KIPS Trans. Softw. and Data Eng.
Vol.11, No.3 pp.125~132
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2022.11.3.125

Deletion-Based Sentence Compression Using Sentence Scoring Reflecting Linguistic Information

Jun-Beom Lee† · So-Eon Kim†† · Seong-Bae Park†††

ABSTRACT

Sentence compression is a natural language processing task that generates concise sentences that preserves the important meaning of the original sentence. For grammatically appropriate sentence compression, early studies utilized human-defined linguistic rules. Furthermore, while the sequence-to-sequence models perform well on various natural language processing tasks, such as machine translation, there have been studies that utilize it for sentence compression. However, for the linguistic rule-based studies, all rules have to be defined by human, and for the sequence-to-sequence model based studies require a large amount of parallel data for model training. In order to address these challenges, Deleter, a sentence compression model that leverages a pre-trained language model BERT, is proposed. Because the Deleter utilizes perplexity based score computed over BERT to compress sentences, any linguistic rules and parallel dataset is not required for sentence compression. However, because Deleter compresses sentences only considering perplexity, it does not compress sentences by reflecting the linguistic information of the words in the sentences. Furthermore, since the dataset used for pre-learning BERT are far from compressed sentences, there is a problem that this can lad to incorrect sentence compression. In order to address these problems, this paper proposes a method to quantify the importance of linguistic information and reflect it in perplexity-based sentence scoring. Furthermore, by fine-tuning BERT with a corpus of news articles that often contain proper nouns and often omit the unnecessary modifiers, we allow BERT to measure the perplexity appropriate for sentence compression. The evaluations on the English and Korean dataset confirm that the sentence compression performance of sentence-scoring based models can be improved by utilizing the proposed method.

Keywords : Sentence Compression, Linguistic Information, Language Model, Perplexity

언어 정보가 반영된 문장 점수를 활용하는 삭제 기반 문장 압축

이 준 범† · 김 소 언†† · 박 성 배†††

요 약

문장 압축은 원본 문장의 중요한 의미는 유지하면서 길이가 축소된 압축 문장을 생성하는 자연어처리 태스크이다. 문법적으로 적절한 문장 압축을 위해, 초기 연구들은 사람이 정의한 언어 규칙을 활용하였다. 또한 시퀀스-투-시퀀스 모델이 기계 번역과 같은 다양한 자연어처리 태스크에서 좋은 성능을 보이면서, 이를 문장 압축에 활용하고자 하는 연구들도 존재했다. 하지만 언어 규칙을 활용하는 연구의 경우 모든 언어 규칙을 정의하는 데에 큰 비용이 들고, 시퀀스-투-시퀀스 모델 기반 연구의 경우 학습을 위해 대량의 데이터셋이 필요하다는 문제점이 존재한다. 이를 해결할 수 있는 방법으로 사전 학습된 언어 모델인 BERT를 활용하는 문장 압축 모델인 Deleter가 제안되었다. Deleter는 BERT를 통해 계산된 perplexity를 활용하여 문장을 압축하기 때문에 문장 압축 규칙과 모델 학습을 위한 데이터셋이 필요하지 않다는 장점이 있다. 하지만 Deleter는 perplexity만을 고려하여 문장을 압축하기 때문에, 문장에 속한 단어들의 언어 정보를 반영하여 문장을 압축하지 못한다. 또한, perplexity 측정을 위한 BERT의 사전 학습에 사용된 데이터가 압축 문장과 거리가 있어, 이를 통해 측정된 perplexity가 잘못된 문장 압축을 유도할 수 있다는 문제점이 있다. 이를 해결하기 위해 본 논문은 언어 정보의 중요도를 수치화하여 perplexity 기반의 문장 점수 계산에 반영하는 방법을 제안한다. 또한 고유명사가 자주 포함되어 있으며, 불필요한 수식어가 생략되는 경우가 많은 뉴스 기사 말뭉치로 BERT를 fine-tuning하여 문장 압축에 적절한 perplexity를 측정할 수 있도록 하였다. 영어 및 한국어 데이터에 대한 성능 평가를 위해 본 논문에서 제안하는 LI-Deleter와 비교 모델의 문장 압축 성능을 비교 실험을 진행하였고, 높은 문장 압축 성능을 보임을 확인하였다.

키워드 : 문장 압축, 언어 정보, 언어 모델, 펄플렉시티

※ 이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(No.2020R1A4A1018607)과 정보통신기획평가원의 지원(No.2013-0-00109, WiseKB: 빅데이터 이해 기반 자가학습형 지식베이스 및 추론 기술 개발)을 받아 수행된 연구임.
※ 이 논문은 2021년 한국정보처리학회 춘계학술발표대회에서 “언어 정보를 반영한 문장 점수 측정 기반의 문장 압축”의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : 경희대학교 컴퓨터공학과 석사후과정
†† 비 회 원 : 경희대학교 컴퓨터공학과 박사과정
††† 비 회 원 : 경희대학교 컴퓨터공학과 교수
Manuscript Received : June 30, 2021
First Revision : August 24, 2021
Accepted : September 14, 2021
* Corresponding Author : Seong-Bae Park(sbpark71@khu.ac.kr)

1. 서 론

문장 압축은 문장에서 불필요한 단어를 삭제하여 간결한 문장을 생성하는 자연어처리 분야의 태스크이다. 문장 압축은 문서 요약, 텍스트 단순화, 데이터 증강과 같이 다양한 태스크에 적용될 수 있어 활발히 연구되고 있다. 대부분의 문장 압축 연구는 입력 문장에서 삭제할 단어를 선택하여 문장을 압축하는데, 삭제할 단어를 어떻게 결정하느냐에 따라 규칙 기반 방식, 인공신경망 기반 방식으로 나누어진다.

규칙 기반 문장 압축은 파스 트리 재단을 통한 방법[1]이 대표적이다. 이 방법은 문법이 적절한 압축 문장을 생성하기 위해 사람이 정의한 문법 규칙을 사용하며, 규칙을 만족하지 않는 단어를 삭제한다. 파스 트리 트리밍 방법은 문장의 문법성을 충분히 고려하여 압축된 문장을 생성할 수 있다는 장점이 있지만, 사람이 모든 문법 규칙을 정의하는 데에 큰 비용이 든다는 문제점이 존재한다.

반면 딥러닝 기반의 문장 압축은 모델이 원본 문장과 압축 문장 사이의 관계를 학습하도록 함으로써 어떤 단어가 삭제되어야 할지 모델 스스로 결정하도록 하는 방법이다. 딥러닝 기반 문장 압축에는 LSTM을 기반으로 하는 방법[2]이 대표적이다. 이 방법은 문장을 구성하고 있는 단어들을 삭제할지, 삭제하지 않을지를 연속 분류하는 방식으로 문장을 압축한다. LSTM 기반의 문장 압축은 사람이 직접 정의한 압축 규칙 없이 좋은 문장 압축 성능을 보였지만, 문법적으로 적절하지 못한 문장을 생성할 확률이 높고 모델 학습을 위해 충분한 크기의 데이터셋이 필요하다는 문제점이 있다.

이러한 문제점을 해결하기 위하여 모델 학습을 위한 데이터가 없는 상황에서도 사용될 수 있는 모델인 Deleter[3]가 제안되었다. Deleter는 문장의 단어들을 임의로 삭제한 압축 후보 문장들을 생성한 뒤, 후보 문장들의 문장 점수 측정을 통해 어떤 후보 문장을 선택할지 결정하는 과정을 반복하여 문장을 압축한다. 이때, 후보 문장의 문장 점수를 측정하기 위해 사전 학습된 언어 모델 BERT를 사용하고, BERT로부터 계산된 perplexity를 문장 점수로 간주한다. 즉, Deleter는 사전 학습된 언어 모델이 존재할 경우, 학습 데이터 없이 문장을 압축할 수 있다는 장점이 있다.

전술한 장점에도 불구하고 Deleter의 문장 압축 성능은 사전 학습된 언어 모델에 의존적이기 때문에 파생되는 몇 가지 문제점이 있다. 첫 번째로, perplexity에는 단어들의 언어 정보가 반영되지 않는다. 두 번째로, BERT는 길이에 상관없이 일반적인 문장으로 사전 학습되었기 때문에, 일반적으로 압축된 문장에 존재하지 않아야 하는 불필요한 수식어가 포함된 문장의 perplexity를 항상 높게 계산하지는 않는다. 마지막으로 BERT는 고유명사와 같이 자주 등장하지 않은 단어가 포함된 문장의 perplexity를 과도하게 높게 계산한다. 이러한 문제점들로 인해 Deleter는 문장 압축을 수행할 때 삭제되어야 할 수식어가 올바르게 삭제되지 않거나, 원본 문장의 중요한 정보를 담고 있는 단어가 삭제되는 문제점이 존재한다.

본 논문에서는 Deleter의 첫 번째 문제점을 해결하기 위하여 단어의 품사, 구조, 개체명 정보의 중요도를 수치화한 뒤 문장 점수에 반영하는 방법을 제안한다. 또한 고유명사가 자주 등장하며 불필요한 수식어 사용을 최소화한 텍스트 데이터인 뉴스 기사 말뭉치를 통해 BERT를 fine-tuning 하여 Deleter의 두 번째, 세 번째 문제점을 완화하였다. 본 논문이 제안하는 모델을 LI-Deleter (Deleter with Linguistic Informaition)라 명명하였으며, Google compression dataset[4]과 한국어 문장 압축 데이터셋[5]에 대해 정량 평가 및 정성 평가를 수행하여 기존 Deleter 및 LSTM 모델을 사용한 지도학습 기반 문장 압축 모델과의 문장 압축 성능을 비교하였다.

본 논문의 다음과 같이 구성된다. 2장에서는 베이스 모델인 Deleter의 문장 압축 과정 및 perplexity 기반의 문장 점수 계산에 대해 구체적으로 설명한다. 3장에서는 LI-Deleter가 언어 정보의 중요도를 어떻게 수치화하며, 언어 정보의 중요도를 perplexity 기반의 문장 점수에 반영하는 방법을 구체적으로 설명한다. 4장에서는 실험을 위한 데이터셋과, 정량 평가 및 정성 평가를 위한 실험 설정에 대해 설명하고, 5장에서는 실험 결과 분석을 통해 LI-Deleter의 우수성을 보인다. 6장에서는 문장 압축 관련 연구에 대해 설명하며 마지막 7장은 결론과 향후 연구내용에 대해 다룬다.

2. Deleter

2.1 Deleter의 문장 압축

Deleter는 압축 후보 문장 생성 단계, 후보 문장 점수 계산 단계, 문장 압축 단계를 통해 문장의 단어를 점진적으로 삭제한다. 압축 후보 문장 생성 단계에서는 입력 문장에서 임의의 단어를 삭제한 압축 후보 문장들을 생성한다. 후보 문장 점수 계산 단계에서는 BERT를 사용해 모든 후보 문장의 점수를 계산한다. 문장 압축 단계에서는 점수가 가장 낮은 후보 문장을 선택해 문장을 압축한다. 선택된 압축 문장은 Deleter의 새로운 입력 문장으로 하여 앞선 세 단계의 문장 압축 과정을 종료 조건을 만족할 때까지 반복한다. 문장 점수에 대한 자세한 설명은 2.2장에서 서술하며, 2.3장에서는 Deleter의 동작이 종료되는 조건에 대해 설명한다.

2.2 BERT 기반의 문장 점수

m개의 단어로 구성된 문장 $W=(w_1,w_2,\cdots,w_m)$으로부터 n개의 단어 $D=(d_1,d_2,\cdots d_n)$가 삭제된 압축 후보 문장 $T=(t_1,t_2,\cdots,t_{m-n})$의 문장 점수는 Equation (1)과 같이 계산되며, perplexity를 기반으로 하는 Average Perplexity Score (AvgPPL)라고 명명된다.

$$AvgPPL(T)=\exp\left(-\frac{1}{m}\left(\sum_{i}^{m-n}\log p(t_i \mid T_{\setminus t_i})+\sum_{j}^{n}\log p(d_j \mid W_{\setminus d_j})\right)\right) \quad (1)$$

이때 $T_{\setminus t_i}$는 압축 후보 문장 T에서 i번째 단어인 t_i를 제외한 단어들을 의미하며, $p(t_i \mid T_{\setminus t_i})$는 t_i의 우도를 의미한다. 또한 $W_{\setminus d_j}$는 원본 문장 W에서 D의 j번째 단어인 d_j를 제외한 단어들을 의미하며, $p(d_j \mid W_{\setminus d_j})$는 원본 문장 W에서 d_j의 우도를 의미한다. 따라서 후보 문장에 속한 단어들의 likelihood가 높을수록, 즉 언어 모델이 생성할 확률이 높은 문장일수록 후보 문장의 AvgPPL가 낮아진다.

2.3 Deleter의 종료 조건

Deleter의 후보 문장 점수 계산 단계에서 Equation (2)를 만족하는 후보 문장이 존재하지 않는다면, 압축 후보 문장을 생성하기 위해 삭제하는 단어의 개수를 증가시킨다.

$$\frac{AvgPPL_{candidate}}{AvgPPL_{input}} < 1 + \lambda \log(L_{root}) \quad (2)$$

이때 $AvgPPL_{input}$은 입력 문장의 점수를 의미하며, $AvgPPL_{candidate}$는 후보 문장의 점수를 의미한다. 또한 λ는 Deleter의 압축 정도를 결정하기 위한 하이퍼파라미터이며, L_{root}는 최초 입력 문장의 단어 개수를 의미한다. 삭제 단어 개수는 3까지 증가하며, 삭제 단어 개수가 4가 되면 동작을 종료한다.

3. LI-Deleter

Deleter는 BERT를 사용하여 계산한 압축 후보 문장들의 AvgPPL을 바탕으로 문장을 압축한다. 하지만 AvgPPL에는 단어들의 언어 정보가 반영되지 않기 때문에 Deleter가 압축한 문장에서 중요한 정보가 삭제되는 경우가 발생한다. 또한 BERT는 일반적인 문장으로 사전 학습되기 때문에, 문장의 정보를 전달하는데 불필요한 수식어의 likelihood가 특별히 낮게 계산되지는 않는다. 따라서 간결한 정보 전달에 불필요한 수식어가 포함된 압축 후보 문장이라고 하더라도 AvgPPL이 낮게 계산되어 압축 문장으로 선택될 수 있다. 그리고 고유명사의 경우 일반적인 문장에 자주 등장하지 않기 때문에 낮은 likelihood를 가진다. 따라서 고유명사가 포함된 문장의 AvgPPL이 높아지고, 압축 문장으로 선택되지 않을 가능성이 높아진다.

LI-Deleter는 후보 문장 점수 계산 단계에서 압축 후보 문장이 되기 위해 삭제된 단어의 언어 정보의 중요도를 수치화하고, 이를 해당 후보 문장의 AvgPPL에 더하여 준다. 이를 통해 중요한 정보를 담고 있는 단어가 삭제된 압축 후보 문장이 압축 문장으로 선택되는 빈도를 줄인다. 또한 LI-Deleter는 AvgPPL 계산을 위해 뉴스 기사 말뭉치를 사용해 fine-tuning한 BERT를 사용한다. 뉴스 기사 말뭉치는 고유명사가 자주 등장하며 불필요한 수식어가 절제된 텍스트이기 때문에, 이를 통해 fine-tuning한 BERT는 불필요한 수식어가 포함된 후보 문장이나 고유명사가 포함된 후보 문장의 AvgPPL을 올바르게 계산할 수 있다.

m개의 단어로 이루어진 문장 $W=(w_1, w_2, \cdots, w_m)$의 t번째 단어인 w_t가 삭제된 압축 후보 문장 $W_{\setminus t}=(w_1, w_2, \cdots, w_{t-1}, w_{t+1}, \cdots, w_m)$이 존재할 때, LI-Deleter의 후보 문장 점수 $Score(W_{\setminus t})$는 Equation (3)과 같이 계산된다.

$$Score(W_{\setminus t}) = AvgPPL(W_{\setminus t}) + LI(w_t) \quad (3)$$

이때 AvgPPL은 뉴스 기사 말뭉치로 fine-tuning된 언어 모델 BERT를 통해 계산한다. 또한 $LI(w_t)$는 w_t가 갖는 언어 정보의 중요도이며 Equation (4)와 같이 계산된다.

$$LI(w_t) = \alpha PI(w_t) + \beta SI(w_t) + \gamma EI(w_t) \quad (4)$$

Equation (4)의 $PI(w_t)$, $SI(w_t)$, $EI(w_t)$는 각각 w_t의 품사 정보(Part Of Speech Information), 의존 관계 정보(Dependency Information), 개체명 정보(Entity Information)의 중요도 점수이며, α, β, γ는 세 정보의 중요도 점수 사이의 가중치를 의미한다. 이때 품사 정보, 의존 관계 정보, 개체명 정보는 [4]에서 구축 및 공개한 데이터셋[1]에서 제공하는 품사, 의존 관계, 개체명 정보를 사용하였으며, 각 정보의 중요도는 Equation (5)와 같이 계산한다.

$$PI(w_t) = \frac{count_{target}(Part\ of\ Speech\ Info\ Tag(w_t))}{count_{source}(Part\ of\ Speech\ Info\ Tag(w_t))} \quad (5)$$

$$DI(w_t) = \frac{count_{target}(Dependency\ Info\ Tag(w_t))}{count_{source}(Dependency\ Info\ Tag(w_t))}$$

$$EI(w_t) = \frac{count_{target}(Entity\ Info\ Tag(w_t))}{count_{source}(Entity\ Info\ Tag(w_t))}$$

Equation (5)의 $count_{source}$와 $count_{target}$는 각각 말뭉치의 전체 원본 문장에 각 정보 태그가 등장한 횟수, 말뭉치의 전체 압축 문장에 각 정보 태그가 등장한 횟수를 의미한다. 즉 말뭉치의 원본 문장에 등장했던 정보 중 압축 문장에 자주 남아있는 정보일수록 높은 점수를 갖는다.

Equation (4)의 품사 정보 점수, 의존 관계 정보 점수, 개체명 정보 점수의 가중치인 α, β, γ를 구하기 위하여 단일 완전 연결 레이어로 구성된 인공신경망을 사용하였다. 인공신경망은 Fig. 1과 같이 Equation (5)에서 계산한 단어의 품사 정보 점수, 의존 관계 정보 점수, 개체명 정보 점수를 입력으로 받아 해당 단어가 압축 문장에 포함되어야 한다면 1, 포함되지 않아야 한다면 0에 가까운 값을 출력하도록 하는 이진 분류 모델이다. 최종적으로 학습된 완전 연결 레이어의 세 입력 값에 대한 가중치를 α, β, γ로 사용하였다.

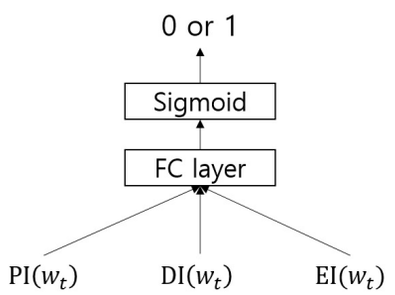


Fig. 1. Binary Classification Model for Word Deletion

1) https://github.com/google-research-datasets/sentence-compression

4. 실험 설정

4.1 모델 세부 설정

영어 및 한국어에 대한 LI-Deleter의 문장 압축 성능을 Deleter와 LSTM 기반의 문장 압축 모델 (LSTM base)과 비교하였다. Deleter와 LSTM base 모델의 소스코드가 공개되어있지 않아 논문의 설명을 따라 직접 구현하여 성능을 비교하였다. 영어에 대한 실험을 위해 LI-Deleter는 Deleter 모델과 같이 Hugging-Face에서 제공하는 사전 학습된 언어 모델 BERT를 사용하였으며 한국어에 대한 실험을 위해 한국전자통신연구원(ETRI)에서 제공하는 KorBERT를 사용하였다. 이때 Deleter는 문장 압축의 단위로 BERT의 토큰 단위인 sub-word를 사용하였는데, 단어보다 작은 단위인 sub-word 단위로 문장을 압축할 경우 의미가 변질될 수 있기 때문에 LI-Deleter는 한 단어에 포함된 모든 sub-word의 likelihood의 평균을 해당 단어의 likelihood로 사용하여 단어 단위의 문장 압축을 가능하도록 하였다.

4.2 데이터셋

LI-Deleter의 문장 압축 성능을 확인하기 위해 Google compression dataset을 사용하였다. Google compression dataset은 구글 뉴스 기사 데이터로부터 생성되었으며 문장과 정답 압축 문장, 각 문장의 품사 정보 태그, 의존 관계 정보 태그, 개체명 정보 태그를 제공하며 20만 쌍의 학습 데이터, 1만 쌍의 평가 데이터로 구성되어있다. LI-Deleter는 20만 쌍의 학습 데이터를 사용해 BERT fine-tuning 및 이진 분류 모델 학습을 수행하였고, Deleter의 평가 데이터 설정과 같이 1만 쌍의 평가 데이터 중 처음 1천 쌍을 사용하여 문장 압축 성능을 평가하였다.

LI-Deleter가 다양한 언어에 적용될 수 있는 모델임을 확인하기 위하여, 한국어에 대한 문장 압축 실험 또한 수행하였다. 이를 위해 추상적 한국어 뉴스 제목 생성 연구[5]에서 사용한 한국어 문장 압축 데이터셋을 사용하였다. 한국어 문장 압축 데이터셋은 2017년 6월부터 2018년도 3월까지의 네이버 뉴스 중 스포츠, 연예, 경제, 사회 분야 기사를 크롤링 한 뒤, 기사의 첫 번째 문장을 원본 문장으로 하여 주석자가 기사의 내용과 제목을 바탕으로 원본 문장으로부터 기사 내용을 이해하는데 덜 중요하다고 판단되는 단어를 삭제하여 압축 문장을 생성한 데이터셋이다. 하지만 해당 데이터셋은 뉴스 제목 생성을 위해 구축한 데이터셋이기 때문에, Table 1의 예시와 같이, 문장의 종결 어미가 명사형으로 수정되거나, 조사 중 일부분이 생략되었다.

본 논문은 삭제 기반 문장 압축을 수행하기 때문에, 이에 맞추어 명사형 종결어미를 서술형으로 되돌리고, 생략된 조사를 복원하는 후처리를 진행하였다. 또한 후처리 과정에서 종결어미를 명사형에서 서술형으로 변경하였을 때 문장이 어색해지는 경우를 필터링하여 최종 1,600쌍의 데이터셋을 사용하였다. 이 중 1,500쌍을 KorBERT fine-tuning 및 이진 분류 모델 학습에, 100쌍를 문장 압축 성능 평가에 사용하였다. 또한 한국어 문장의 언어 정보 태그를 얻기 위하여 ETRI 언어분석기를 사용하였다. 이때 한국어의 경우 영어와 달리 형태소 단위로 품사가 결정되기 때문에, 품사 정보 대신 단어 단위의 정보인 구문 정보를 사용하였다. ETRI 언어분석기가 제공하는 구문 정보, 의존 관계 정보, 개체명 정보 태그셋과 그에 대한 설명은 각각 Table 2, Table 3, Table 4에서 확인할 수 있다.

Table 1. Examples of Korean Compression Dataset

Input Sentence
1. 대한항공이 승점 60을 달성하며 2위 자리에 올랐다.
2. 도로공사가 3년 만의 정규시즌 우승을 확정했다.
Gold Compression
1. 대한항공이 2위 자리에 오름
2. 도로공사가 정규시즌 우승 확정
Post-processed Gold Compression
1. 대한항공이 2위 자리에 올랐다.
2. 도로공사가 정규시즌 우승을 확정했다.

Table 2. Korean Phrase Information Tag Set and Its Description

Tag	Description
NP	Che-eon (Noun, Pronoun, Number)
VP	Yong-eon (Verb, Adjective, Auxiliary Verb)
AP	Adverb phrase
VNP	Positive Jijeong-sa phrase (Noun + -ida)
DP	Determiner phrase
IP	Interjection phrase

Table 3. Korean Dependency Information Tag Set and Its Description

Tag	Description
SBJ	Subject
OBJ	Object
MOD	Che-eon Modifier
AJT	Yong-eon Modifier
CMP	Complement
CNF	Conjunction

Table 4. Korean Named Entity Information Tag Set and Its Description

Tag	Description	Tag	Description
PS	Person	PT	Plant
LC	Location	QT	Quantity
OG	Organization	FD	Study field
AF	Artifacts	TR	Theory
DT	Date	EV	Event
TI	Time	MT	Material
CV	Civilization	TM	Term
AM	Animal		

4.3 정량평가 및 정성평가

정량평가를 위해 Deleter와 LSTM base, 그리고 LI-Deleter을 사용하여 평가 데이터셋의 문장을 압축한 뒤, 정답 데이터셋을 통해 F1 score를 측정하였다. 이때 지도학습 기반 문장 압축 모델인 LSTM base의 경우 학습 데이터의 수에 따른 성능 변화 실험을 통해 학습 데이터가 충분하지 않은 상황에서의 문장 압축 성능을 LI-Deleter와 비교하였다. 또한 한국어의 경우 정보 전달력 및 가독성에 대한 정성평가를 수행하여 LI-Deleter가 지도학습 기반의 문장 압축 모델인 LSTM base보다 적절한 압축 문장을 생성함을 확인하였다. 구체적으로 100개의 평가 데이터 중 무작위로 추출한 50개 문장에 대해 LSTM base와 LI-Deleter 모델을 사용해 압축 문장을 생성하였고, 어떤 모델이 생성한 압축 문장인지 알려주지 않은 상태로 두 명의 평가자에게 정보 전달력 및 가독성을 기준으로 어떤 압축 문장이 더 적절한지 평가하도록 하였다.

5. 결과 및 분석

Table 5를 통해 Google compression dataset에 대한 문장 압축 실험 결과를 확인할 수 있다. Deleter, LI-Deleter, LSTM base 모델은 각각 0.39, 0.39, 0.38의 압축률을 보여 문장 압축을 위해 비슷한 수의 단어를 삭제하였다. LI-Deleter의 F1 score는 52.13으로 44.19의 Deleter보다 7.94만큼 높은 성능을 보였지만, 지도학습 기반의 문장 압축 모델인 LSTM base의 F1 score인 71.70보다는 크게 낮은 성능을 보였다. 하지만 학습 데이터셋 크기에 따른 LSTM base 모델의 성능 변화 실험 결과인 Table 6를 보면, 학습 데이터셋의 크기가 10만 개보다 적을 경우, 오히려 LI-Deleter의 성능이 뛰어남을 확인하였다.

Table 7을 통해 한국어 문장 압축 데이터셋에 대한 문장 압축 실험 결과를 확인할 수 있다. Deleter, LI-Deleter, LSTM base 모델은 각각 0.56, 0.58, 0.58의 압축률을 보여 비슷한 수의 단어를 삭제하였다. 또한 Deleter는 58.92, LI-Deleter는 64.52, LSTM base는 69.26의 F1 score를 보여 영어 데이터에 대한 실험과 마찬가지로 LI-Deleter는 Deleter보다 뛰어난 성능을 보였으나 지도학습 기반의 문장 압축 모델인 LSTM base보다 낮은 성능을 보였다. 하지만 Table 8에서 확인할 수 있듯이 정보 전달력(Informativeness)과 가독성(Readability)에 대한 정성평가 결과, LI-Deleter가 LSTM base 모델보다 정보 전달력에선 0.72, 가독성에선 0.93의 승률을 보여 문장의 중요 의미를 더 잘 보존하고, 문법적으로 더 적절한 압축 문장을 생성함을 확인할 수 있다. 이때 괄호 안의 수치는 kappa 상관계수 값으로 두 명의 평가자가 얼마나 일치하는 답변을 보였는지를 측정한 값이다.

Table 9와 Table 10은 각각 Google compression dataset과 한국어 문장 압축 데이터셋의 테스트 데이터에 대한 좋은 문장 압축 예시를 보여준다. 영어 데이터와 한국어 데이터에 대한 좋은 문장 압축 예시의 경우, LI-Deleter가 입력 문장의 정보를 왜곡하지 않고 길이를 잘 압축한 것을 확인할 수 있다.

Table 5. F1 Results on the Google Compression Dataset

Model	F1 score	Compression Rate
Deleter	44.19	0.39
LI-Deleter	52.13	0.39
LSTM base	71.70	0.38

Table 6. F1 Results According to the Size of the Training Data

Size (1k)	20	40	60	80	100	120	140	160	180	200
F1 score	39.04	45.83	48.62	50.75	52.65	57.18	60.06	63.91	69.77	71.70

Table 7. F1 Results on the Korean Compression Dataset

Model	F1 score	Compression Rate
Deleter	58.92	0.56
LI-Deleter	64.52	0.58
LSTM base	69.26	0.58

Table 8. Human Evaluation for Korean Compression Dataset

Evaluation	Informativeness	Readability
win rate (kappa)	0.72 (0.90)	0.93 (0.85)

Table 9. Good Example of a Model Prediction for Google Compression Dataset

Input sentence
Paris Saint-Germain player Zlatan Ibrahimovic has urged Manchester United striker Wayne Rooney to join him at PSG .
Gold compression
Paris Saint-Germain player Zlatan Ibrahimovic has urged Wayne Rooney to join .
LI-Deleter compression
Ibrahimovic urged Wayne Rooney to join him at PSG.

Table 10. Good Example of a Model Prediction for Korean Compression Dataset

Input sentence
유은혜 더불어민주당 의원이 대학등록금의 카드 수수료를 1% 이내로 제한하는 내용의 고등교육법 개정안을 대표 발의했다고 1일 밝혔다.
Gold compression
유은혜 더불어민주당 의원이 대학등록금의 카드 수수료를 1% 이내로 제한하는 고등교육법 개정안을 대표 발의했다고 밝혔다.
LI-Deleter compression
유은혜 더불어민주당 의원이 대학등록금의 수수료를 제한하는 고등교육법 개정안을 발의했다고 밝혔다.

Table 11. Bad Example of a Model Prediction for Google Compression Dataset

Input sentence
A new lieutenant was introduced and two deputies who helped save a man's life were honored Monday at the Bell County Jail.
Gold compression
A new lieutenant was introduced and two deputies were honored.
LI-Deleter compression
lieutenant and two deputies helped man's life.

Table 12. Bad Example of a Model Prediction for Koeran Compression Dataset

Input sentence
1일 오후 9시8분쯤 서울 노원구 상계동 한신아파트 인근 수락산에서 불이 나 소방당국이 진화에 나섰다.
Gold compression
1일 서울 노원구 상계동 수락산에서 불이 나
LI-Deleter compression
한신아파트 수락산에서 불이 진화에 나섰다.

Table 11과 Table 12는 각각 Google compression dataset과 한국어 문장 압축 데이터셋의 테스트 데이터에 대한 나쁜 문장 압축 예시를 보여준다. 나쁜 문장 압축 예시의 경우, LI-Deleter가 입력 문장의 정보를 왜곡하여, 올바르지 않은 압축 문장을 생성하였다. 이는 LI-Deleter는 절에 대한 정보 대신 단어 단위의 언어 정보만을 사용하기 때문에 각 절의 주어와 서술어 관계가 파괴되어 절과 절 사이의 경계가 모호한 압축 문장을 생성하기 때문으로 분석된다. LI-Deleter의 이러한 한계점을 극복하기 위해서, 입력 문장이 여러 개의 절로 구성된 문장인 경우 각 단어가 어떤 절에 포함되어 있는지를 파악하고, 어떤 절이 문장에서 중요한 절 인지를 결정할 수 있는 방법에 대한 연구가 추가적으로 필요할 것으로 보인다. 본 논문에서는 이러한 연구를 향후 연구 주제로 두었다.

Table 13은 세 개의 한국어 예시 문장에 대해 LSTM base 모델과 LI-Deleter가 생성한 압축 문장을 두 명의 평가자가 평가한 결과를 보여준다. 이를 통해 정보 전달력과 가독성에 대한 정성 평가가 어떻게 이루어졌는지에 대해 알 수 있다. 첫 번째 예시의 경우, 두 평가자 모두 LI-Deleter가 생성한 압축 문장이 LSTM base 모델이 생성한 압축 문장보다 원본 문장의 중요한 정보를 잘 보존하고 있고, 가독성이 뛰어나다고 평가하였다. 두 번째 문장의 경우 정보 전달력에서는 두 평가자의 평가가 갈렸지만, 가독성에 대해서는 두 평가자 모두 LI-Deleter가 생성한 압축 문장이 더 뛰어나다고 평가하였다. 세 번째 문장의 경우, 정보 전달력에서는 두 평가자가 모두 LSTM base 모델이 생성한 압축 문장이 뛰어나다고 평가하였고, 가독성의 경우 LI-Deleter가 생성한 압축 문장이 뛰어나다고 평가하였다.

Table 13. Examples of Human Evaluation for Korean Compression Dataset

Input sentence				
1. 예능드라마 〈최고의 한방〉에 김준호-데프콘이 카메오로 출연한다.				
2. 국토교통부는 메르세데스벤츠코리아가 E클래스 등 1,071대의 차량에 대한 리콜을 실시한다고 1일 밝혔다.				
3. 아이돌그룹 빅뱅 멤버 탑이 대마초 혐의로 적발됐다.				
Model prediction (LSTM base)				
1. 드라마 한방〉 김준호 데프콘이 카메오로 출연한다.				
2. 메르세데스 1071대 리콜 실시한다고 밝혔다.				
3. 빅뱅 멤버 대마초 혐의로 적발됐다.				
Model prediction (LI-Deleter)				
1. 〈최고의 한방〉에 김준호-데프콘이 출연한다.				
2. 메르세데스벤츠코리아가 E클래스 차량에 대한 리콜을 실시한다고 밝혔다.				
3. 빅뱅 멤버 탑이 적발됐다.				
Gold compression				
1. 〈최고의 한방〉에 김준호-데프콘이 카메오로 출연한다.				
2. 메르세데스벤츠코리아가 1071대 차량에 대한 리콜을 실시한다고 밝혔다.				
3. 탑이 대마초 혐의로 적발됐다.				
Human evaluation result				
	Evaluator #1		Evaluator #2	
#	Informativeness	Readability	Informativeness	Readability
1.	LI-Deleter	LI-Deleter	LI-Deleter	LI-Deleter
2.	LSTM base	LI-Deleter	LI-Deleter	LI-Deleter
3.	LSTM base	LI-Deleter	LSTM base	LI-Deleter

6. 관련 연구

문장 압축는 문서 요약의 성능 향상뿐 아니라[1,11], 모바일환경과 같이 제한된 공간에서 텍스트 정보를 빠르고 간결하게 전달하기 위해 수행된다. 대부분의 문장 압축 연구는 문장에서 불필요한 단어를 삭제하는 방식으로 문장을 압축하는데, 문법적으로 적절한 압축 문장을 생성하기 위하여 많은 문장 압축 연구들[6-8]은 파스 트리 트리밍 기반의 방법을 사용하였다. 또한 시퀀스-투-시퀀스 모델이 기계 번역 등 자연어 처리 분야의 다양한 태스크[9,10]에서 좋은 성능을 보임에 따라 이를 문장 압축 연구에 적용하려는 시도들이 존재했고, 문장 압축을 시퀀스-투-시퀀스 모델을 통한 연속 이진 분류 태

스크로 풀어낸 문장 압축 연구[2]가 뛰어난 성능을 보였다. 하지만 시퀀스-투-시퀀스 기반의 모델은 학습을 위해 거대한 양의 데이터를 필요로 한다는 문제점이 존재한다. 모델 학습을 위한 대규모 영어 문장 압축 데이터셋 생성 연구[4] 및 이를 기반으로 한 한국어 문장 압축 데이터셋 생성 연구[12] 등 학습 데이터 자동 구축을 위한 연구들이 수행되었지만, 결국 시스템이 생성한 압축 문장을 사용하기 때문에, 실제 전문가가 생성한 압축 문장과는 차이가 존재한다.

Deleter[3]는 비지도학습 기반의 문장 압축 모델로, BERT를 통해 문장 점수를 계산하고, 어떤 단어가 삭제되었을 때 문장의 점수가 증가하는지를 바탕으로 문장의 단어를 삭제해 나간다. Deleter는 문장 압축을 위해 데이터 학습이 필요하지 않다는 장점이 있으며 perplexity 기반의 문장 점수를 사용하기 때문에 매끄러운 압축 문장을 생성할 수 있다. 하지만 Deleter는 문장의 단어들의 언어 정보를 문장 압축에 활용하지 않기 때문에 중요한 정보가 삭제된 압축 문장을 생성하는 문제점이 존재하며, 문장 점수 측정을 BERT에 의존하기 때문에 고유명사 또는 불필요한 수식어가 포함된 문장의 점수를 올바르지 못하게 측정하는 경우가 존재한다.

LI-Deleter는 단어들의 품사, 의존관계, 개체명의 중요도를 수치화한 뒤 문장 점수에 반영하여 문장 압축 시 중요한 언어 정보가 삭제되지 않도록 하였다. 또한 많은 고유명사를 포함하며, 간결한 정보 전달을 위해 불필요한 수식어 사용이 절제된 텍스트 데이터인 뉴스 기사 말뭉치를 통해 BERT를 fine-tuning하여 적절한 문장 점수를 계산할 수 있도록 하였다.

7. 결 론

본 논문은 문장 점수 측정 기반의 문장 압축 모델인 Deleter의 문장 점수 계산에 단어의 품사 정보, 의존 관계 정보, 개체명 정보의 중요도를 반영하는 방법을 제안하였다. 또한 고유명사가 자주 등장하며 불필요한 수식어 사용을 최소화한 말뭉치인 뉴스 기사 데이터를 통해 BERT를 fine-tuning하여 문장 압축에 보다 적절한 문장 점수를 계산할 수 있도록 하였다. 제안한 방법들을 사용한 LI-Deleter는 영어 및 한국어 데이터셋에 대한 문장 압축 성능을 평가 결과 Deleter보다 뛰어난 성능을 보였다. 반면 LI-Deleter는 여러 개의 절로 구성된 복잡한 구조의 문장의 경우 문장 압축 성능이 떨어짐을 확인하였고, 향후 연구로 절의 깊이에 대한 정보를 문장 점수에 반영하는 방법을 개발하고자 한다.

또한 LI-Deleter의 경우 추출된 언어 정보에 오류가 있을 경우 이에 따라 모델의 성능이 좌우되는 오류 전이에 대한 문제점이 존재한다. 따라서 이에 대한 대안이 필요하며, 본 논문에서는 이를 향후 연구 주제로 두어, 지속적으로 연구하고자 한다. 우선 외부 모듈을 사용하여 추출한 언어 정보를 활용하는 대신, 문장의 단어 사이의 부모-자식 관계와 같은 언어 정보를 예측할 수 있는 모듈을 설계하고, 특정 정보에 대한 모듈의 예측 점수를 해당 정보 점수의 가중치로 사용하여 특정 정보에 오류가 존재할 확률을 해당 정보 점수에 반영하는 방법에 대해 연구하고자 한다.

References

[1] H. Jing, "Sentence Reduction for Automatic Text Summarization," In *Proceedings of the 6th Applied Natural Language Processing Conference*, Seattle, pp.310-315, 2000.

[2] K. Filippova, E. Alfonseca, C. Colmenares, L. Kaiser, and O. Vinyals, "Sentence compression by deletion with LSTMs," In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Lisbon, pp.360-368, 2015.

[3] T. Niu, C. Xiong, and R. Socher, "Deleter: Leveraging BERT to perform unsupervised successive text compression," *arXiv preprint arXiv:1909.03223*, 2019.

[4] K. Filippova and Y. Altun, "Overcoming the lack of parallel data in sentence compression," In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Seattle, pp.1481-1491, 2013.

[5] I. Jung, S. Choi, and S. Park, "Single sentence summarization with an event word attention mechanism," *Journal of Korean Institute of Information Scientists and Engineers*, Vol.47, No.2, pp.155-161, 2020.

[6] K. Knight and D. Marcu, "Statistics-based summarization-step one: Sentence compression," In *Proceedings of the Conference on Innovative Applications of Artificial Intelligence*, Texas, pp.703-710, 2000.

[7] T. Berg-Kirkpatrick, D. Berg-Kirkpatrick, and D. Klein, "Jointly learning to extract and compress," In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp.481-490, 2011.

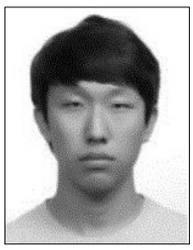
[8] K. Filippova and M. Strube, "Dependency tree based sentence compression," In *Proceedings of the Fifth International Natural Language Generation Conference*, pp.25-32, 2008.

[9] I. Sutskever, O. Vinyals, and Q. Le, "Sequence to sequence learning with neural networks," In *Proceedings of the Advances in Neural Information Processing Systems*, pp.3104-3112, 2014.

[10] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," In *Proceedings of the International Conference on Learning Representations*, San Diego, 2015.

[11] G. Lee, "A study on korean document summarization using extractive summarization and sentence compression," Ph.D. Dissertation. Chungnam National University, Korea, 2020.

[12] G. Lee, Y. Park, and K. Lee, "Building a Korean sentence-compression corpus by analyzing sentences and deleting words," *Journal of Korean Institute of Information Scientists and Engineers*, Vol.48, No.2, pp.193-194, 2021.



이 준 범

https://orcid.org/0000-0001-6935-4446
e-mail : jblee9410@khu.ac.kr
2020년 경희대학교 컴퓨터공학과(학사)
2021년 경희대학교 컴퓨터공학과(석사)
2021년 ~ 현 재 경희대학교 컴퓨터공학과 석사후과정
관심분야 : 연속 학습, 자연언어 처리

김 소 언

https://orcid.org/0000-0002-6395-8246
e-mail : sekim0211@khu.ac.kr
2020년 경희대학교 컴퓨터공학과(학사)
2021년 경희대학교 컴퓨터공학과(석사)
2021년 ~ 현 재 경희대학교 컴퓨터공학과 박사과정
관심분야 : 대화 모델, 자연언어 처리

박 성 배

https://orcid.org/0000-0002-6453-0348
e-mail : sbpark71@khu.ac.kr
1994년 한국과학기술원 전산학과(학사)
1996년 서울대학교 컴퓨터공학부(석사)
2002년 서울대학교 컴퓨터공학부(박사)
2004년 ~ 2018년 경북대학교 컴퓨터학부 교수
2018년 ~ 현 재 경희대학교 컴퓨터공학과 교수
관심분야 : 머신 러닝, 텍스트 마이닝, 자연언어 처리