Journal of The Korean Association
of Information Education Vol. 20, No. 4,
August 2016, pp. 333-340

http://dx.doi.org/10.14352/jkaie.2016.20.4.333

초등 글쓰기 교육을 위한 유사 문장 자동 선별

박영기
서울대학교 컴퓨터공학부

요 약

자신이 쓴 문장과 유사한 문장을 살펴보는 것은 초등 글쓰기 교육을 위한 효과적인 방법 중 하나이지만, 매번 글을 쓸 때마다 교사의 지도가 필요하기 때문에 현실적으로 활용하기 쉽지 않다. 본 논문에서는 이 한계를 극복하기 위해 컴퓨터가 자동으로 자신이 쓴 문장과 유사한 문장을 실시간으로 선별해 주는 방법을 제안한다. 이 방법은 단어의 구성 성분을 쪼개는 단계, 쪼갠 단어를 입력으로 활용하여 인코더-디코더 모델을 학습하는 단계, 모델을 통해 얻어낸 추상화된 문장을 활용해 검색하는 단계로 구성된다. 실험 결과, 작은 규모의 데이터에 대해 75%의 정확도를 보임으로써 실용화 가능성이 높은 것으로 나타났다. 이 방법을 통해 학생들은 자신의 어색한 문장을 교정하거나 새로운 표현을 익히고 싶은 경우 다른 사람이 작성한 좋은 예문을 쉽게 참조할 수 있어 자신의 글쓰기 능력을 향상시키는 데에 큰 도움이 될 것으로 기대된다.

키워드 : 초등 글쓰기 교육, 자동 유사 문장 선별, 인코더-디코더 모델

Automatic Selection of Similar Sentences for Teaching Writing in Elementary School

Youngki Park
School of Computer Science and Engineering, Seoul National University

ABSTRACT

When elementary students write their own sentences, it is often educationally beneficial to compare them with other people's similar sentences. However, it is impractical for use in most classrooms, because it is burdensome for teachers to look up all of the sentences written by students. To cope with this problem, we propose a novel approach for automatic selection of similar sentences based on a three-step process: 1) extracting the subword units from the word-level sentences, 2) training the model with the encoder-decoder architecture, and 3) using the approximate k-nearest neighbor search algorithm to find the similar sentences. Experimental results show that the proposed approach achieves the accuracy of 75% for our test data.

Keywords : Automatic Selection of Similar Sentences, Teaching Writing in Elementary School, Encoder-decoder Architecture

논문투고 : 2016-06-15
논문심사 : 2016-06-15
심사완료 : 2016-06-24

1. 서론

글쓰기는 초등 교육을 통해 필수적으로 학습해야 하는 과정이다. 국어와 영어 등 언어 교과목과 가장 관련이 있지만, 모든 교과목에서 필요로 하는 기본 교육이기 때문에 그 중요성이 매우 크다. 최근에는 수학적 글쓰기 또는 과학적 글쓰기[13]에 대한 중요성도 강조되고 있고, 컴퓨터를 통해 글쓰기를 학습하는 방법론도 다양하게 제시되고 있다[2].

가장 대표적인 글쓰기 교육 방법은 (1) 사전을 활용하거나, (2) 학생들이 틀리기 쉬운 표현들을 요약하여 가르치거나, 또는 (3) 예문을 통해 학습하는 방법 등이 있다[4]. 현 국어 교육과정에서는 첫 번째, 두 번째 방식이 더 널리 쓰이는데, 예를 들어 '듣기말하기쓰기' 1학년 교과서에서는 '지난 여르메 한국에 왔는데 벌써 일 년이 지났다.'와 같은 문장을 교정하는 내용을 학습한다. 본 논문은 예문을 통해 글쓰기를 학습하는 방법에 초점을 맞춘다. 예를 들어 만약 학생이 글을 썼을 때 그것과 유사한 문장을 제시해줄 수 있다면, 학생은 자신이 만든 문장을 바탕으로 새로운 글쓰기 방법을 학습할 수 있기 때문에, 내적 문법(mental grammar)을 재구성할 수 있다는 점에서 매우 효과적이다[15]. 그러나 이 방법은 학생이 글을 쓸 때마다 지도가 필요하기 때문에, 실제로 적용하기에는 제한적일 수밖에 없다. 만약 컴퓨터를 통해 유사한 문장을 찾아줄 수 있다면, 많은 비용을 들이지 않고도 효과적인 교육 방법이 될 수 있을 것이다.

기존의 컴퓨터를 활용해 유사 문장을 찾는 기술은 대체로 문법적 유사도에 기반하였다. 예를 들어 정보 검색(Information Retrieval) 분야에서는 동일 키워드가 많이 등장할수록 유사한 문장이라 보았는데, '밥을 먹었다'와 '밥을 좋아한다'는 하나의 공통된 키워드가 존재하기 때문에 유사하다고 판단하는 식이다. 또, 어떤 키워드는 다른 키워드에 비해 더 중요할 수 있으므로, 더 중요한 키워드가 공통적으로 존재할 경우 더 유사하다고 판단할 수도 있다[3][6][7][8][14]. 기계 번역 분야에서는 키워드의 연속성을 중요시하여, 연속된 키워드가 중복될 경우 더 유사하다고 보았다[5]. 예를 들어 '나는 밥을 먹었다'는 '나는 먹는 것을 좋아한다'라는 문장보다 '나는 밥을 좋아한다'라는 문장과 더 유사하다고 판단하는데, 왜냐하면 '나는 밥을'이라는 연속된 키워드가 일치하기 때문이다. 그러나 의미적 유사도를 고려하지 않기 때문에, 글쓰기 교육적 관점에서는 그 활용도가 제한될 수밖에 없다.

본 논문에서는 문법적/의미적 유사도를 모두 고려할 수 있는 딥 러닝(Deep Learning)에 기반한 기법을 활용함으로써 학생들에게 적절한 예문을 추천해 주는 방식을 제안한다. 이 방법은 단어의 구성 성분을 쪼개는 단계, 쪼갠 단어를 입력으로 활용하여 인코더-디코더 모델을 학습하는 단계, 모델을 통해 얻어낸 추상화된 문장을 활용해 검색하는 단계로 구성된다. 본 논문의 구성은 다음과 같다. 2절에서는 유사 예문을 통한 글쓰기 교육의 이론적 배경을 소개하고, 3절에서는 컴퓨터를 활용해 유사 예문을 선별하는 방법을 설명한다. 4절에서는 컴퓨터가 얼마나 유사 예문을 잘 찾아줄 수 있는지를 검증하고, 5절에서 결론 및 향후 연구 방향을 제시한다.

2. 이론적 배경

현 초등 국어 교육과정은 어순, 문장 성분, 문장부호, 꾸미는 말, 문장의 종류, 호응 관계, 문장의 확장/축소, 문장 고치기 등에 대해 가르치는 것을 목표로 한다[4]. 예를 들어 어순과 관련해서는 '나는 먹었다 밥을'이라고 쓰는 대신에 '나는 밥을 먹었다'라고 표현하는 법을 가르칠 수 있고, 문장 성분과 관련해서는 '친구에게 주었다'라는 표현보다 '나는 친구에게 선물을 주었다'라는 표현이 모든 서술어가 쓰인 표현임을 알려줄 수 있다. 이와 같은 교육 방식은 학생들이 틀리기 쉬운 문법 및 어휘를 가르치는 방식으로 요약될 수 있는데, 글쓰기가 아닌 다른 분야의 교육에서도 가장 널리 사용된다.

또 다른 글쓰기 교육법으로, 자신의 쓴 글을 교사가 교정하는 방식이 있다. Thornbury[15]는 학생이 직접 작성한 문장을 교정하는 방식으로 교사가 원하는 문장을 만들어가는 TTT 모형을 제시했는데, 이 방법은 Chomsky가 제시한 내적 문법(mental grammar)을 재구성할 수 있는 방식이어서 효과적 교육 방법이라 하였다. 비슷한 방식으로, Pham[11]은 학생이 쓴 글에 대해 바꿔쓰기(paraphrase)를 수행함으로써 효과적인 글쓰기 학습이 가능하다고 했다. 이때의 바꿔쓰기는 단어 수준 및 문장 수준의 바꿔쓰기를 모두 포괄하는 개념이다.

Thornbury와 Pham이 제안한 것과 같은 교육 방법은 그 효과가 입증되었지만, 교사의 적극적 개입을 필요로 하기 때문에 제한된 인력과 시간으로는 수행하기 어려운 단점이 있다. 중, 고등학생이나 어른들을 대상으로 가르칠 때에는 동료들을 활용하여 자신의 글을 교정하는 방식을 활용할 수도 있으나, 초등학생의 경우 직접 지도해 주는 방식을 대체하기는 어렵다.

본 논문에서는 최신 컴퓨터 기술을 활용하여 위 글쓰기 교육법을 작은 비용으로 효과적으로 보조할 수 있다고 보았다. 컴퓨터를 이용해 학생들이 어떤 문장을 입력하면, 그 문장과 가장 유사한 문장을 찾아주는 식이다. 이때 유사하다는 것은 문법적으로 유사할 수도 있고, 의미적으로 유사할 수도 있고, 아니면 문장에서 핵심적인 역할을 하는 일부 어휘만 유사할 수도 있다. 학생에 따라, 그리고 교육의 목표에 따라 어떤 유사한 문장을 보여주는 것이 정답인지는 달라질 수 있기 때문에, 컴퓨터가 교사보다 더 좋은 문장을 추천하는 것은 어려운 일이다. 그러나 컴퓨터는 교사보다 더 많은 표현들을 알고 있기 때문에, 다양성 측면에서는 교육적 효과가 더 클 수도 있다.

문장을 표현하는 방법은 매우 다양하기 때문에, 컴퓨터가 유사한 문장을 찾아준다는 것은 쉽지 않다. 특히 문법적으로 유사한 문장을 찾는 것은 자연어 처리 분야에서 오랜 기간 연구되어 왔지만 의미적으로 유사한 문장을 찾는 것은 해결하기 어려운 과제로 여겨져 왔다. 그렇지만 최근 딥 러닝 기술의 발달로 의미적으로 유사한 문장을 효과적으로 찾는 연구가 진행되고 있다. 본 논문에서는 그중에서도 인코더-디코더 아키텍처를 이용해 유사한 문장을 효과적으로 찾는 기법에 기반하여 학생들이 작성한 문장과 유사한 문장을 찾는 방법을 제시할 것이다. 이를 통해 교육적 목표를 얼마나 달성할 수 있을지를 실험을 통해 분석할 것이다.

3. 유사 문장의 자동 선별 방법

본 논문에서는 유사 문장을 자동 선별하는 알고리즘을 제안한다. 이 방법은 (Fig. 1)에서 나타낸 바와 같이 '데이터 전처리', '인코더-디코더 학습', '근사 k-인접 이웃 탐색' 등 크게 3가지 단계로 이루어진다. 가장 첫 번째 단계는, 어떤 문장이 입력으로 주어지면 그것에 대해 전처리 과정을 수행하는 것이다. 두 번째 단계는, 전처리된 문장이 '인코더-디코더' 아키텍처의 입력으로 들어가는데, 이때 인코더에 의해 추상화된 문장 벡터(sentence embedding vector)가 만들어진다. 이 추상화된 문장 벡터는 디코더의 입력으로 주어짐으로써 '인코더-디코더' 아키텍처의 신경망을 학습하게 된다. 세 번째 단계는, 학습된 신경망에 기반하여 만들어진 추상화된 문장 벡터를 이용하여, 유사한 다른 문장 벡터를 찾는 것이다. 이때 많은 데이터에 대해 빠른 검색을 하기 위해 시그니처 벡터(signature vector)를 생성하게 된다. 위 각 단계에 대해 3.1절부터 3.3절에서 상세히 설명한다.

3.1. 데이터 전처리

유사 문장을 선별하는 모델을 학습할 때 사용되는 데이터는 문장 집합이다. 문장은 단어들의 순열인데, 단어의 종류는 언어에 따라 다르지만 보통 몇 십만 개에 달한다. 이때 단어의 종류가 많으면 크게 2가지 문제가 발생한다. 첫째, 단어를 표현하는 벡터의 크기가 커지기 때문에 메모리 문제가 발생한다. 둘째, 데이터의 양이 충분하지 않을 경우 과적합(overfitting) 문제가 발생할 수 있다.

본 논문에서는 단어를 준단어(subword)[12] 형태로 치환함으로써 단어의 종류를 줄이면서 용이한 학습이 가능하게 하였다. 준단어는 단어를 더 작은 단위로 분리한 단위인데, 어떤 단어의 일부분이 학습에 사용될 문장들에 자주 나타났을 경우 준단어로 분리된다. 예를 들어 '준단어'라는 단어는 '준', '단어'로 분리될 수 있고, '학교에'라는 단어는 '학교', '에'로 분리될 수 있다. 준단어를 통해 학습된 모델을 이용하면 그 결과물도 준단어로 나올 수 있으나, 간단한 후처리를 통해 쉽게 단어로 조합이 가능하므로 활용 시 문제가 발생하지 않는다.

3.2. 인코더-디코더 학습

본 논문에서는 bahdanau가 제안한 인코더-디코더 아키텍처(encoder-decoder architecture)[1]를 이용하여 유사 문장을 학습한다. 이 아키텍처는 인코더와 디코더를 학습시키는 것이 목적으로, 원 논문에서는 기계 번역을

하기 위한 용도로 활용되었으나, 유사 문장을 찾는 데에도 사용될 수 있다.

이 아키텍처에서 인코더는 준단어의 순열을 이용하여, 벡터로 표현되는 추상화된 문장 정보(sentence embedding vector)를 만드는 역할을 한다. 예를 들어 <나, 는, 밥, 을, 먹었, 다>라는 준단어의 순열을 입력으로 받는다면 그 결과물은 (1.037, 5.207, -1.572, ..., 2.713, -1,423, -3.759)와 같은 고차원 벡터 형태의 추상화된 문장 정보가 될 것이다. 만약 입력으로 주어지는 문장이 유사하다면, 인코더를 통해 만들어지는 추상화된 문장 정보의 유클리디안 거리(Euclidean distance)가 짧다는 것이 주요 특징이다. 즉, 추상화된 문장 정보가 있으면 유사한 문장을 찾아내는 데에 활용할 수 있다. 이에 대해서는 3.3절에서 자세히 설명한다.

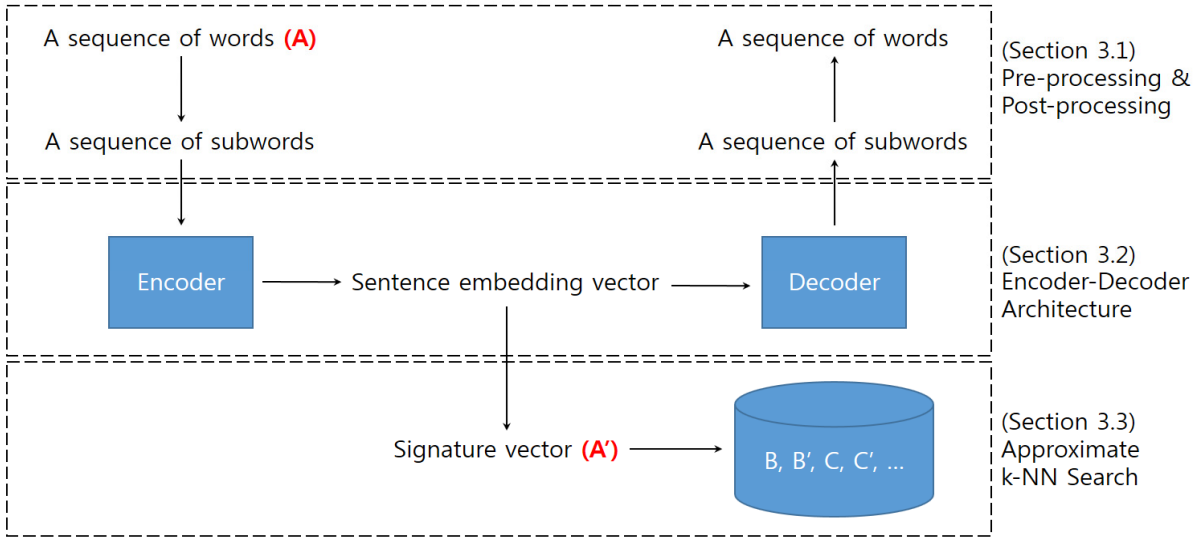
디코더는 기본적으로 인코더가 했던 것과 반대되는 일을 수행한다. 입력으로 추상화된 문장 정보를 받아, 준단어의 순열을 반환한다. 그러나 본 논문에서는 추상화된 문장 정보를 얻는 것이 목적이기 때문에 디코더는 테스트 시 사용할 필요가 없다. 그러나 인코더와 디코더는 모두 하나의 인공 신경망(artificial neural network)으로 구성되어 있기 때문에, 개별적으로 학습이 불가능하고 동시에 학습해야 하는 특징이 있다.

3.3. 근사 k-인접 이웃 탐색

인코더-디코더 아키텍처를 통해 추상화된 문장 정보를 얻었다면, 유클리디안 거리를 계산하여 유사한 문장을 예측할 수 있다. 예를 들어, A, B라는 문장을 이용해 추상화된 문장 정보 A', B'을 얻었다고 하자. 만약 A', B'의 유클리디안 거리가 짧다면, A, B 문장은 유사하다고 판단할 수 있다. 이때 다음과 같은 2가지 문제를 고려해야 한다.

- 유클리디안 거리가 얼마나 짧아야 유사하다고 판단할 수 있는지 결정할 방법이 필요하다.
- 추상화된 문장 정보는 고차원 벡터이기 때문에 유클리디안 거리를 계산하는 것은 매우 큰 소요 시간을 요구한다.

위 두 가지 문제를 해결하기 위해 근사 k-인접 이웃(k-NN) 검색 방법[9][10]을 사용한다. 즉, 가장 가까운 k개의 문장을 유사한 문장으로 간주하면서 첫 번째 문제를 해결하고, 근사적으로 k-NN을 찾는 방식으로 속도 문제를 개선하는 것이다. 근사 k-NN을 찾기 위해서는 추상화된 문장 정보를 나타내는 벡터의 차원을 축소하여 시그니처 벡터(signature vector)를 만드는 과정이 수반된다.



(Fig. 1) A Process of Finding Similar Sentences

<Table 1> Test Data (Query Sentences) for Measuring Precision@5

번호	질의 문장	선정된 이유
1	나는 너와 틀리다.	잘못된 어휘 선택
2	내 친구는 종아리가 얇다.	잘못된 어휘 선택
3	쓰레기 분리수거를 잘 해야지.	잘못된 어휘 선택
4	눈이 왔다. 길이 미끄럽다.	이어주는 말을 잘못 사용
5	나는 선물을 주었다.	서술어 중 하나가 생략
6	밥을 먹습니다 나는.	어순 뒤바뀜
7	그 물건 이리 줘.	문장 종류에 따라 어감이 달라짐
8	속도 위반 단속 취재하던 기자 차에 다쳐	문장부호 사용에 따라 완전히 다른 의미

4. 실험 결과

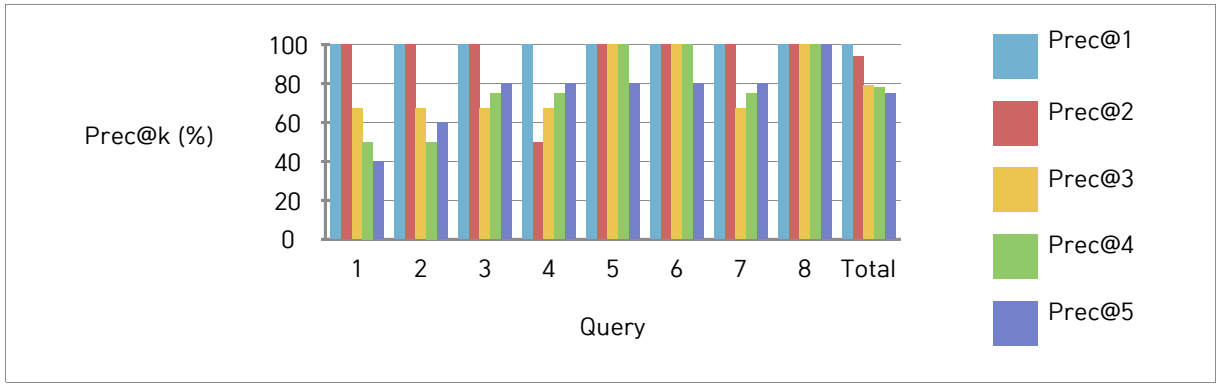
4.1. 실험 환경 구축

유사한 문장을 얼마나 잘 찾아주는지 검증하기 위해 다음과 같이 실험을 설계하였다. 먼저, 논문 [4]을 참조하여 초등학생들이 글쓰기를 할 때 오류를 범하기 쉽거나 유사한 표현을 찾아볼 필요가 있는 8개의 문장을 선정하였다. <Table 1>은 선정된 각 문장들과, 이 문장들이 선정된 이유에 대해 정리하였다.

그 다음 단계로, 각 질의마다 유사한 문장 5개와 유사하지 않은 문장 5개를 생성하였다. 이때 생성한 '유사한 문장'은 문법적 또는 의미적으로 유사한 문장일 수 있고, 그렇지 않더라도 학생들이 참조했을 때 도움이 되는 표현을 의미한다. 예를 들어 '나는 너와 다르다'와 '나는 너와 다른 점이 많다'는 문법적/의미적으로 유사한 문장이다. 다른 예로, '내 친구는 종아리가 가늘다'와 '구름을 이루는 층이 얇다'는 일반적으로 유사한 문장이라 보기는 어렵지만, '가늘다'와 '얇다'라는 혼동하기 쉬운 동사의 적절한 사용 예를 나타내기 때문에 유사한 문장이라 간주한다.

각 질의에 대해 10개씩 문장을 생성했으므로 총 80개의 문장이 있는데, 우리의 목표는 각 질의마다 이 문장들 중 우리가 생성한 5개의 유사 문장을 자동으로 선별하는 것이다. 성능을 측정하기 위해 Precision@k 척도를 사용한다. 즉, 컴퓨터가 유사 문장으로 k개를 선택했을 때, 그 중에서 몇 개 문장이 우리가 생성했던 유사 문장이냐를 기준으로 성능을 평가한다. 예를 들어 컴퓨터가 유사 문장으로 5개를 선택했는데, 그 중에서 우리가 생성했던 유사 문장이 3개가 있다면 Precision@5는 60%가 된다.

컴퓨터가 유사 문장을 선택할 때에는 3.1절, 3.2절에서 설명한 인코더-디코더 아키텍처와 준단어 테크닉을 통해 학습된 모델을 사용한다. 새 질의가 입력으로 주어지면, 디코더의 초기 hidden 상태를 구한 후, 이에 대해 유클리디안 거리를 계산하여 결과를 반환한다. 본 실험



(Fig. 2) Precision@k for Every Query

<Table 2> The Similar Sentences Automatically Selected by Our Algorithm

질의	거리	컴퓨터에 의해 선별된 문장	정답 여부
1	6.68893	나는 너와 다르다.	O
	8.74927	나는 너와 다른 점이 많다.	O
	9.55747	나는 열심히 공부한다.	X
	9.98589	그 물건을 저에게 전달해 주세요.	X
	10.1409	나는 밥을 먹습니다.	X
2	5.88126	내 친구는 종아리가 가늘다.	O
	6.77754	구름을 이루는 층이 얇다.	O
	8.67891	눈이 와서 길이 미끄럽다.	X
	8.75788	그는 단숨에 잔을 비웠다.	X
	8.87829	내 다리는 가늘다.	O
3	8.37471	재활용품 분리 배출을 잘하자.	O
	8.62781	쓰레기를 버릴 때에는 재활용품을 잘 분리해야 합니다.	O
	8.93753	급브레이크를 밟다.	X
	9.02306	쓰레기를 버릴 때 재활용품을 잘 분류하여 배출해야 합니다.	O
	9.30467	청소부 아저씨들이 재활용품을 분류하여 수거하고 있습니다.	O
4	5.04752	눈이 와서 길이 미끄럽다.	O
	8.28984	구름을 이루는 층이 얇다.	X
	8.75507	눈길이 미끄러워 엉덩방아를 찧었다.	O
	9.19889	미끄러운 눈길을 걸었다.	O
	9.35613	눈이 오는 것은 좋지만, 길이 미끄러운 것은 싫다.	O
5	5.97317	나는 친구에게 선물을 주었다.	O
	7.17447	어머니는 나에게 선물을 주었다.	O
	7.30496	아버지에게 선물을 주었다.	O
	8.14467	나는 어머니와 아버지에게 선물을 주었다.	O
	9.52593	나는 열심히 공부한다.	X
6	4.20527	나는 밥을 먹습니다.	O
	6.84768	많은 사람들이 밥을 먹습니다.	O
	6.93418	나는 밥을 먹고 있습니다.	O
	7.4799	나는 식사를 하고 있습니다.	O
	9.4945	나는 열심히 공부한다.	X
7	6.96464	그 물건을 저에게 전달해 주세요.	O
	8.82896	그 물건 좀 줄래?	O
	9.17212	이 지도에 표시해 주세요.	X
	9.26534	그 물건을 저에게 주시겠습니까?	O
	9.37293	그 물건을 저에게 주시겠어요?	O
8	3.46259	속도위반 단속 취재하던 기자, 차에 다쳐	O
	7.00531	속도위반 단속을 취재하던 기자의 차에 다쳤다.	O
	7.97137	속도위반 단속을 취재하던 기자가 차에 다쳤다.	O
	8.37054	그는 속도위반 단속을 취재하던, 기자 차에 다쳤다.	O
	8.66379	속도위반 단속을 취재하던 기자는 차에 부딪혀 다쳤다.	O

에서는 거리를 계산할 문장의 개수가 80개 밖에 되지 않으므로 시그니처 벡터를 사용하지 않는다.

4.2. 정확도 측정 및 분석

실험 결과, 모든 질의에 대한 Precision@1은 100%로 나타났다. 그러나 k의 값이 증가함에 따라 정확도는 대체적으로 서서히 감소했으며, 모든 질의에 대한 Precision@5는 75%로 측정되었다. 자세한 정확도 분석을 위해 각 질의에 대한 Precision@k를 (Fig. 2)에 나타내었다. 알고리즘에 의해 자동으로 찾은 문장들은 <Table 2>에 자세히 기재하였는데, 예를 들어 '나는 선물을 주었다'라는 질의에 대해서는 '나는 친구에게 선물을 주었다', '어머니는 나에게 선물을 주었다', '아버지에

게 선물을 주었다' 등 유사 문장들을 찾아냄을 확인할 수 있다.

실험 결과에 따르면 다음 3가지 이유로 교육적 효과를 기대할 수 있다.

- 첫째, 문법/의미 둘 중 하나라도 크게 비슷하면 유사 문장으로 선정되었다. 예를 들어 '나는 너와 틀리다'와 '나는 너와 다르다'는 문법/의미적으로 유사하여 유사 문장으로 선정되었고, '쓰레기 분리수거를 잘 해야지.'와 재활용품 분리 배출을 잘하자.'는 의미적으로 유사하기 때문에 유사 문장으로 선정되었다.
- 둘째, 어떤 종류의 변형된 문장이라도 유사 문장을 고르게 잘 찾았다. 테스트로 사용된 8개 질의에 대해 모두 precision@1 결과가 100%로 나타난 것은 매우 고무적이며, 잘못된 어휘 선택을 한 질의 1, 2, 3번의 경우 precision@5가 평균 60%, 4, 5, 6, 7, 8번의 경우 precision@5가 모두 80%가 넘은 것은 어떤 종류의 질의에도 고른 성능을 나타내는 방법임을 나타낸다고 볼 수 있다.
- 셋째, 많은 수의 데이터에 대해서도 추상화된 문장 정보를 잘 만들 수 있음을 확인했다. 테스트에 활용된 문장의 개수는 100개가 되지 않지만, 실제 학습에 사용된 데이터는 200만 문장이 넘었다. 즉, 200만 문장 이상의 많은 문장들을 효과적으로 고차원 벡터로 표현할 수 있는 방법임을 검증했다는 점에서, 실용화 가능성이 매우 높다고 결론내릴 수 있다.

5. 결론 및 향후 연구

본 논문에서는 유사 문장을 자동으로 선별하는 방법을 제안했다. 이 방법은 단어의 구성 성분을 쪼개는 단계, 쪼갠 단어를 입력으로 활용하여 인코더-디코더 모델을 학습하는 단계, 모델을 통해 얻어낸 추상화된 문장을 활용해 검색하는 단계로 구성된다. 실험 결과, 작은 규모의 데이터에 대해 precision@1이 100%, precision@5가 75%의 정확도를 보임으로써 실용화 가능성이 매우 높은 것으로 나타났다.

향후 연구의 방향은 크게 두 가지다. 첫째, 더 많은 데이터를 학습에 활용함으로써 더 높은 성능을 확보할 필요가 있다. 특히, 초등학생이 많이 사용하는 유형의 문장을 집중적으로 학습할 필요가 있다. 둘째, 유사 문장을 추천하는 것이 교육적 효과가 있는 것은 분명하나 본 알고리즘을 실제 교육 현장에 어떤 방식으로 활용할지에 대해서는 더 많은 고민이 필요하다. 예를 들어 몇 개의 유사 문장을 보여줄 것인지, 어떤 형식의 앱 또는 프로그램을 제공할 것인지, 어떤 유형의 글쓰기를 하도록 교육할 것인지에 대한 구체적 사항이 결정되어야 할 것이다.

참고문헌

[1] Bahdanau, D., Cho, K. & Bengio, Y. (2015). Neural Machine Translation By Jointly Learning to Align and Translate. Proceedings of the International Conference on Learning Representations, 1-15.

[2] Kim, S. & Lee, H. (2012). The Effects of Online Bulletin Board on Korean Primary School Students' English Writing and Learning Attitudes. *Primary English Education, 18*(1), 131-150.

[3] Manning, C., Raghavan, P. & Schutze, H. (2008). Introduction to Information Retrieval. Cambridge University Press.

[4] Lee, C. (2010). A Study on Teaching Contents of Sentence Writing in Elementary School. *Grammar Education, 12*(1), 321-341.

[5] Papineni, K., Roukos, S., Ward, T. & Zhu, W. (2002). BLEU: a method for Automatic Evaluation of Machine Translation. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. 311-318.

[6] Park, Y., Park, S., Lee, S. & Jung, W. (2014). Greedy Filtering: A Scalable Algorithm for K-Nearest Neighbor Graph Construction. Proceedings of the 19th International Conference on Database Systems for Advanced Applications, 8421, 327-341.

[7] Park, Y., Park, S., Lee, S. & Jung, W. (2013). Scalable

k-Nearest Neighbor Graph Construction Based on Greedy Filtering. Proceedings of the 22nd International World Wide Web Conference, 227-228.

[8] Park, Y., Hwang, H. & Lee, S. (2016). A Novel Algorithm for Scalable k-Nearest Neighbour Graph Construction. *Journal of Information Science, 42*(2), 274-288.

[9] Park, Y., Hwang, H. & Lee, S. (2015). A Fast k-Nearest Neighbor Search Using Query-Specific Signature Selection. Proceedings of the 24th ACM International Conference on Information and Knowledge Management, 1883-1886.

[10] Park, Y., Hwang, H. & Lee, S. (2016). Query-Specific Signature Seletion for Efficient k-Nearest Neighbour Approximation, doi: 10.1177/0165551516644176.

[11] Pham, T. (2012). A Study on Teaching and Learning Korean Grammars Method based on Paraphrasing Activities. Master's Thesis, Seoul National University.

[12] Sennrich, R., Haddow, B. & Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. Proceedings of the Annual Meeting of the Association for Computational Linguistics, arXiv:1508.07909.

[13] Son, J. (2009). The Study of Scientifically Gifted Students' Scientific Thinking and Creative Problem Solving Ability through Science Writing. *Korean Science Education Society for the Gifted, 1*(3), 21-32.

[14] Salton, G. & Buckley, C. (1988). Term-Weighting Approachs in Automatic Text Retrieval. *Information Processing & Management, 24*(5), 513-523.

[15] Thornbury, S. (2000). How to Teach Grammar, Longman.

저자소개

박 영 기

2008 KAIST 전산학전공(학사)

2010 서울대학교 컴퓨터공학부 (석사)

2015 서울대학교 컴퓨터공학부 (박사)

2015~현재 삼성전자 전문연구원

관심분야: 컴퓨터 교육, 기계 학습, 데이터 마이닝, 소프트웨어 공학

e-mail: ypark@europa.snu.ac.kr