LLaMA, 그리고 Chinchilla[23]와 같은 다양한 거대 언어 모델들[24]이 빠르게 등장했다. 거대 언어 모델은 광범위한 매개 변수와 방대한 데이터로 학습함으로써, 여러 자연어 처리 분야 작업에서 일반화 능력이 높다는 특징을 갖는다. 최소 수 백억 개의 매개 변수로 이루어져 복잡한 언어의 규칙과 문맥을 파악하고 이해할 수 있으며, 방대한 데이터를 학습해 얻은 범용적인 지식으로 자연어 분석의 전반적인 작업들에 대해 매우 높은 성능을 보인다.

우수한 성능의 거대 언어 모델을 개발하기 위해서는 엄청난 계산 시간, 에너지 비용, 그리고 컴퓨팅 자원이 필요하다. 하지만 각 조직이 자유롭게 거대 언어 모델을 개발하거나 활용할 수 있는 이러한 조건들을 갖추는 것은 현실적으로 어려움이 있다. 이러한 이유로 최근 계산의 효율성을 높여 자원이 제한된 환경에서도 거대 언어 모델을 운용할 수 있는 여러 연구가 활발하게 이루어지고 있다. 구체적으로 PEFT의 T-few[25] 또는 LoRA[26] 같이 상대적으로 적은 양의 매개 변수만을 학습하거나 문맥 내 학습을 통하여 추가적인 학습 없이 프롬프트만을 활용하는 등, 적은 자원으로 거대 언어 모델을 최대한 효율적으로 운용할 수 있는 새로운 방법들이 등장하고 있다.

3. In-Context Learning

언어 모델을 사용자의 목적에 맞게 활용하기 위한 지금까지의 연구는 대부분 모든 매개 변수의 값을 미세조정하는 방식으로 이루어졌다. 하지만 거대 언어 모델의 크기를 생각하면, 이전과 동일한 방법으로 미세조정을 수행하기 위해서는 막대한 시간, 자원, 그리고 비용이 필요함을 알 수 있다. 따라서 최근 이러한 한계점을 극복하는 다양한 방법들이 등장하였는데, 프롬프트 엔지니어링을 통해 기존 모델의 지식을 효과적으로 활용하는 문맥 내 학습[27]이 가장 대표적인 예이다.

문맥 내 학습은 거대 언어 모델이 방대한 데이터를 통해 학습되어 이미 일반적인 언어 모델보다 훨씬 더 많은 지식을 가지고 있다는 것을 전제로 고안되었다. 따라서 학습 시 매개 변수를 미세조정하지 않고 프롬프트와 예시만으로 최적화된 결과를 도출한다. 즉, 프롬프트와 소수의 예시를 통해 사용자의 활용 목적에 대한 명확한 정보를 전달받고, 이미 학습한 지식으로 목적에 맞는 작업을 수행[28]한다.

문맥 내 학습은 새로운 데이터를 학습하지 않는다. 즉, 기존 언어모델의 학습 과정과 같이 손실 값 계산 및 역전파를 하지 않으며, 모델의 가중치를 미세조정하지도 않는다. 대신 적절한 프롬프트와 명확한 예시로 새로운 작업과 도메인에 대한 이해를 높여 일반화 능력을 향상시킨다. 이러한 접근은 시간, 자원, 그리고 비용적인 측면에서 매우 효율적임을 보여주었으며, 성능 면에서도 효과적임을 증명하였다[32, 33].

문맥 내 학습의 방식은 프롬프트와 함께 주어지는 예시의 수에 따라 예시 없이 프롬프트만으로 학습하는 제로-샷(Zero-Shot)[29], 단 하나의 예시가 주어지는 원-샷(One-Shot)[30], 그리고 둘 이상의 예시로 학습하는 퓨-샷(Few-Shot)[31]으로 구분된다. 또한 문맥 내 학습의 효

과를 극대화하기 위해서는 적은 예시들로도 모델이 작업에 맞는 원하는 결과를 도출할 수 있도록 최적화된 프롬프트를 설계하는 것이 매우 중요하며, 최근 이에 대한 많은 연구들[34, 35, 36]이 진행되고 있다. 이처럼 모델의 학습 및 일반화 성능 향상을 위해 모델에게 작업에 맞는 정보를 정확히 전달하는 프롬프트의 역할은 더욱 강조되고 있다.

III. The Proposed Method

1. Research Process

본 장에서는 학습 데이터와 문맥이 유사한 PVP(Pattern Verbalizer Pair)[11]를 선정하여 프롬프트 기반 학습의 성능을 향상시키기 위한 방법론을 소개하고, 단계별 구체적인 프로세스를 설명한다. 제안 방법론의 전체적인 과정은 Fig. 2와 같다.

먼저 Phase 1은 기존 패턴(Seed Patterns)과 생성형 언어 모델을 활용하여 형태가 유사한 새로운 후보 PVP(Candidate PVPs)를 생성하고(①), 학습용 데이터에 대한 전처리 과정을 거친다(②). 이후 각 후보와 전처리 된 데이터를 사전학습 언어 모델로 임베딩하여 벡터를 생성한 뒤(③), 각 후보 PVP의 임베딩 벡터와 전처리 된 학습용 데이터의 평균 벡터 사이의 코사인 유사도를 비교하여 데이터와 가장 유사한 참여 PVP(Participant PVPs)를 선정한다(④). 최종 Phase 2에서는 앞의 과정에서 선정된 PVP 들과 데이터를 합쳐 입력 프롬프트로 만든 뒤, 프롬프트별 모델을 미세 조정하여(⑤), 전체 모델들을 앙상블을 진행한다(⑥). 각 단계에 대한 세부적인 프로세스는 다음 절에서 설명하며, 실제 데이터를 적용한 제안 방법론의 성능 평가는 4장에서 소개한다.

2. Generation and Embedding of Candidate PVPs

본 절에서는 Fig. 2의 단계 중 Phase 1에서 이루어지는 과정, 즉 PVP 후보의 추가 생성(①), 학습 데이터의 전처리(②), 그리고 사전학습 언어 모델을 이용한 각 PVP 들과 학습 데이터 임베딩(③)을 소개한다.

PVP란 패턴(Pattern)과 버벌라이저(Verbalizer)의 쌍으로, 입력 데이터의 특징을 추출하는 역할을 하는 문장을 의미한다. PVP를 통해 초기 작업(Origin Task)을 클로즈 테스트(Cloze Test)로 변경하고 마스킹 된 부분에 들어갈 알맞은 토큰(버벌라이저)를 찾는 방식으로 학습의 성능을 향상시키는 방법은 기존 연구인 PET에서 이미 제안된 바 있

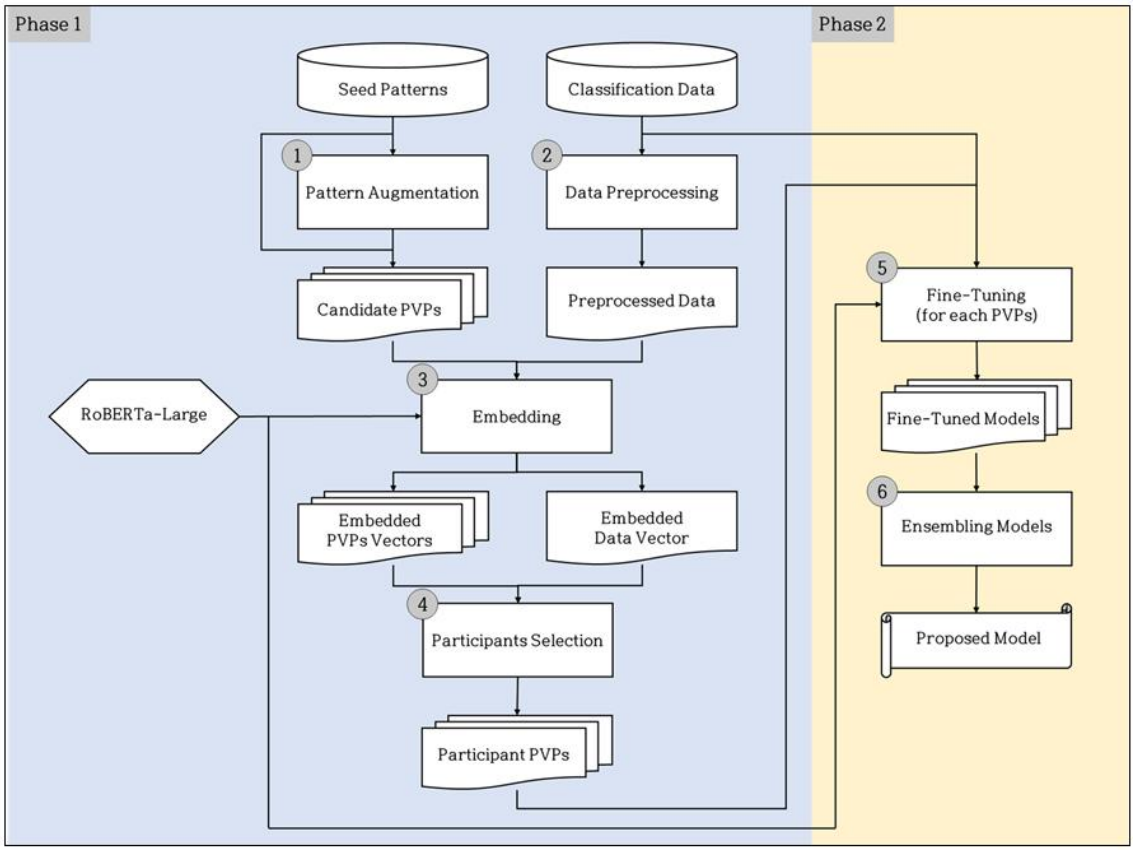

Fig. 2. Overall Research Process

다. 이때 Fig.3과 같이 데이터의 특징을 추출하기 위해 추가된 클로즈 형식(Cloze-Style)의 문장을 패턴, 그리고 기존 라벨을 단어화한 정답 토큰을 버벌라이저라고 한다. 또한 이러한 PVP와 기존 데이터를 합친 모델의 입력값을 프롬프트라고 지칭한다. 즉 프롬프트는 새로운 입력값이 되어 모델의 MLM(Masked language Modeling) 작업에 사용된다.

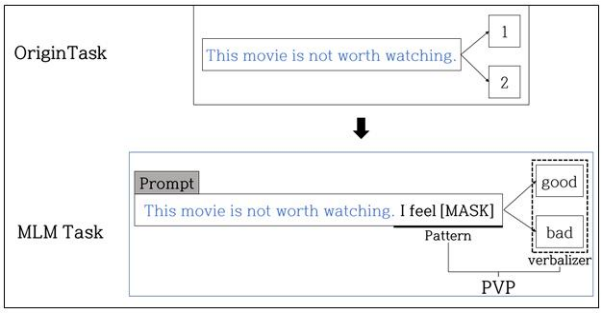

Fig. 3. Example of Prompt and PVP

Phase 1의 첫 단계에서는 기존 연구에서 사용했던 패턴을 시드 패턴(Seed Pattern)으로 설정하고, 생성형 언어 모델을 통해 시드 패턴과 상이한 형태를 갖는 더 많은 패턴을 생성한다(Fig. 4). 이후 모든 패턴을 종합하여 PVP 후보군을 생성한다(①).

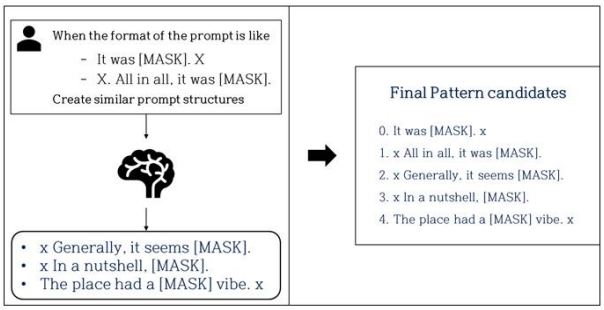

Fig. 4. Example of Pattern Augmentation

또한 데이터 전처리 과정을 통해 분류 학습용 데이터를 언어 모델의 입력에 적합한 형태로 가공한다. 본 연구에서 사용되는 비정형 텍스트 데이터의 경우 구조화되지 않고 통일된 형식이 없는 동시에, 구두점이나 특수문자 등 다양한 종류의 토큰들을 포함하고 있다. 이러한 특징들은 불필요한 노이즈로 인해 모델이 특정 토큰에 과적합 되는 등 학습 성능을 저하시키는 요인으로 작용한다. 따라서 필요하지 않은 토큰을 사전에 제거하고 일련의 정제 과정을 진행함으로써, 언어 모델이 텍스트의 본질을 더 잘 파악할 수 있도록 한다. 구체적으로 본 단계에서는 불필요한 특수문자, 중복 문자, 그리고 공백 등을 제거한다(②).

이후 앞선 단계(①, ②)에서 추출한 PVP 후보와 정제된 데이터를 컴퓨터가 이해할 수 있는 벡터로 변환시키기 위해 임베딩을 수행한다. 텍스트 임베딩에는 Word2vec[37], GloVe[38], 그리고 언어 모델 등 매우 다양한 방법들이 존재한다. 본 연구에서는 그 중 방대한 데이터와 다양한 방식으로 학습하여 최근까지도 좋은 성능을 보이는 사전학습 언어 모델을 활용하여 임베딩 벡터를 추출한다(③).

3. Selection of Participant PVPs

본 절에서는 Fig. 2의 Phase 1에서 이루어지는 과정 중 최종 PVP를 선정하는 과정(④)을 소개한다. PET 기법은 학습을 수행하기 전에는 어떤 PVP로 구성된 프롬프트를 사용하는 것이 모델의 성능을 높일 수 있는지 명확히 알 수 없다는 한계를 갖는다. 기본적으로는, 학습에 효과적인 프롬프트를 찾기 위해 각 PVP로 구성된 프롬프트로 독립적인 학습을 진행한 후 각 모델의 성능을 비교하여 양질의 프롬프트를 확인해야 한다. 하지만 이 방법은 많은 자원과 시간이 소요되기 때문에 매우 비효율적이다.

따라서 PET는 이러한 난제를 해결하기 위해, 무작위로 복수의 PVP를 선정하여 PVP별로 프롬프트를 구성하고 이들 각각을 학습한 모델을 앙상블 하는 방법론을 제안하였다. 이 방법은 앙상블을 통해 최종 모델에 대해 안정적인 성능을 기대할 수 있다는 장점이 있지만, 동시에 특정한 기준 없이 무작위로 PVP를 선정함으로써 성능이 좋지 않은 모델만으로 앙상블이 구성될 가능성이 존재한다는 한계를 갖는다. 따라서 본 연구에서는 일반화된 성능을 기대할 수 있는 앙상블 기법은 유지하되, 무작위로 PVP를 선정하는 대신 원본 입력 문장과 PVP 간의 유사도를 기준으로 PVP를 선정하여 모델의 성능을 개선하는 방안을 제시한다.

최근 다양한 분야에서 활약하고 있는 언어 모델의 대부분은 셀프 어텐션(Self-Attention)[15] 메커니즘을 핵심 구성 요소로 하는 트랜스포머 기반의 모델이다. 셀프 어텐션은 쿼리(Query), 키(Key), 그리고 밸류(Value) 벡터를 활용하여 유사도를 기반으로 어텐션 점수(Attention Score)를 계산하는 메커니즘으로, 이를 통해 입력 문장 내 모든 단어의 연관성과 중요성을 동시에 평가한다. 트랜스포머 기반 모델과 관련한 최근 연구에서, 유사한 문맥의 문장들로 입력값을 구성할수록 모델이 효과적으로 입력값을 분석하고 이해할 수 있을 것이라는 가설을 제시하였다. 즉 입력값의 문장들이 유사한 문맥을 공유함으로써, 모델이 입력값을 일관된 방식으로 해석하고 모델의 성능을 향상시킬 수 있음을 보였다[39]. 또한 무작위로 프롬프트를 생성하는 것이 언어 모델의 해석력을 저하시키며, 기존 예

제와 의미적으로 유사한 패턴으로 프롬프트를 생성함으로써 모델의 성능을 향상시킬 수 있음을 보인 연구[40, 41]도 진행된 바 있다.

이처럼 프롬프트를 이루는 구성 요소들의 유사함이 프롬프트의 파악에 영향을 줄 수 있다는 기존 연구 결과에 기반을 두어, 본 연구에서는 유사도를 기반으로 PVP를 선정하여 모델의 성능을 향상시키고자 한다. 하나의 PVP와 전체 입력 문장과의 유사도를 산출하는 과정은 Fig. 5와 같다.

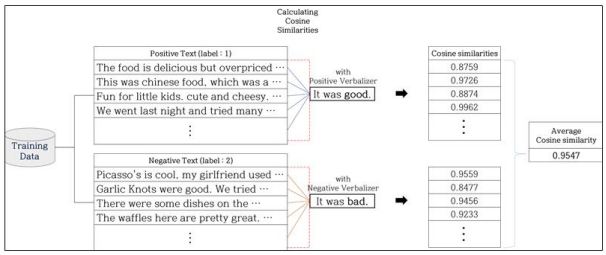

Fig. 5. Example of Similarity Calculation

Fig. 5에서 Training Data는 각 라벨에 따라 두 묶음으로 구분되어 있다. 먼저 데이터의 라벨과 후보 PVP의 감정 표현을 맞춰 긍정 라벨의 데이터는 긍정 버벌라이저를 포함한 PVP와, 부정 라벨의 데이터는 부정 버벌라이저를 포함한 PVP와의 코사인 유사도를 계산한다. 코사인 유사도는 벡터로 표현된 두 문장 간 각도의 코사인 값으로 계산되며, 비교 문장 간의 길이에 차이가 있어도 안정적으로 계산할 수 있어 고차원 공간에서의 벡터 간 유사도 계산에 효과적으로 활용된다. 본 방법론에서는 기존 데이터와 PVP 간 길이의 차이로 인한 유사도 왜곡 효과를 최소화하기 위해 코사인 유사도를 활용하여 유사도를 측정한다. 이렇게 각 라벨 별 데이터와 후보 PVP의 유사도를 측정하고, 그 값들을 평균 내어 해당 PVP의 유사도 점수를 산출한다. 이러한 과정은 Fig. 6과 같이 각 후보 PVP 별로 진행되며, 최종 평균 유사도가 높은 상위의 PVP를 선정하여 PET 학습을 진행할 프롬프트를 구성한다.

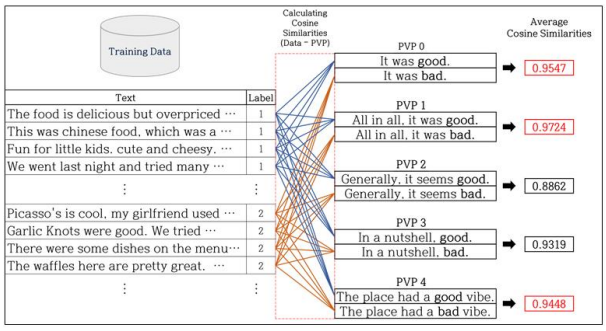

Fig. 6. Example of Participants Selection

4. Fine-Tuning and Ensembling Models

본 절에서는 제안 방법론(Fig. 2)의 마지막 단계인 Phase 2에서 이루어지는 미세조정(⑤)과 앙상블(⑥) 과정을 소개한다. 미세조정은 MLM 학습을 통해 이루어지는데, 먼저 X를 원본 데이터의 집합으로 정의하고 버벌라이저(v)를 라벨(l)로 매핑하여 각각의 $x \in X$에 대해 변환 함수 P(x)를 적용한다. 이를 통해 Fig. 7과 같이 앞의 과정(④)에서 선정된 PVP를 새로운 프롬프트와 라벨로 변형하고, 미세조정 학습의 데이터로 지정한다.

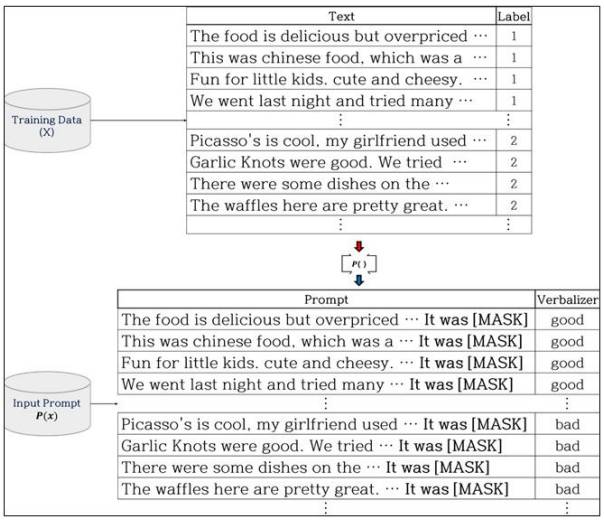

Fig. 7. Example of Transforming Data to Prompts

이후 P(x)를 언어 모델의 입력값으로 사용하여 라벨의 값을 나타내는 s_p를 출력하고, 소프트맥스(Softmax) 함수를 통해 정규화하여 확률 분포인 q_p로 변환한다. 마지막으로 q_p와 실제 라벨 값을 비교하여 크로스-엔트로피 손실 값(Cross-Entropy Loss)을 계산하고, 이를 통하여 언어 모델을 미세조정한다. 이러한 과정을 선정된 PVP 별로 진행하여 서로 다른 모델들을 획득하며, 그 과정은 Fig. 8과 같다(⑤).

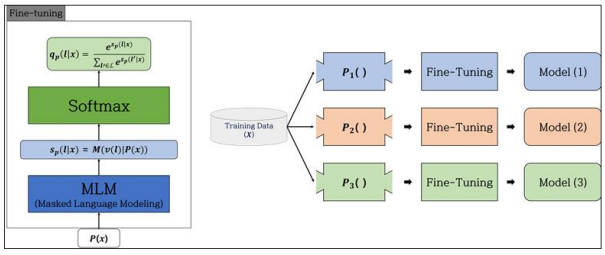

Fig. 8. Multiple Fine-Tuning for Each PVP

이후, 최종 결괏값을 얻기 위해, 각 프롬프트에 대해 미세조정된 모델을 앙상블 한다. 본 연구에서는 여러 앙상블 기법 중 기존 연구와 동일한 가중 평가 방식을 통해 앙상블을 수행하였고, 그 과정은 Fig. 9와 같다.

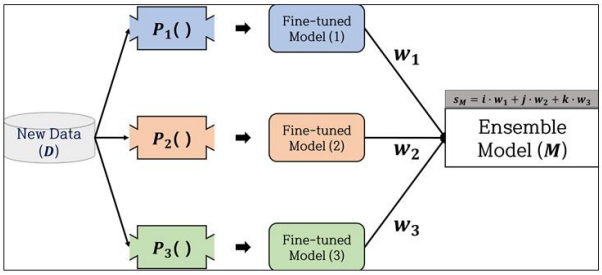

Fig. 9. Example of Ensembleing Models

먼저 평가 데이터에서 나타난 각 모델의 성능을 기반으로 하여 가중치를 계산한다. 예를 들어 미세조정한 모델들의 성능을 각각 i,j,k라고 했을 때, 각 가중치 w는 (각 모델의 성능)/$(i+j+k)$으로 계산한다. 이와 같은 방식으로 각 모델의 성능에 기반을 두어 가중치를 계산하고, 최종 출력에 대한 기여를 조절하여 앙상블 모델(M)의 최종 스코어를 도출한다. 이를 통해 앙상블 모델의 성능에 대한 각기 다른 모델의 기여도를 반영하며 안정적인 결과를 얻을 수 있다.(⑥)

IV. Experiment

1. Experiment Overview

본 장에서는 앞서 소개한 제안 방법론을 실제 데이터에 적용한 실험 결과 및 성능 분석 결과를 소개한다. 실험에는 온라인 비즈니스 리뷰 플랫폼인 Yelp에서 수집된 레스토랑 리뷰 데이터 셋[42] 중 'yelp_polarity' 데이터를 사용하였다. 본 데이터 셋은 부정(1) 및 긍정(2)의 레이블로 이루어진 텍스트 분류 데이터이며, 560,000개의 학습 데이터와 38,000개의 평가 데이터로 구성되어 있다. 본 실험 환경은 Python 3.9를 통해 구축하였으며, 구체적인 H/W 및 S/W 환경은 Table 1과 같다. 또한 성능 비교 실험의 전체 프로세스는 Fig. 10과 같다.

Table 1. System Environment

HW	CPU	16 core 2.1Ghz
	GPU	56TF(NVIDIA V100×4, 128B)
SW	Memory	128GB
	Python	3.9.16
	Pytorch	1.13.1+cu116

Fig. 10의 (A)는 제안 방법론을 통해 학습된 모델을 평가하는 과정으로, 기존 데이터와 비교하여 코사인 유사도가 가장 높은 상위 4개의 PVP를 선정하고 이를 통해 PET 과정을 진행한 최종 모델의 분류 정확도(Classification Accuracy)를 측정한다. 또한 (B), (C), (D), 그리고 (E)는

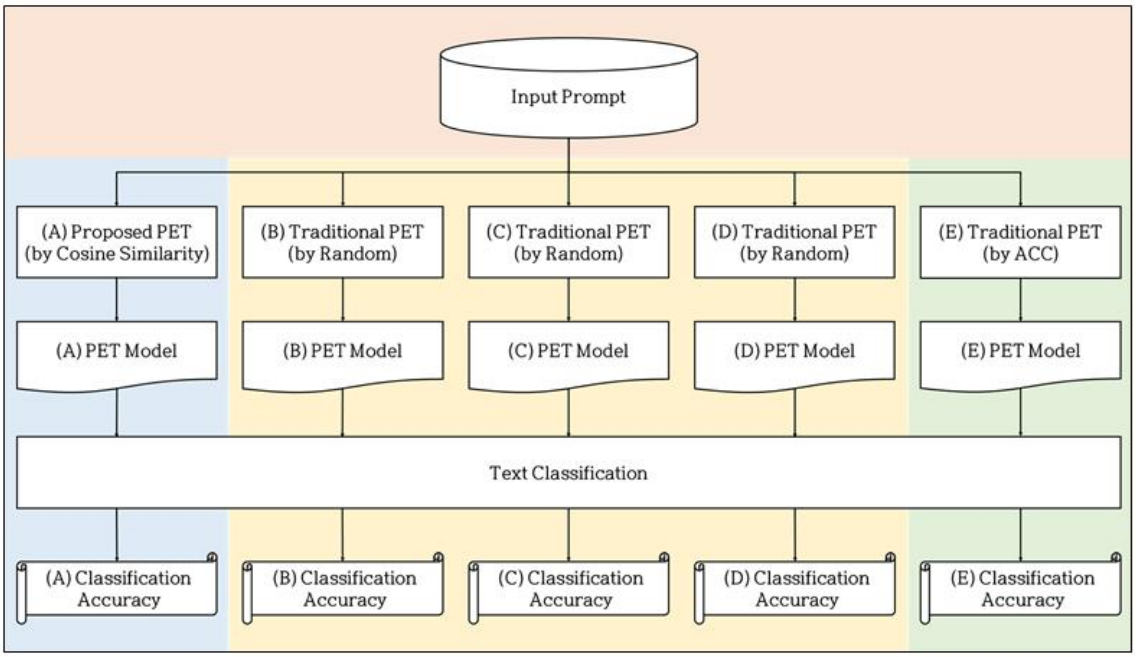

Fig. 10. Overall Process of Performance Evaluation

제안 방법론과의 상대적인 성능 비교를 위해 수행한 모델이다. (B) ~ (D)는 기존 PET 학습을 수행한 모델로, 무작위로 PVP를 선정한 3가지 모델 각각의 성능을 확인한다. 마지막 (E)는 PVP별로 미세조정을 한 뒤, 분류 정확도가 가장 높은 모델들로 앙상블 한 모델을 의미한다.

2. Results of Generation and Embedding of Candidate PVPs

본 절에서는 PVP 후보를 생성하고 기존 입력 데이터에 대해 전처리를 수행한 후, 각 데이터에 대한 임베딩 벡터를 획득하는 과정과 결과를 소개한다. 먼저 기존 PET 논문에서 사용된 패턴을 chatGPT의 입력 예시로 활용하여, PVP 후보로 사용할 추가 패턴을 증강을 통해 생성한다. 본 실험에서는 기존 4개의 패턴과 추가적으로 생성한 9개의 패턴을 합쳐 총 13개의 패턴을 사용한다. Table 2는 본 실험에서 사용할 모든 패턴 후보를 나타낸다. 각 PVP의 [MASK]에 들어가는 버벌라이저로는 부정(0)과 긍정(1)에 각각 'bad'와 'good'을 사용하였다.

Table 2. Result of Pattern Augmentation

	No.	Pattern
Base Pattern	1	It was [MASK]. x
	2	x All in all, it was [MASK].
	3	x Just [MASK]!
	4	x In summary, the restaurant is [MASK].
Generated Pattern	5	x Taking everything into account, it was [MASK].
	6	x To wrap it up, it was [MASK].
	7	x Generally, it seems [MASK].
	8	x In a nutshell, [MASK].
	9	The place had a [MASK] vibe. x
	10	Everything gave off a [MASK] impression.x
	11	x Broadly speaking, it appears [MASK].
	12	x Summing it all up, the restaurant appears [MASK].
	13	x To conclude, the feeling is [MASK].

또한 사용할 데이터 일부를 전처리하여 새로운 데이터셋을 구축한다. 구체적으로 먼저 대문자 치환, 특수 문자 제거, 그리고 Null 값 및 중복 공백 제거 등의 과정을 거친다. 이후 문장 길이가 데이터와 PVP 간의 코사인 유사도에 미치는 영향을 줄이기 위해, 300자에서 400자 사이의 문장으로만 한정하여 실험 데이터 셋을 구축하였다. 이러한 과정을 거친 데이터 중 파인튜닝에 사용할 훈련용 10,000개, 검증용 100개, 평가용 10,000개, 그리고 최종 PET 모델을 평가할 10,000개를 무작위로 선정하여 총 30,100개의 데이터로 실험을 진행하였다. 이후 PVP의 [MASK] 자리에 'good'과 'bad'의 버벌라이저를 삽입하여 구분하고, RoBERTa-Large를 활용하여 이를 각 데이터의 문장들과 함께 1024차원 벡터로 임베딩한다. Fig. 11은 본 과정을 모두 마친 결과의 예이다.

	Text	Embedding Vector	Label
Data	i haven't gone on a weekday. but omg ...	[-0.49161455, 0.00937985 ... -0.42132816, 0.33149087]	0
	meh we ordered this for room service ...	[-0.2866585 , -0.08225754 ... -0.09439274, 0.25233802]	0
	⋮	⋮	⋮
	been here about 4x now close to my work ...	[-0.45306563, 0.01464486 ... -0.42401642, 0.17359558],	1
	just ordered 2 club sandwiches for pickup ...	[-0.17389078, 0.0071531 ... 0.10591157, 0.13888127]	1
PVP1	It was bad.	[-0.13328248, -0.19338389 ... 0.11074048, 0.17279987]	0
	It was good.	[-0.12902051, -0.21542926 ... 0.0867081 , 0.2061094]	1
PVP2	All in all, it was bad.	[-0.18373042, -0.1782788 ... 0.06201676, 0.21077672]	0
	All in all, it was good.	[-0.13576847, -0.25907117 ... 0.07311352, 0.26262787]	1
⋮	⋮	⋮	⋮
PVP13	Everything gave off a bad impression.	[-0.17378566, -0.22074927 ... 0.03281051, 0.18878801]	0
	Everything gave off a good impression.	[-0.1681392 , -0.23617451 ... 0.02667564, 0.24621701]	1

Fig. 11. Example of Preprocessed Data

3. Selection of Participant PVPs

본 절에서는 앞에서 생성한 각 후보 PVP 중 미세조정에 활용될 참여 PVP를 선정하는 과정을 소개한다. 먼저 본 연구에서 제안한 유사도 기반 PVP 선정 모델, 즉 Fig. 10의 모델 (A)에 대해 소개한다. 구체적으로 각 PVP 별 데이터와의 코사인 유사도 비교를 통해 기존 데이터와 가장 유사한 PVP로 참여 그룹을 구성한다. 이를 위해, 이전 과정에서 임베딩한 PVP들과 데이터들을 각자의 라벨에 맞추어 한 쌍으로 매칭한다. 이후 매칭된 쌍 내의 PVP와 데이터 간의 코사인 유사도를 계산한 뒤, PVP별 평균 코사인 유사도를 산출한다. Table 3은 본 과정의 결과를 코사인 유사도 기준 내림차순으로 정리한 것이다. 이 결과에 따라 제안 모델은 이 중 상위 4개의 PVP(5번, 12번, 6번, 그리고 13번 PVP)를 참여 PVP로 선정한다.

Table 3. Cosine Similarity of Each PVP

Rank	PVP	Average Cosine Similarity
1	**PVP 5**	0.9792
2	**PVP 12**	0.9696
3	**PVP 6**	0.9658
4	**PVP 13**	0.9603
5	PVP 11	0.9575
6	PVP 4	0.9561
7	PVP 2	0.9554
8	PVP 9	0.955
9	PVP 8	0.9549
10	PVP 10	0.9546
11	PVP 7	0.9543
12	PVP 1	0.9536
13	PVP 3	0.8898

한편 Fig. 10의 모델 (B) ~ (D)는 기존 PET를 적용한 실험으로, 각 모델은 13개의 프롬프트 가운데 무작위로 4개의 참여 PVP를 선정한다. 무작위 방식 PVP 선정 모델의 성능을 안정적으로 평가하기 위해, 본 실험에서는 무작위 모델 3개를 개별적으로 구축하여 이들의 평균 성능을 측정한다. 마지막으로 모델 (E)는 모델 (A) ~ (D)와 달리, 참여 PVP를 미세 조정 이후에 선정한다. 해당 모델은 우선 전체 PVP를 사용하여 모델들을 각각 미세조정한 후, 평가 데이터에 대한 분류 성능이 가장 높게 나타난 상위 4개의 모델의 PVP를 참여 PVP로 선정한다. 즉 성능이 우수하게 나타나는 개별 모델들을 선별적으로 앙상블 하여 최적 모델을 구축하기 위한 전략이다. 따라서 본 모델은 다른 모델들에 비해 많은 시간과 자원을 요구한다.

4. Fine-Tuned and Ensemble Models

본 절에서는 선정된 참여 PVP들과 기존 데이터로 새로운 입력 프롬프트를 구성하고, 이에 대한 미세조정 및 앙상블을 통해 최종 PET 모델을 형성하는 과정을 소개한다. 먼저 PVP와 전처리한 데이터를 활용하여 미세조정을 위한 입력 프롬프트를 새롭게 구성한다. 앞서 유사도 비교를 위해 매칭한 PVP와 데이터 쌍을 합쳐 하나의 입력값으로 변형한다. 이와 같이 PVP 별 프롬프트를 생성한 뒤, 이를 입력으로 사용하여 각 모델을 미세조정한다. 각 PVP 별 미세조정에 사용한 모델과 실험 구성은 Table 4와 같다.

Table 4. Configuration of Fine-tuning

	Setup
Model	RoBERTa-Large
Learning Rate	5e-5
Epoch	8
Optimizer	AdamW
Batch Size	8
Scheduler	linear

모델 (A) ~ (D)는 참여 PVP를 선정한 후 이에 대해 미세조정 및 앙상블을 진행한다. 이때 앙상블은 미세조정 모델의 평가 데이터에 대한 출력의 가중평균을 산출하여 최종 PET 모델을 생성한다. 반면 모델 (E)는 모든 후보 PVP 각각을 사용하여 미세조정을 한 뒤, 성능 상위 모델 4개를 선정하여 앙상블 한다. (E)의 과정을 진행하기 위한 각 PVP 별 미세조정 모델의 성능은 Table 5와 같다.

Table 5. Accuracy of Each Fine-tuned Model

PVPs	Accuracy	PVPs	Accuracy
PVP 1	0.8834	**PVP 8**	0.7962
PVP 2	0.9292	**PVP 9**	0.5476
PVP 3	0.9406	**PVP 10**	0.8564
PVP 4	0.9636	**PVP 11**	0.9574
PVP 5	0.9424	**PVP 12**	0.9369
PVP 6	0.9627	**PVP 13**	0.9569
PVP 7	0.9223		

Table 5의 결과에 따라 4번, 6번, 11번, 그리고 13번 PVP를 (E)의 참여 PVP로 선정하고 이를 통해 최종 PET 모델을 만든다. 이상의 과정을 통해 선정된 각 모델의 참여 PVP는 Table 6과 같다.

Table 6. Results of Participant PVPs

Model	PVPs
Model (A)	5, 6, 12, 13
Model (B)	3, 5, 7, 13
Model (C)	1, 2, 4, 10
Model (D)	5, 8, 9, 10
Model (E)	4, 6, 11, 13

5. Performance Evaluation

본 절에서는 PET 수행 과정에서 데이터와의 유사도를 기반으로 참여 PVP를 선정하는 제안 방법론의 성능을 비교 모델들과 함께 평가한 결과를 소개한다. 이때, 최종 PET 모델의 성능 평가에는 실험의 이전 과정에서 사용되지 않은 별도의 데이터 10,000개를 사용한다. Table 7은 개별 모델이 미세조정에서 나타낸 정확도의 평균(Average Accuracy of Individual Models)과 모델별로 4개의 개별 모델을 앙상블 하여 구성한 최종 PET 모델의 성능(Accuracy of Ensemble Model)을 나타낸 결과이다.

Table 7. Average Accuracy and Ensemble Accuracy

Model	Average Accuracy of Individual Models				Accuracy of Ensemble Model	
(A) Proposed	PVP 5	0.9463	**0.9522**		**0.9703**	
	PVP 6	0.9639				
	PVP 12	0.9377				
	PVP 13	0.9607				
(B) Random Selection	PVP 3	0.9403	0.9437	**0.8802**	0.9677	**0.9579**
	PVP 5	0.9463				
	PVP 7	0.9273				
	PVP 13	0.9607				

(C) Random Selection	PVP 1	0.8842	0.9094		0.9574	
	PVP 2	0.9341				
	PVP 4	0.9637				
	PVP 10	0.8554				
(D) Random Selection	PVP 5	0.9463	0.7875		0.9488	
	PVP 8	0.8056				
	PVP 9	0.5426				
	PVP 10	0.8554				
(E) Accuracy-based Selection	PVP 4	0.9637	**0.9625**		**0.9603**	
	PVP 6	0.9639				
	PVP 11	0.9616				
	PVP 13	0.9607				

우선 개별 모델들의 정확도 평균을 비교한 결과, 당연히 정확도가 높은 개별 모델들을 조합한 모델 (E)의 정확도가 0.9625로 가장 높게 나타남을 확인하였다. 또한 무작위 추출 모델의 경우 0.7875 ~ 0.9437로 정확도의 편차가 참여 PVP에 따라 크게 나타났으며, 제안 모델은 무작위 추출 모델보다는 높고, 정확도 기반 추출 모델보다는 낮은 성능인 0.9522의 정확도를 나타냈다. 개별 모델들의 정확도 평균은 단순히 참고용으로 산출한 것으로, 실제 모델의 성능은 최종 앙상블 모델의 정확도를 기반으로 평가해야 한다.

최종 앙상블 모델의 성능 평가에서는 제안 모델의 정확도가 0.9703으로 가장 우수하게 나타났다. 이는 정확도가 높게 나타난 4개의 개별 모델을 앙상블 한 모델 (E)에 비해서도 제안 모델이 우수한 성능을 보임을 나타내는 결과이다. 한편 전통적인 PET 모델인 무작위 선정 모델은 최종 앙상블 모델의 성능에서도 모델별 편차를 보였으며, 평균 정확도 역시 모델 (A)와 모델 (E)에 비해 낮게 나타났다 (Fig. 12).

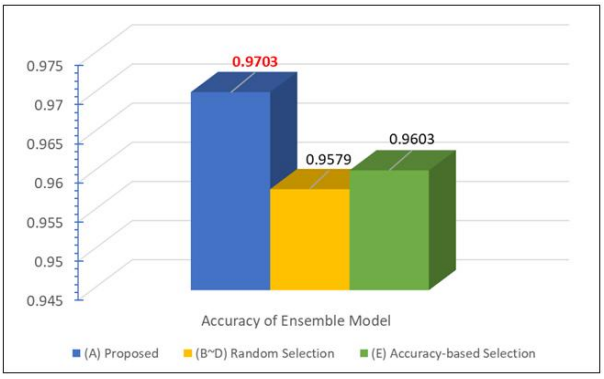

Fig. 12. Performance Comparison

이상 실험의 결과를 활용하여 제안하는 모델인 유사도 기반 PVP 선정 모델과 다른 기법의 모델들을 정확도, 안정성, 그리고 효율성 측면에서 정성적 비교를 수행하였다. 먼저 정확도의 측면에서 앞서 Table 7에서 확인하였듯이 비교 제안 모델 (A), 정확도 기반 추출 모델 (E), 그리고 전통적인 PET 모델 순으로 최종 앙상블 모델의 정확도가 높은 것을 확인할 수 있다. 또한 안정성의 측면에서 선정 조합에 따라 성능의 편차가 발생하는 모델 (B) ~ (D)와 달리, 확실한 기준으로 PVP를 선정하는 제안 모델 (A)와 모델 (E)가 모델을 구현하는 데 있어 편차 없이 안정적이라 할 수 있다. 마지막으로 효율성의 측면에서 모델 (E)와 같이 각기 다른 PVP로 학습한 모든 모델의 성능을 비교하여 가장 높은 PVP를 선정하는 방식과 달리, 제안 모델 (A)와 전통적인 PET 모델은 학습 전 미리 선정하여 불필요한 학습을 방지하기 때문에 효율성이 높다고 할 수 있다. 이를 표를 통해 확인한 결과, 모든 측면에서 제안 방법론이 우수함을 확인하였다(Table 8).

Table 8. Overall Evaluation of the Proposed Model

Criteria	Proposed Model	Traditional Model	Accuracy-based Selection
Accuracy	High	Low	Medium
Robustness	High	Low	High
Efficiency	High	High	Low

V. Conclusions

최근 거대 언어 모델이 다양한 자연어 처리 분야에서 광범위하게 활용됨으로써, 그 우수성을 입증하고 있다. 이를 효과적으로 활용을 위해서는 해당 작업에 맞는 학습이 필요하지만, 이는 방대한 데이터와 대규모 자원을 필요로 한다는 한계점이 존재한다. 이에 따라 최근 적은 자원을 효율적으로 활용한 문맥 내 학습에 대해 활발한 연구가 이루어지고 있다. 그중 본 연구에서는 PVP를 활용하여 프롬프트로 학습하는 기존 PET 기법을 개선하기 위해, 기존 데이터의 문맥과 유사한 패턴을 선정하여 학습하는 방식을 제안하였다. 또한 제안 방법론을 사용하여 분류 실험을 수행한 결과, 제안 방법론이 정확성과 안정성, 그리고 효율성 측면 모두에서 기존 방식에 비해 우수한 성능이 보임을 확인하였다.

본 연구는 문맥 내 학습의 프롬프트 생성에 필요한 PVP를, 무작위가 아닌 기존 데이터의 문맥과의 유사성 기준으로 선정하여 입력 프롬프트를 생성하는 새로운 관점을 제안했다는 점에서 학술적 기여를 인정받을 수 있다. 또한 기존의 후보 PVP를 활용하여 다양한 후보 PVP를 추가적

으로 생성할 수 있다는 점과 우수한 성능을 보일 수 있는 프롬프트 구성을 학습 전에 미리 선정하여 학습을 수행할 수 있다는 점은, 학습의 효율성 측면에서 본 연구의 실무적 기여를 높일 수 있을 것으로 기대한다.

다만, 본 연구는 새롭게 제안한 기준점에 따른 성능 향상에 중점을 두었기 때문에, 기존 PET의 PVP 형식을 기반으로 증강하여 실험을 수행하였다. 이에 따라 각 모델이 제한된 형식의 프롬프트만을 학습하였음을 의미하며, 따라서 선정된 PVP가 각 모델에 최적화된 PVP 임을 보장할 수는 없다는 한계를 갖는다. 향후 다양한 형식의 변형을 이룬 PVP를 활용하여 보다 세밀한 실험이 수행될 필요가 있다. 또한 본 연구는 각 모델에 대한 미세조정 이후 앙상블을 수행하여 최종 모델을 구축하였다. 이는 여러 모델의 예측 편향과 분산을 줄여 최종 모델의 성능과 안정성을 향상시키기 위한 방법이지만, 앙상블의 블랙박스적인 특성으로 인해 결과 해석이 어렵다는 한계가 존재한다. 특히 정확도 기반 추출 모델 (E)의 성능이 향상되지 않고 오히려 저하된 이유에 대해 앙상블 전후 모델의 예측 결과에 편향이나 분산의 차이가 거의 없었을 것이라고 예상할 수 있지만, 이는 추측일 뿐 정확한 해석이 어렵다. 향후 후속 연구에서는 앙상블 과정에 있어 더욱 다양한 PVP 조합을 탐색함으로써, 프롬프트가 앙상블에 미치는 영향과 앙상블 기법이 성능에 미치는 영향 등을 명확히 파악할 수 있을 것으로 기대한다.

REFERENCES

[1] J. Kaplan et al., “Scaling Laws for Neural Language Models.” arXiv.2001.08361, Jan, 2020. DOI: 10.48550/arXiv.2001.08361.

[2] T. B. Brown et al., “Language Models are Few-Shot Learners.” arXiv.2005.14165, May, 2020. DOI: 10.48550/arXiv.2005.14165.

[3] A. Chowdhery et al., “PaLM: Scaling Language Modeling with Pathways.” arXiv.2204.02311, Apr, 2022. DOI: 10.48550/arXiv.2204.02311.

[4] H. Touvron et al., “LLaMA: Open and Efficient Foundation Language Models.” arXiv.2302.13971, Feb, 2023. DOI: 10.48550/arXiv.2302.13971.

[5] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language Models are Unsupervised Multitask Learners," OpenAI blog, Vol. 1, No. 8, pp. 9, 2019.

[6] F. Petroni et al., “Language Models as Knowledge Bases?” arXiv.1909.01066, Sep, 2019. DOI: 10.48550/arXiv.1909.01066.

[7] X. Liu et al., “GPT Understands, Too.” arXiv.2103.10385, Oct, 2021. DOI: 10.48550/arXiv.2103.10385.

[8] T. Shin, Y. Razeghi, R. L. Logan, E. Wallace, and S. Singh, “AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts.” arXiv.2010.15980, Nov, 2020. DOI: 10.48550/arXiv.2010.15980.

[9] T. Gao, A. Fisch, and D. Chen, “Making Pre-trained Language Models Better Few-shot Learners.” arXiv.2012.15723, Dec, 2020. DOI: 10.48550/arXiv.2012.15723.

[10] X. L. Li and P. Liang, “Prefix-Tuning: Optimizing Continuous Prompts for Generation.” arXiv.2101.00190, Jan, 2021. DOI: 10.48550/arXiv.2101.00190.

[11] T. Schick and H. Schütze, “Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference.” arXiv.2001.07676, Jan, 2020. DOI: 10.48550/arXiv.2001.07676.

[12] J. Howard and S. Ruder, “Universal Language Model Fine-tuning for Text Classification.” arXiv.1801.06146, Jan, 2018. DOI: 10.48550/arXiv.1801.06146.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” arXiv.1810.04805, Oct, 2018. DOI: 10.48550/arXiv.1810.04805.

[14] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," 2018.

[15] A. Vaswani et al., “Attention Is All You Need.” arXiv.1706.03762, Jun, 2017. DOI: 10.48550/arXiv.1706.03762.

[16] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, “ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.” arXiv.1909.11942, Sep, 2019. DOI: 10.48550/arXiv.1909.11942.

[17] Y. Liu et al., “RoBERTa: A Robustly Optimized BERT Pretraining Approach.” arXiv.1907.11692, Jul, 2019. DOI: 10.48550/arXiv.1907.11692.

[18] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, “ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators.” arXiv.2003.10555, Mar, 2020. DOI: 10.48550/arXiv.2003.10555.

[19] J. Lee et al., “BioBERT: a pre-trained biomedical language representation model for biomedical text mining,” Bioinformatics, vol. 36, no. 4. Oxford University Press (OUP), pp. 1234–1240, Sep. 10, 2019. DOI: 10.1093/bioinformatics/btz682.

[20] D. Araci, “FinBERT: Financial Sentiment Analysis with Pre-trained Language Models.” arXiv.1908.10063, Aug, 2019. DOI: 10.48550/arXiv.1908.10063.

[21] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, “LEGAL-BERT: The Muppets straight out of Law School.” arXiv.2010.02559, Oct, 2020. DOI: 10.48550/arXiv.2010.02559.

[22] OpenAI et al., “GPT-4 Technical Report.” arXiv.2303.08774,

Mar, 2023. DOI: 10.48550/arXiv.2303.08774.

[23] J. Hoffmann et al., "Training Compute-Optimal Large Language Models." arXiv.2203.15556, Mar, 2022. DOI: 10.48550/arXiv.2203.15556.

[24] W. X. Zhao et al., "A Survey of Large Language Models." arXiv.2303.18223, Mar, 2023. DOI: 10.48550/arXiv.2303.18223.

[25] H. Liu et al., "Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning." arXiv.2205.05638, May, 2022. DOI: 10.48550/arXiv.2205.05638.

[26] E. J. Hu et al., "LoRA: Low-Rank Adaptation of Large Language Models." arXiv.2106.09685, Jun, 2021. DOI: 10.48550/arXiv.2106.09685.

[27] Q. Dong et al., "A Survey on In-context Learning." arXiv.2301.00234, Dec, 2023. DOI: 10.48550/arXiv.2301.00234.

[28] E. Akyürek, D. Schuurmans, J. Andreas, T. Ma, and D. Zhou, "What learning algorithm is in-context learning? Investigations with linear models." arXiv.2211.15661, Nov, 2022. DOI: 10.48550/arXiv.2211.15661.

[29] S. Min et al., "Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?" arXiv.2202.12837, Feb, 2022. DOI: 10.48550/arXiv.2202.12837.

[30] F. Pourpanah et al., "A Review of Generalized Zero-Shot Learning Methods," IEEE Transactions on Pattern Analysis and Machine Intelligence. Institute of Electrical and Electronics Engineers (IEEE), pp. 1–20, 2022. DOI: 10.1109/tpami.2022.3191696

[31] G. Koch, R. Zemel and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition." 2015.

[32] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a Few Examples," ACM Computing Surveys, vol. 53, no. 3. Association for Computing Machinery (ACM), pp. 1–34, Jun. 12, 2020. DOI: 10.1145/3386252.

[33] Y. Gu, X. Han, Z. Liu, and M. Huang, "PPT: Pre-trained Prompt Tuning for Few-shot Learning." arXiv.2109.04332, Sep, 2021. DOI: 10.48550/arXiv.2109.04332.

[34] N. Wies, Y. Levine, and A. Shashua, "The Learnability of In-Context Learning." arXiv.2303.07895, Mar, 2023. DOI: 10.48550/arXiv.2303.07895.

[35] V. Liu and L. B. Chilton, "Design Guidelines for Prompt Engineering Text-to-Image Generative Models." arXiv.2109.06977, Sep, 2021. DOI: 10.48550/arXiv.2109.06977.

[36] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large Language Models are Zero-Shot Reasoners." arXiv.2205.11916, May, 2022. DOI: 10.48550/arXiv.2205.11916.

[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space." arXiv.1301.3781, Jan, 2013. DOI: 10.48550/arXiv.1301.3781.

[38] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics, 2014. DOI: 10.3115/v1/d14-1162.

[39] K. Guu, T. B. Hashimoto, Y. Oren, and P. Liang, "Generating Sentences by Editing Prototypes." arXiv.1709.08878, Sep, 2017. DOI: 10.48550/arXiv.1709.08878.

[40] T. Gao, A. Fisch, and D. Chen, "Making Pre-trained Language Models Better Few-shot Learners." arXiv.2012.15723, Dec, 2020. DOI: 10.48550/arXiv.2012.15723.

[41] J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen, "What Makes Good In-Context Examples for GPT-3?" arXiv.2101.06804, Jan, 2021. DOI: 10.48550/arXiv.2101.06804.

[42] N. Asghar, "Yelp Dataset Challenge: Review Rating Prediction." arXiv.1605.05362, May, 2016. DOI: 10.48550/arXiv.1605.05362.

Authors

Lib Kim received the B.A. degree in English Language and Literature from Chungbuk National University in 2021 and currently enrolled in Graduate School of Business IT, Kookmin University.

He is interested in prompt-based learning, deep learning, and natural language processing

Namgyu Kim received the B.S. in Computer Engineering from Seoul National University in 1998, M.S. and Ph.D. degrees in Management Engineering from KAIST, Korea, in 2000 and 2007, respectively.

Dr. Kim joined the faculty of the School of Management Information Systems at Kookmin University, Seoul, Korea, in 2007. He served as the Dean of the Graduate School of Business IT at Kookmin University and is currently a professor at the Business IT. He is interested in deep learning, text mining, and data modeling.