
An Efficient Framework for Learning Sentence Representations

(ICLR 2018)

박은환 (20163108)

judepark@kookmin.ac.kr

Kookmin University @ Machine Intelligence Lab

Contents

1. Background
2. Proposed Framework
3. Implementation details
4. Experimental Results
5. Conclusion

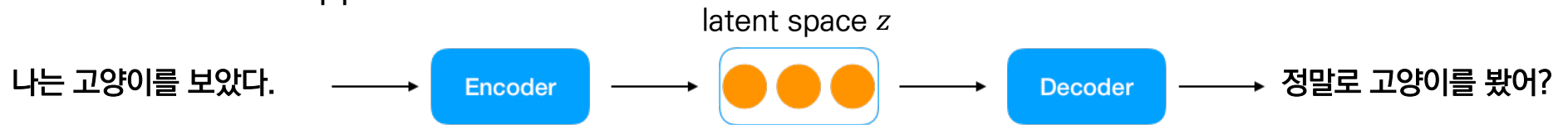
Background

1. 문장 임베딩이란?

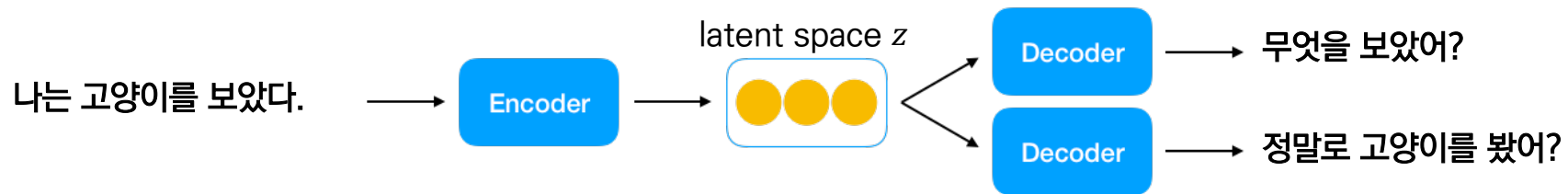
- 문장 전체를 벡터 공간 상의 표현하는 것임.
- 이를 통하여, 분류나 클러스터링과 같은 기법 뿐만 아니라, 자동 질의 응답 등의 과업에 응용 가능.

Background: Conventional Approach

1. Conventional Approach



2. Skip Thought (kiros et al.)



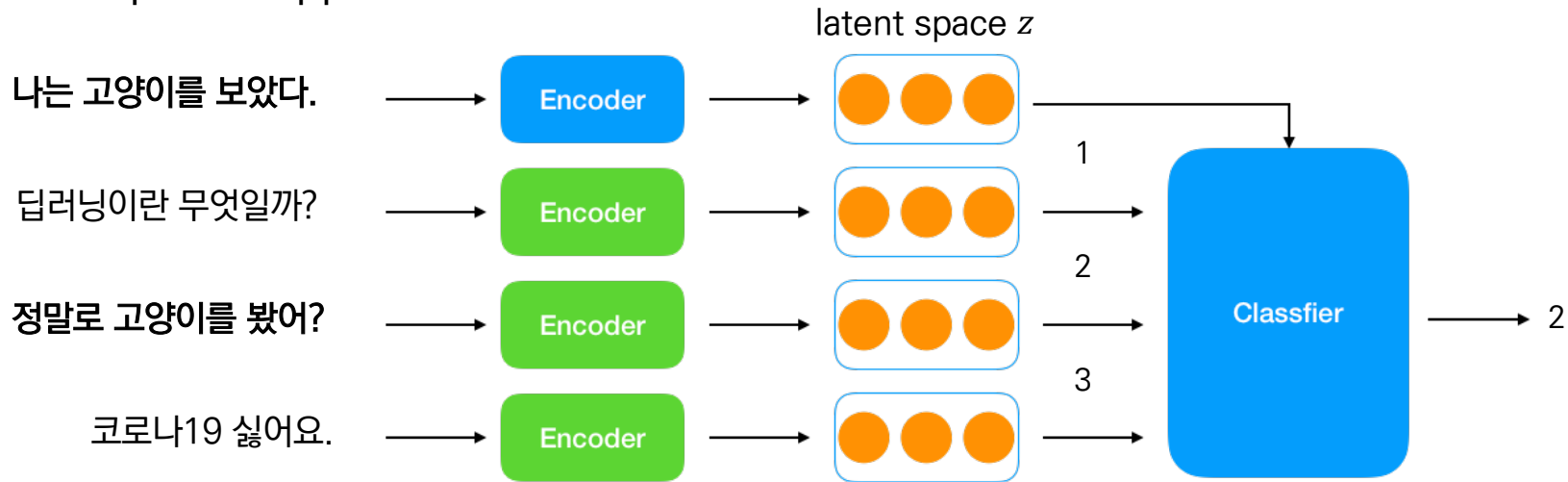
문장 임베딩에서 좋은 결과를 보여줬을 뿐만 아니라 구현하기에도 쉬움.

Limitation: 문장의 의미를 고려하지만 이와 관련없는 부분도 학습하도록 함.

또한, 디코딩 과정에서 많은 연산을 필요로 함. (단어 사전의 크기로 인한 연산 비용 발생)

Proposed Framework

1. Proposed Approach



현재 문장에 맞는 맥락 문장은 무엇인지 분류하는 것으로 문제를 정의함.

Proposed Framework

1. Proposed Approach

$$\mathbb{E}_{\{x, x^+, x^-\}} \left[-\log \left(\frac{e^{c(x, x^+)}}{e^{c(x, x^+)} + e^{c(x, x^-)}} \right) \right] \begin{cases} x = \text{given sentence} \\ x^+ = \text{positive sample} \\ x^- = \text{negative sample} \end{cases}$$

x 와 x^+ 는 positive 이고 x 와 x^- 가 negative 라고 정의한다면
학습을 통하여 x 와 x^+, x^- 간의 유사성을 알아보는 것임.

2. Alternative Approach: Sentence-BERT

$$\mathbb{E}_{\{x_a, x_p, x_n\}} = \max \left(\|x_a - x_p\| - \|x_a - x_n\| + \epsilon, 0 \right), \text{ where } \|\cdot\| \text{ is distance metric.}$$

Distance metric 을 통하여 임베딩 공간에서 x_a 와 x_p 는 가깝게, x_a 와 x_n 는 멀도록 조정.
이를 통하여, 임베딩 공간에서 x 와 $sample$ 간의 유사성을 알아볼 수 있음.

Proposed Framework

1. 왜 이런 접근이 가능할까?

문장과 문장 사이의 빈 칸에 넣는다면 어떠한 문장이 들어갈까?

문장 : 무엇을 보았어?

빈칸 : _____ = 🤔

문장 : 정말 고양이를 보았어?

In language domain, Distributional Hypothesis! 😎

비슷한 위치에 등장하는 단어들은 그 의미도 유사할 것이라는 가정. (Word2Vec)

그렇다면, 비슷한 맥락에 있는 문장들은 그 의미도 유사할 것이라는 가정이 가능.

Implementation Details

Let D be a minibatch of sentences.

$$D = \begin{bmatrix} s_0 \\ \vdots \\ s_n \end{bmatrix} \in \mathbb{R}^{n \times \max_seq_len} \text{ then } \begin{cases} \text{encoder } f = \begin{bmatrix} f(s_0) \\ \vdots \\ f(s_n) \end{bmatrix} \in \mathbb{R}^{n \times h} \\ \text{encoder } g = \begin{bmatrix} g(s_0) \\ \vdots \\ g(s_n) \end{bmatrix} \in \mathbb{R}^{n \times h} \end{cases}$$

Implementation Details

$$Score = fg^T = \begin{bmatrix} f(s_0)g(s_0)^T & \cdots & f(s_0)g(s_n)^T \\ \vdots & \ddots & \vdots \\ f(s_n)g(s_0)^T & \cdots & f(s_n)g(s_n)^T \end{bmatrix} \in \mathbb{R}^{n \times n} \quad Target = \begin{bmatrix} 0 & \cdots & 0 \\ 0 & \ddots & 0 \\ \vdots & \ddots & 1 \\ 0 & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

$$Masked\ Score = \begin{bmatrix} 0 & \cdots & f(s_0)^T g(s_n) \\ \vdots & \ddots & \vdots \\ f(s_n)^T g(s_0) & \cdots & 0 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

$$prob = softmax(masked_score)$$

$$\mathbb{L} = D_{KL}(target_i || prob_i)$$

Experimental Results

1. Model
 - BERT-based Encoder
 - \mathbb{R}^{768}
2. Setting
 - Batch Size 50 (RTX 2070)
 - Adam Optimizer (lr=1e-5)
 - Gradient Clipping (max_norm=2.0)
3. Dataset
 - BookCorpus (40,000,000 sentences.)

Experimental Results

Model	Dim	Batch Size	MRQA
Combine-QT	4800	400	88.0
BERT-based	768	50	89.6
SBERT-NLI-base	N/A	N/A	89.86

SentEval Task 중 하나인 MRQA(Machine Reading for Question Answering) 로 평가를 진행함.

Quick Thought

- 4800차원의 문장 임베딩 구축, 88.0% 의 정확도를 달성

BERT-based

- 본 논문의 Encoder 를 BERT 로 교체하여 768차원의 문장 임베딩 구축
1.6% 만큼 높은 정확도 달성

SBERT-NLI-base

- 최근 SOTA 인 문장 임베딩. BERT-based 에 비해 정확도 0.26% 높음.

Conclusion

1. 문장 임베딩에선 맥락 간 관계성을 보는 것이 중요함.
 - NSP (Next Sentence Prediction, BERT)
 - Text Similarity
2. 단순한 접근으로 이전의 성공적이었던 Skip-Thought 보다 더 좋은 문장 임베딩을 구축할 수 있음.

References

- [1]. An Efficient Framework for Learning Sentence Representations
- [2]. Skip-Thought Vectors
- [3]. A Theoretical Analysis of Contrastive Unsupervised Representation Learning
- [4]. SentEval: Evaluation Toolkit for Sentence Embedding (<https://github.com/facebookresearch/SentEval>)
- [5]. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks
- [6]. Efficient Estimation of Word Representations in Vector Space

감사합니다.