ISSN 2287-5026 (Print)
ISSN 2288-159X (Online)

Journal of the Institute of Electronics and Information Engineers

2026 4 제63권 4호

Vol.63, No.4 April 2026

www.theieie.org Vol.63, No.4 April 2026
The Institute of Electronics and Information Engineers (IEIE)
Room #907, The Korea Science Technology Center The first building, 22,
Teheran-ro 7 Gil, Gangnam-gu, Seoul, Republic of Korea

전자공학회 논문지
2026 4
제63권 4호
IEIE
사단법인 대한전자공학회

전자공학회논문지
•이 논문집은 한국연구재단 **우수등재학술지**임.

차 례

2026년 4월 제63권 제4호

SD / 반도체

[SoC 설계]

[RF 집적회로기술]

CI / 컴퓨터

[인공지능 및 보안]

AISP / 인공지능 신호처리

[영상 신호처리]

[음향 및 신호처리]

IE / 산업전자

[신호처리 및 시스템]

Journal of The Institute of Electronics and Information Engineers Vol.63, NO.4, April 2026 https://doi.org/10.5573/ieie.2026.63.4.84
ISSN 2287-5026(Print) / ISSN 2288-159X(Online)

논문 2026-63-4-10

성능 향상을 위한 Positional Encoding을 통합한 텐서 분해 기반 차원 축소 기법

(Performance-improving Dimensionality Reduction with Tensor Decomposition and Integrated Positional Encoding)

이 희 열*, 이 승 호**

(Hee-Yeol Lee and Seung-Ho Lee©)

요 약

본 연구에서는 Transformer 기반 언어 모델에서 텐서 분해를 이용한 차원 축소 과정에서 발생할 수 있는 성능 저하 문제를 완화하고, 이를 통해 성능을 향상시키기 위해 Positional Encoding을 통합한 텐서 분해 기반 차원 축소 기법을 제안한다. 기존의 저차원 투영 방식은 입력 표현의 차원을 줄이는 과정에서 위치 정보가 약화되어 성능 저하로 이어질 수 있으며, 단순한 위치 인코딩 추가는 차원 축소된 표현과 충분히 상호작용하지 못해 데이터 특성과 설정에 따라 학습 안정성을 저해할 가능성이 있다. 제안한 방법은 고차원 임베딩을 Tucker 분해 기반 저차원 표현으로 투영한 뒤, 차원 축소로 인해 약화될 수 있는 위치 정보를 저차원 표현 공간에서 직접 보정하도록 설계되었다. 정규화된 사인 코사인 위치 특징과 문맥 기반 게이팅을 결합하여 위치 보정의 강도를 입력 문맥에 따라 적응적으로 조절함으로써, 과도한 위치 정보 주입이나 정보 부족으로 인한 학습 불안정성을 완화하고 차원 축소와 위치 정보 보존을 동시에 달성하도록 설계하였다. WikiText와 IMDb 데이터셋에서 수행한 언어 모델링 실험 결과, 제안한 Tucker 분해와 문맥 적응적 위치 보정 결합 방식은 Transformer 대비 더 낮은 loss와 Perplexity를 기록하며 성능 향상을 확인하였다. 특히 문맥 길이가 긴 IMDb에서 개선 효과가 뚜렷하게 나타나, 장기 문맥 보존에 대한 유효성을 보여준다. 이러한 결과는 본 연구의 기법이 단순 경량화를 넘어 안정적인 성능 유지 및 향상을 가능하게 함을 입증하며, 향후 다양한 언어 처리 과제에서 성능 개선을 위한 차원 축소 모듈로 활용될 수 있을 것으로 기대된다.

Abstract

This study proposes a tensor decomposition based dimensionality reduction method with integrated positional encoding to improve performance degradation caused by dimensionality reduction and improve performance in Transformer language models. Conventional low-dimensional projection approaches can degrade performance by weakening positional information during compression, while simply adding positional encodings may fail to interact effectively with compressed representations and destabilize training depending on dataset characteristics and hyperparameter settings. To address these issues, the proposed method projects high-dimensional token embeddings into a low-dimensional representation using Tucker decomposition and explicitly compensates positional information within the reduced representation by adaptively calibrating positional compensation strength through a combination of normalized sinusoidal positional features and a context-dependent gating mechanism. This design enables effective dimensionality reduction while robustly preserving sequential information and stabilizing training behavior under dimensionality reduction.. Language modeling experiments on the WikiText and IMDb datasets show that the proposed Tucker decomposition with context-adaptive positional compensation achieves lower cross-entropy loss and perplexity than the Transformer baseline. In particular, more pronounced improvements are observed on the IMDb dataset, which contains longer contexts, indicating the effectiveness of the proposed approach in preserving long-range dependencies. These results demonstrate that the proposed method enables stable performance improvement beyond simple dimensionality reduction and can serve as a performance-oriented dimensionality reduction module for various natural language processing tasks.

Keywords : Positional encoding, Tensor decomposition, Dimensionality reduction, Transformer, Embedding representation

*학생회원, **평생회원, 국립한밭대학교 전자공학과(Dept. Electronic Engineering, Hanbat National University)
© Corresponding Author(E-mail : shyolee@hanbat.ac.kr)
※ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 학·석사연계ICT핵심인재양성사업의 연구결과로 수행되었음(IITP-2026-RS-2022-00156212).
Received ; January 2, 2026 Revised ; January 15, 2026 Accepted ; January 22, 2026

Ⅰ. 서 론

최근 트랜스포머 계열 모델의 발전과 함께, 고차원 입력 표현을 효율적으로 처리하기 위한 차원 축소 기법이 자연어 처리와 컴퓨터 비전 분야에서 중요한 연구 주제로 부각되고 있다[1]. 특히 긴 시퀀스 입력을 다루는 언어 모델에서는 연산 비용과 메모리 사용량이 급격히 증가하는 문제가 지속적으로 제기되어 왔다[2]. 이를 완화하기 위해 다양한 파라미터 압축 기법과 저랭크 근사 기법이 제안되었으며[3], 그중 텐서 분해 기반 접근법은 입력 또는 파라미터를 구조적으로 분해함으로써 계산 효율을 개선하는 방법으로 주목받아 왔다[4]. 기존 연구들은 CP 분해, Tucker 분해 등을 활용하여 트랜스포머 구조의 효율화를 시도해 왔다[5]. 한편, 시퀀스 기반 모델에서 위치 정보는 토큰의 의미를 해석하는 데 필수적인 요소이며, 이를 반영하기 위해 Positional Encoding이 널리 사용된다[6]. 일반적인 트랜스포머 구조에서는 위치 정보를 입력 임베딩에 단순히 더하거나, attention 단계에서 별도의 바이어스로 적용하는 방식이 주로 사용되어 왔다[7]. 그러나 입력 표현이 텐서 분해를 통해 저차원으로 압축되는 경우, 위치 정보가 분해 구조 외부에서 독립적으로 처리되면서 내용 정보와의 상호작용이 충분히 반영되지 못할 가능성이 있다. 즉, 텐서 분해 기반 차원 축소는 효율성을 제공할 수 있지만, 위치 정보 결합 방식에 따라 표현력 손실이나 학습 안정성 문제가 발생할 수 있다. 특히 Tucker 분해와 같은 텐서 분해 기법을 입력 표현에 적용할 때, 분해된 저차원 표현 공간에서 위치 정보를 어떤 방식으로 결합할 것인지가 중요한 설계 요소가 된다. 단순 결합 방식은 구현이 간단하지만, 위치 정보가 내용 정보와 분리된 채 주입될 경우 모델이 위치 패턴에 과도하게 의존하거나, 반대로 위치 정보의 효과가 충분히 전달되지 않는 문제가 발생할 수 있다. 그럼에도 불구하고 기존 텐서 분해 기반 연구들은 주로 분해 구조 및 랭크 설정에 초점을 두어 왔으며, 분해된 표현 공간에서 분해된 저차원 표현 공간에서 위치 정보를 어떻게 보정하고 안정적으로 주입할 것인지에 대한 논의는 상대적으로 제한적이었다. 또한 데이터 특성이나 시퀀스 길이에 따라 이러한 부작용의 양상이 달라져 학습 불안정성이 심화될 수 있다. 본 논문은 이러한 문제 의식에 기반하여, 저차원/저랭크 표현을 이용한 모델 압축 연구와 텐서 분해 기반 경량화, 그리고 Positional Encoding 변형 연구 흐름을 연결하는 관점에서 접근한다.

본 논문에서는 Tucker 분해를 기반으로 한 입력 차원 축소 구조에서 Positional Encoding을 저차원 표현 공간에 통합하는 방법을 제안하고, 위치 정보 결합 방식이 모델의 학습 및 성능에 미치는 영향을 체계적으로 분석한다. 특히 문맥 기반 게이팅 전략을 도입하여 위치 보정의 강도를 입력 특성에 따라 적응적으로 조절함으로써, 차원 축소 과정로 인한 위치 정보 손실과 학습 불안정성을 완화한다. 구체적으로, 텐서 분해를 통해 얻어진 저차원 표현과 위치 정보를 동일한 표현 공간에서 결합하고, 위치 정보와 내용 정보가 저차원 단계에서 상호작용할 수 있도록 설계함으로써, 위치 정보와 내용 정보가 저차원 표현 단계에서 상호작용할 수 있는 구조를 구성한다. 또한 공정한 언어 모델링 평가를 위해 causal masking을 적용하고 비교 실험한다.

Ⅱ. 본 론

1. 제안하는 연구의 개요도

그림 1은 본 연구에서 제안하는 텐서 분해 기반 차원 축소 방법의 전체 구조를 나타낸다. 입력 시퀀스는 먼저 토큰 임베딩을 통해 고차원 표현으로 변환되며, 이후 각 위치에 대응하는 Positional Encoding이 결합되어 위치 정보를 포함하는 입력 표현이 구성된다. 다음으로 입력 표현은 차원 D를 h 곱하기 w 형태로 재구성하여 토큰별 2차 텐서 표현으로 변환된다. 변환된 텐서는 Tucker 분해 기반 모듈을 통해 저차원 텐서로 투영되며, 벡터화를 통해 저차원 토큰 임베딩 벡터가 생성된다. 또한 본 연구에서는 Tucker 분해 과정에서 학습되는 기저벡터를 활용하여 Positional Encoding을 생성하고 코어 텐서와 통합함으로써, 위치 정보가 차원

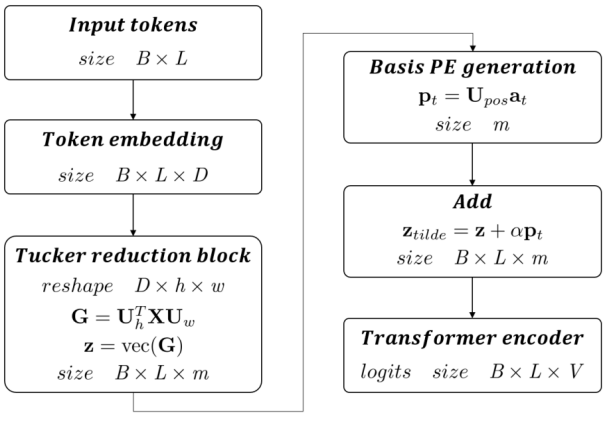

그림 1. 제안한 연구의 개요도
Fig. 1. Overview of the proposed study.

축소 구조 내부에서 일관되게 반영되도록 설계하였다. 최종적으로 생성된 저차원 시퀀스 표현은 트랜스포머 기반 언어 모델의 입력으로 사용되어 학습 및 평가를 수행한다.

2. Tucker 분해 기반 텐서 차원 축소 과정

본 연구에서는 고차원 입력 표현을 효율적으로 압축하기 위해 Tucker 분해를 기반으로 한 차원 축소 과정을 설계하였다. 기존의 벡터 기반 차원 축소 방식은 입력을 1차원 벡터로 취급하는 경향이 있어, 텐서의 축 방향 구조와 축 사이 상호작용을 반영하기 어렵다. Tucker 분해는 그림 2와 같이 입력을 텐서 구조로 해석한 뒤 각 축에 대한 저랭크 근사를 수행하므로, 구조적 관계를 유지하면서 표현 차원을 줄일 수 있다. 먼저 입력 임베딩 E는 D차원의 벡터로 변환된다. 이후 D를 $h \times w$로 구성하여 각 토큰 벡터를 $h \times w$ 크기의 2차원 텐서로 reshape 한다. 이 텐서화는 차원 재배열이지만, Tucker 분해에서 요구하는 축 구조를 명시적으로 제공하며, 이후 분해 과정에서 축 방향 변동과 축 사이 결합 구조를 더 직접적으로 표현할 수 있도록 한다. 결과적으로 배치 크기 B와 시퀀스 길이 L에 대해 입력은 $B \times L \times h \times w$ 형태의 텐서로 표현된다. 본 연구에서는 토큰별 2차원 텐서에 대해 두 축 방향 요인 행렬을 학습 가능한 파라미터로 두고[8], 이를 이용해 랭크가 축소된 코어 텐서를 계산한다. 구체적으로 h 축에 대한 요인 행렬 U_h와 w 축에 대한 요인 행렬 U_w를 사용하여 각 토큰 텐서 X에 대해 $U_h^T \times X \times U_w$ 형태로 저차원 텐서를 얻는다. 여기서 분해 랭크 r_0와 r_1 은 각각 h 축과 w축에서 유지할 저차원 차원을 의미하며, 최종적으로 각 토큰은 $r_0 \times r_1$ 크기의 텐서로 압축된다. 마지막으로 압축된 텐서는 1차원 벡터로 변환되어 토큰 표현 z로 사용된다. 분해 랭크 r_0와 r_1 은 차원 축소 정도와 표현력에 직접적인 영향을 미치는 핵심 파라미터이다. $r_0 \times r_1$ 이 작을수록 토큰당 표현 차원이 감소하여 계산량이 줄어들지만 정보 손실이 증가할 수 있다. 반대로 랭크를 증가시키면 정보 보존은 유리해지지만 압축 이점은 감소한다. 그림 3은 Tucker 분해 기반 텐서 차원 축소를 나타낸다. 식 (1)은 일반적인 Tucker 분해 식이고, 식 (2)부터 식 (6)은 그림 3에 제시된 처리 흐름을 수식으로 정리한 것이다. 식 (2)는 입력 임베딩 $E_{b,t}$를 2차원 텐서 $X_{b,t}$로 재구성하는 단계를 나타내며, 식 (3)과 식 (4)는 각 축에 대한 요인 행렬을 정의한다. 이후 식 (5)는 Tucker 분해를 통해 코어 텐서 $G_{b,t}$를 계산하는 과정이며, 식 (6)은 이를 벡터화하여 최종 저차원 토큰 표현 $z_{b,t}$를 생성한다.

$$\mathcal{X} \approx \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_3 \mathbf{U}^{(3)} \tag{1}$$

$$\mathbf{E} \in \mathbb{R}^{B \times L \times D} W \tag{2}$$

$$D = h \cdot w, \qquad \mathbf{X} = \mathrm{reshape}(\mathbf{E}), \qquad \mathbf{X} \in \mathbb{R}^{B \times L \times h \times w} \tag{3}$$

$$\mathbf{U}_h \in \mathbb{R}^{h \times r_0}, \qquad \mathbf{U}_w \in \mathbb{R}^{w \times r_1} \tag{4}$$

$$\mathbf{G}_{b,t} = \mathbf{U}_h^{\top} \mathbf{X}_{b,t} \mathbf{U}_w, \qquad \mathbf{G}_{b,t} \in \mathbb{R}^{r_0 \times r_1} \tag{5}$$

$$\mathbf{z}_{b,t} = \mathrm{vec}(\mathbf{G}_{b,t}), \qquad \mathbf{z}_{b,t} \in \mathbb{R}^{r_0 r_1}, \qquad \mathbf{Z} \in \mathbb{R}^{B \times L \times r_0 r_1} \tag{6}$$

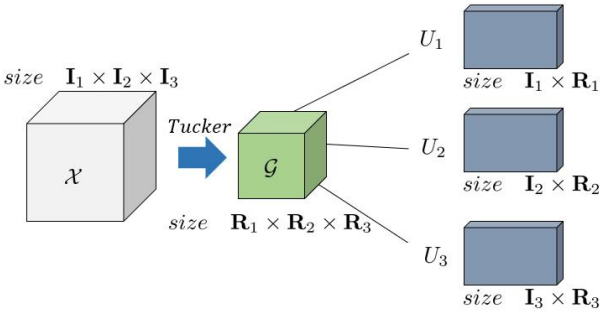

그림 2. 일반적인 Tucker 분해
Fig. 2. General Tucker decomposition.

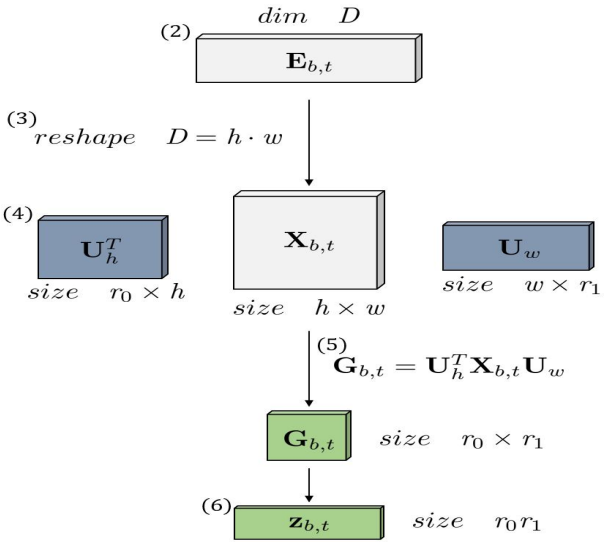

그림 3. Tucker 분해 기반 텐서 차원 축소
Fig. 3. Tucker decomposition-based tensor dimensionality reduction.

3. Tucker 기저벡터 기반 Positional Encoding 통합 과정

본 연구에서는 Tucker 분해를 통해 학습되는 기저벡터를 활용하여 Positional Encoding을 구성하고, 이를 저차원 표현 공간에 통합하는 방법을 설명한다. 일반적인 Positional Encoding은 시퀀스 위치마다 독립적인

벡터를 정의하거나, 사인 코사인 형태의 고정 함수를 사용하여 위치 벡터를 생성한다. 이러한 방식은 구현이 단순하지만, 차원 축소가 적용된 경우에는 위치 정보가 축소된 표현 공간과 분리된 형태로 주입될 수 있으며, 결과적으로 내용 정보와 위치 정보가 동일한 기저 위에서 상호작용하기 어렵다는 문제가 발생할 수 있다. 본 연구는 이러한 분리 구조를 완화하기 위해, Tucker 분해 과정에서 학습되는 기저벡터를 위치 정보 표현의 기반으로 활용하는 방식을 채택한다. 먼저 본 연구에서의 Tucker 분해는 토큰별 입력 텐서를 요인 행렬과 저차원 텐서로 투영하여 압축 표현을 생성한다. 이때 요인 행렬은 입력 텐서의 각 방향에서 데이터의 주요 변동을 설명하는 기저벡터들의 집합으로 해석할 수 있다. 본 절의 핵심 아이디어는 위치 정보를 별도의 임베딩 테이블로 정의하는 대신, Tucker 요인 행렬이 제공하는 기저벡터 공간에서 위치 벡터를 생성함으로써 위치 정보가 차원 축소 구조 내부의 기저와 직접적으로 정렬되도록 만드는 것이다. 즉, 위치 벡터는 독립적인 파라미터가 아니라, Tucker 기저를 선형 결합한 결과로 정의된다. 그림 4는 Tucker 기저벡터 기반 Positional Encoding의 생성 및 결합 과정을 나타낸다. 우선 Tucker 분해를 통해 얻어진 기저벡터 집합을 위치 기저 행렬로 두고, 각 시퀀스 위치에 대해 위치 계수 벡터를 정의한다. 위치 계수 벡터는 위치 인덱스를 입력으로 하는 작은 생성 모듈 또는 학습 가능한 테이블로 구성될 수 있으며, 이 계수 벡터와 위치 기저 행렬의 곱을 통해 위치 벡터가 생성된다. 생성된 위치 벡터는 Tucker 분해를 통해 얻은 저차원 토큰 표현과 동일한 차원을 가지며, 더하기 연산을 통해 최종 입력 표현으로 통합된다. 이때 위치 정보의 영향력을 조절하기 위해 스칼라 계수 알파를 도입하여 학습 안정성을 확보한다. 식 (7)부터 식 (10)은 그림 4에서 기저벡터 기반 Positional Encoding의 정의를 나타낸다. 식 (7)은 Tucker 분해를 통해 얻어진 저차원 토큰 표현 $z_{b,t}$를 정의한다. 식 (8)과 식 (9)는 위치 기저 행렬 U_{pos}와 위치 계수 벡터 a_t를 이용해 위치 벡터 p_t를 생성하는 과정을 나타낸다. 마지막으로 식 (10)은 위치 벡터 p_t를 스칼라 계수 α로 조절하여 토큰 표현에 결합하는 단계로, 이는 그림 4의 최종 출력에 해당한다. 위치 기저 행렬 U_{pos}는 Tucker 분해 과정에서 학습된 요인 행렬 또는 그로부터 도출된 기저를 의미하며, 각 위치 t에 대한 위치 계수 벡터 a_t는 위치별 조합 가중치를 의미한다. 이 두 요소의 선형 결합을 통해 위치 벡터 p_t를 생성하고, 이를 저차원 토큰 표현 z에 더하여 최종 입력 표현을 구성한다. 제안하는 방법은 위치 벡터가 Tucker 기저 공간에서 생성되도록 하여, 위치 정보와 내용 정보가 동일한 저차원 표현 공간에서 결합되도록 유도한다. 또한 위치 계수 벡터를 통해 위치별 변화를 제한된 차원 d에서 표현하므로, 위치 정보 표현의 파라미터 규모를 제어하면서도 필요한 위치 변화를 유연하게 모델링할 수 있다.

$$\mathbf{z}_{b,t} \in \mathbb{R}^m \tag{7}$$

$$\mathbf{U}_{\text{pos}} \in \mathbb{R}^{m \times d} \tag{8}$$

$$\mathbf{a}_t \in \mathbb{R}^d, \qquad \mathbf{p}_t = \mathbf{U}_{\text{pos}}\mathbf{a}_t \in \mathbb{R}^m \tag{9}$$

$$\tilde{\mathbf{z}}_{b,t} = \mathbf{z}_{b,t} + \alpha \mathbf{p}_t \tag{10}$$

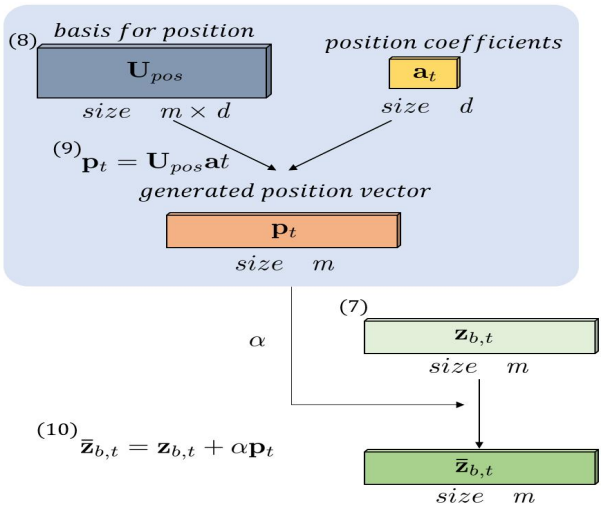

그림 4. 저차원 표현 공간에서의 Positional Encoding 결합
Fig. 4. Combining Positional Encoding in Low-Dimensional Representation Spaces.

Ⅲ. 실 험

1. 실험 환경

실험에 사용된 운영체제 및 하드웨어는 Ubuntu 20.04.2 LTS 운영체제를 기반으로 Intel(R) Xeon(R) Silver 4210 CPU @ 2.20GHz, RAM 128GB, NVIDIA RTX A5000 (VRAM 24GB) GPU로 구성되어 있다. 개발도구는 Visual Studio Code와 PyTorch 1.8.0, CUDA 11.1, cuDNN 8.0.5 라이브러리를 사용하였다.

2. 데이터셋

본 논문에서는 제안하는 Positional Encoding 기반

텐서 분해 차원 축소 기법의 성능을 검증하기 위해 WikiText와 IMDb 두 가지 데이터셋을 사용하였다. 첫 번째 데이터셋은 IMDb 영화 리뷰 데이터셋으로, 긍정과 부정 두 가지 감정 클래스를 갖는 대규모 영어 리뷰 코퍼스이다[9]. IMDb 데이터셋은 총 50,000개의 리뷰 텍스트로 구성되며, 본 연구에서는 라벨 정보를 사용하지 않고 언어 모델링 학습용 텍스트로 활용하였다. 리뷰 문장은 길이 분포가 넓고 장문 문서가 다수 포함되어 있어, 차원 축소 과정에서 장기 문맥과 위치 정보 보존 능력을 평가하기에 적합하다. 두 번째 데이터셋은 WikiText 데이터셋으로, 위키피디아 기반의 문서형 텍스트로 구성된 언어 모델링 벤치마크이다[10]. WikiText는 문장 경계와 문서 구조가 비교적 자연스럽게 유지되어 있으며, 다양한 길이의 시퀀스를 포함하고 있어 순차적 예측 성능과 위치 정보 활용 효과를 분석하는 데 유용하다. 두 데이터셋 모두 고차원 임베딩 입력을 갖는 시퀀스 처리 문제로서, 제안한 텐서 분해 기반 차원 축소 기법의 일반성과 효과를 검증하기 위한 실험 환경으로 활용하였다.

3. 실험 결과 및 고찰

본 연구에서는 제안하는 Positional Encoding 기반 텐서 분해 차원 축소 기법의 유효성을 검증하기 위해 WikiText와 IMDb 데이터셋에서 언어 모델 학습 실험을 수행하였다. 모든 실험은 동일한 학습 예산을 유지한 상태에서 입력 차원 축소 방식과 위치 정보 보정 방식만을 변경하여 비교하였다. 성능 평가는 Cross Entropy loss과 Perplexity를 사용하였다. 모든 실험에서는 동일한 토크나이저와 모델 설정을 사용하여 비교의 공정성을 유지하였다. 토크나이저는 공백 기반 토큰화를 적용하였으며, 어휘 집합은 학습 데이터에서만 구축하였다. 학습은 고정된 랜덤 시드를 사용하여 수행하였고, 각 방법은 동일한 학습 스텝 수와 배치 크기에서 학습되었다. 또한 입력 차원 축소 여부와 관계없이 Transformer 블록의 구조와 파라미터 설정은 동일하게 유지하여, 성능 차이가 차원 축소 및 위치 정보 보정 방식에 의해 발생하도록 설계하였다. 평가는 학습에 사용되지 않은 시퀀스에 대해 autoregressive 언어 모델링 방식으로 수행하였으며, loss와 Perplexity는 동일한 평가 길이 설정에서 계산하였다.

가. 정량적 평가

표 1과 표 2는 네 가지 비교 방법의 평가 결과를 나타낸다. 제안 기법의 효과를 명확히 분석하기 위해 네 가지 비교 방법을 설정하였다. 첫째, 트랜스포머는 입력 임베딩 시퀀스를 차원 축소 없이 그대로 처리하는 기본 구조로 사용하였다. 둘째, 트랜스포머 기반 위치 인코딩 적용 방식은 학습 가능한 위치 인코딩을 추가하여 위치 정보 주입 효과만을 분리해 평가한다. 셋째, Tucker 분해 기반 차원 축소 방식은 입력 임베딩을 Tucker 분해 기반 저차원 표현으로 투영하여 차원 축소 자체의 영향을 확인한다. 넷째, Tucker 분해와 위치 보정 결합 방식은 Tucker 기반 투영에 위치 특징을 결합하되 문맥에 따라 주입 강도를 조절하는 모듈을 포함하여, 차원 축소와 위치 정보 보존을 동시에 강화하는 목적을 갖는다. 이를 통해 단순 위치 정보 주입, 텐서 분해 기반 투영, 그리고 문맥 적응적 위치 보정의 기여도를 각각 분리하여 비교하고 분석하였다. WikiText에서는 Tucker 분해와 문맥 적응적 위치 보정을 결합한 방법이 트랜스포머 대비 더 낮은 loss과 Perplexity를 기록하여 성능 향상을 확인하였다. 반면 학습 가능한 위치 인코딩을 단순히 추가한 방법은 성능이 하락하는 경향을 보였으

표 1. WikiText 데이터셋 성능 비교
Table 1. Performance comparison on WikiText dataset.

Method	Test length	Loss ↓	Perplexity ↓
Transformer[1]	64	5.1570	173.6445
	128	5.1684	175.6300
Transformer + PE[6]	64	5.5045	245.7871
	128	5.5183	249.2102
Tucker[5]	64	5.1847	178.5181
	128	5.2003	181.3329
The proposed method	64	5.1322	169.3834
	128	5.1525	172.8582

표 2. IMDb 데이터셋 성능 비교
Table 2. Performance comparison on IMDb dataset.

Method	Test length	Loss ↓	Perplexity ↓
Transformer[1]	64	5.6482	283.7851
	128	5.6494	284.1189
Transformer + PE[6]	64	6.6671	786.0929
	128	6.6660	785.2680
Tucker[5]	64	6.6714	789.4760
	128	6.6702	788.5278
The proposed method	64	5.5773	264.3580
	128	5.5890	267.4580

며, 이는 위치 정보 주입이 과도할 경우 학습 안정성을 저해할 수 있음을 시사한다. Tucker 분해만 적용한 경우에는 성능이 유사하거나 소폭 악화되었으나, 문맥 기반 주입 강도 조절을 포함하면 성능이 회복되며 가장 우수한 결과를 보였다. IMDb에서도 Tucker 분해와 문맥 적응적 위치 보정을 결합한 방법이 트랜스포머 대비 더 낮은 loss과 Perplexity를 기록하였다. 반면 학습 가능한 위치 인코딩만 적용한 방법과 Tucker 분해만 적용한 방법은 Perplexity가 크게 증가하여 성능 저하가 두드러졌다. 이는 장문 문맥이 중요한 IMDb에서 위치 정보 보정 없이 차원 축소를 적용할 경우 표현 손실이 커질 수 있음을 보여준다.

나. 결과에 대한 고찰

표 1과 표 2의 결과를 통해 제안한 Positional Encoding 통합 Tucker 분해 기반 차원 축소 기법의 효과를 확인할 수 있다. WikiText 데이터셋에서는 Tucker 분해만 적용한 경우 Transformer 대비 성능이 유사하거나 소폭 저하되었으며, 학습 가능한 Positional Encoding을 단순히 추가한 방법은 Perplexity가 크게 증가하였다. 이는 위치 정보를 일괄적으로 주입할 경우 오히려 학습 안정성이 저해될 수 있음을 시사한다. 반면, Tucker 분해와 문맥 적응적 위치 보정을 결합한 방법은 두 평가 길이 설정 모두에서 Transformer 대비 더 낮은 손실과 Perplexity를 기록하여, 차원 축소 환경에서도 순서 정보가 효과적으로 보존됨을 보여준다. IMDb 데이터셋에서는 이러한 경향이 더욱 뚜렷하게 나타났다. 장문 리뷰가 다수 포함된 특성으로 인해, 위치 정보 보정 없이 차원 축소를 적용한 방법들은 성능 저하가 크게 관측되었다. 반면 제안한 방법은 두 평가 길이 모두에서 Transformer 대비 더 낮은 손실과 Perplexity를 기록하며 성능 개선을 보였다. 이는 문맥에 따라 위치 정보 주입 강도를 조절하는 구조가 장기 문맥 의존성을 유지하는 데 효과적으로 작용했기 때문으로 해석된다. 종합하면, 제안한 기법은 단순한 저랭크 근사나 위치 인코딩 추가 방식과 달리, 차원 축소로 인해 약화될 수 있는 순서 정보를 문맥 기반으로 보완함으로써 성능을 향상시킨다. 이러한 결과는 제안 구조가 다양한 시퀀스 길이와 데이터 특성에서도 안정적인 성능을 제공할 수 있는 실용적인 차원 축소 방법임을 시사한다.

Ⅳ. 결 론

본 연구에서는 트랜스포머 기반 언어 모델에서 차원 축소로 인한 성능 저하를 완화하기 위해, Positional Encoding을 통합한 Tucker 분해 기반 차원 축소 기법을 제안하였다. 제안한 방법은 고차원 임베딩을 저차원 텐서 표현으로 투영한 뒤, 문맥 적응적 위치 보정 모듈을 통해 차원 축소 과정에서 약화될 수 있는 순서 정보를 선택적으로 보완하도록 설계되었다. WikiText와 IMDb 데이터셋 실험 결과, 제안한 방법은 트랜스포머 대비 더 낮은 loss와 Perplexity를 기록하며 성능 개선을 확인하였다. 반면 단순 위치 인코딩 추가 방식이나 Tucker 분해만 적용한 방법은 일부 설정에서 성능 저하가 관측되어, 위치 정보 보정의 중요성을 확인할 수 있었다. 특히 장문 문맥이 포함된 IMDb 데이터셋에서 제안한 구조는 차원 축소 환경에서도 장기 문맥 보존에 효과적으로 작용하였다. 이러한 결과는 제안 기법이 성능 향상을 고려한 실용적인 차원 축소 방법임을 시사한다.

향후 연구에서는 다양한 랭크 설정과 위치 보정 강도에 대한 추가 분석과 함께, 보다 대규모 언어 모델과 다양한 언어 처리 과제로 확장하여 제안 기법의 일반성과 활용 가능성을 검증할 필요가 있다.

REFERENCES

[1] Vaswani, Ashish, et al. "Attention Is All You Need." NeurIPS, 2017.
[2] Wang, Sinong, et al. "Linformer: Self-Attention with Linear Complexity." arXiv:2006.04768, 2020.
[3] Novikov, Alexander, et al. "Tensorizing Neural Networks." NeurIPS, 2015.
[4] Kolda, Tamara G., and Brett W. Bader. "Tensor Decompositions and Applications." SIAM Review, 2009.
[5] Tucker, Ledyard R. "Some Mathematical Notes on Three-Mode Factor Analysis." Psychometrika, 1966.
[6] Su, Jianlin, et al. "RoFormer: Enhanced Transformer with Rotary Position Embedding." arXiv:2104.09864, 2021.
[7] Press, Ofir, et al. "Train Short, Test Long." arXiv:2108.12409, 2021.
[8] De Lathauwer, Lieven, et al. "A Multilinear Singular Value Decomposition." SIAM J. Matrix Anal. Appl., 2000.
[9] Maas, Andrew L., et al. "Learning Word

Vectors for Sentiment Analysis." ACL, 2011.
[10] Merity, Stephen, et al. "Pointer Sentinel Mixture Models." arXiv:1609.07843, 2016.

저 자 소 개

이 희 열(학생회원)
2016년 국립한밭대학교 전자공학과 학사 졸업.
2018년 국립한밭대학교 전자공학과 석사 졸업.
2018년~현재 국립한밭대학교 전자공학과 박사과정.
<주관심분야 : 영상신호처리, 딥러닝>

이 승 호(평생회원) - 교신저자
1986년 한양대학교 전자공학과 학사 졸업.
1989년 한양대학교 전자공학과 석사 졸업.
1994년 한양대학교 전자공학과 박사 졸업.
1994년~현재 국립한밭대학교 전자공학과 교수
<주관심분야 : 영상신호처리, 딥러닝, AR>