

그래프와 트리 구조를 활용한 한국어 문장제 수학 문제 풀이의 성능개선 연구

(Solving Korean Math Word Problems Using the Graph and Tree Structure)

요약 선행 연구에서 영어 문장제 수학 문제를 해결하려 한 다양한 시도가 있었다. 많은 연구에서 Sequence-to-Sequence(이하 Seq2seq) 패러다임을 넘어선 트리, 그래프 등의 구조를 도입하여 개선된 성능을 달성할 수 있었다. 하지만 한국어 문장제 수학 문제 풀이 연구에서는 트리(Tree)나 그래프(Graph)등으로 제안된 구조를 활용한 모델 사례가 없다. 이에 본 논문에서는 한국어 사전학습 언어모델을 사용하여 트리 구조를 활용하는 모델, 트리와 그래프 구조를 함께 활용하는 모델에 대한 한국어 문장제 수학 문제 풀이 능력의 가능성을 검토해보고자 한다. 테스트 결과 그래프와 트리 구조를 도입함으로써 Seq2seq 구조의 모델 대비 약 20%의 정확도 향상을 보였고, 나아가 한국어 사전학습 언어모델을 사용한 것이 사용하지 않은 것 대비 4.66 ~ 5.96%의 정확도 향상을 보였다.

키워드: 문장제 수학 문제 풀이, 사전학습 언어모델, 그래프 신경망, 트리 구조 수식 생성

Abstract In previous studies, there have been various efforts to solve math word problems in the English sentence system. In many studies, improved performance was achieved by introducing structures such as trees and graphs, beyond the Sequence-to-Sequence approaches. However, in the study of solving math problems in Korean sentence systems, there are no model cases, using structures such as trees or graphs. Thus, in this paper, we examine the possibility of solving math problems in Korean sentence systems for models using the tree structure, graph structure, and Korean pre-training language models together. Our experimental results showed that accuracy improved by approximately 20%, compared to the model of the Seq2seq structure, by introducing the graph and tree structure. Additionally, the use of the Korean pre-training language model showed an accuracy improvement of 4.66%–5.96%.

Keywords: solving math word problems, pre-trained language models, graph neural network, generating tree structure equation

1. 서론

문장제 수학 문제 풀이란 여러 상황에서 발생할 수 있는 상황이나 수학적 관계가 언어와 숫자 표현으로 구성된 문제로 주어질 때, 자연어 처리 기술을 이용해 문제를 언어적, 수적, 논리적 능력으로 분석하고 개체 간의 관계를 이해하여 문제를 해결하는 방법을 말한다. 영어권 선행 연구에서는 문장제 수학 문제를 풀기 위해 Seq2seq 방법론을 적용하였고 가능성을 확인하였다[1]. 하지만 Seq2seq 구조의 모델은 문장제 수학 문제에서 길이, 개수, 넓이, 단위 등으로 나타나는 개체의 특성을 주어진 수식이나 답으로 유추하기 어렵다는 한계를 보였고, 이를 해결하기 위해 Alignment[2], Tree[3,4],

Graph[5], Common sense[6] 활용 등의 다양한 정보로의 재구축을 통해 문제를 해결하고자 하는 시도가 있었다. 이러한 선행 연구들은 그래프를 활용한 학습법이 문장제 수학 문제에서 나타난 개체들 간의 관계를 이해하는 데에 효과적이라는 것과, 트리구조의 수식 표현이 그 정보를 연결하는 데에 매우 적합하다는 것을 확인시켜 주었다.

국내에서는 한국어 문장제 수학 문제를 해결하는 것을 목표로, 영어권의 선행 연구 결과를 활용한 연구들이 진행되었다[7,8,9]. 이들은 주로 선행 연구에서 사용되는 영어 데이터를 한국어로 번역하여 기계 학습 모델을 사용해 문제 풀이를 시도하였다. 또한 한국어 사전학습 언어모델을 활용한 모델이

수학 문제 풀이에 도움이 된다는 것을 알 수 있었다[8,9]. 하지만 한국어권에서는 영어권 선행 연구에서 시도한 것과 같이 다양한 정보를 활용하여 문제를 풀고자 한 실험은 많이 없었다. 특히 영어권 선행 연구의 높은 정확도를 통해 재귀적인 트리 구조 수식 생성이 Seq2seq 같은 구조로 순차 데이터로 수식을 생성하는 것보다 더 알맞은 방법론임을 알 수 있었지만[3,4] 한국어권에서 트리 구조를 도입하여 수학 문제를 풀고자 한 시도된 사례가 없었다. 이에 본 논문에서는 MAWPS(A Math Word Problem Repository)[10] 데이터셋을 한국어에 맞게 가공을 한 뒤 그래프 및 트리를 활용한 모델과 Seq2seq 기반의 모델 간의 학습 성능 차이를 확인함으로써 그래프 및 트리 구조의 학습법이 한국어 문장제 수학 문제에서도 효과적인지 알아보고자 한다. 나아가 한국어 사전학습 언어모델인 KoBert, KoElectra와 그래프 및 트리 구조를 활용하는 모델을 함께 사용할 때, 그래프 및 트리 구조만을 활용할 시에 부족할 수 있는 언어적 특성에 대해서 사전학습 언어 모델이 이를 보완할 수 있는지 학습 결과를 비교함으로써 논의하고자 한다.

2. 관련 연구

2.1 딥러닝을 사용한 수학 문제 풀이 시도

[1]에서는 파이(π)=3.1415...와 같은 수학적 지식을 활용한 데이터를 구축하고 이를 Seq2seq 기반의 OpenNMT[11]로 평가하였다. 자체적으로 구축한 데이터셋에 대한 정확도가 54.2% 정도의 성능을 보였다. 이후 [3]은 수학 문제 풀이의 식이 트리 구조의 형태를 갖는다는 점에 주목하며, Seq2seq 모델을 기반으로 하여 트리 형태의 수식을 예측하게 하는 시도를 하였다. 하지만 RNN(Recurrent Neural Network) 기반의 디코더(Decoder)가 트리의 재귀적인 특성을 학습할 수 없었고, 이를 보완하기 위해 재귀적 인공 신경망이 도입되었다[4]. [4]은 [3]과 같은 트리 형태의 식을 학습하고 예측한다는 점은 동일하지만, 재귀적 인공 신경망을 사용하여 트리 노드(Node)를 생성할 수 있다는 점이 다르다. 이는 수학 문제와 같이 트리 구조의 데이터가 갖는 노드 간의 종속적인 특징을 학습이 가능하게 하여 Math23K[12] 데이터에 대해 74.3%의 정확도를 달성할 수 있었다. [5]는 [4]에서 제안된 트리 디코더를 기반으로 문제에서 나타나는 개체들 간의 관계를 직접적으로 그래프의 형태로 풀어지게끔 학습을 유도하는 방식을 사용하였고 Math23K에서 75.5%의 정확도를 달성할 수 있게 되었다. 위 논문에서는 그래프 인공 신경망의 사용이 개체가 갖는 수적 정보를 풍부하게 함으로써 성능을 개선할 수 있었다.

자연어 처리(Natural Language Processing, NLP)에서의 대규모 사전 학습 언어 모델이 다양한 분야에서 성능 개선을 기록하였다[13, 14]. 사전

학습 언어 모델이 다양한 도메인에서 성공을 거두면서 자연스럽게 문장제 수학 문제 풀이에서도 적용하고자 하는 시도가 있었다. [15]는 질의응답(Question Answering) 방법에서 언어 모델이 문제를 해석하고 도출하는 임베딩(Embedding)을 정답의 위치, 수 세기, 부정 등 여러가지 정답 타입에 맞게 새로운 순방향 신경망(Feed-Forward Network)들을 구성하여 학습하고자 하였다. 특히 추가된 신경망들이 수리적 추론 능력을 갖추면서, 질의응답 분야에서의 성능이 향상되었다. 그리고 언어 모델의 파라미터를 디코더에서도 활용함과 동시에 사칙연산의 문제와 답을 학습시키는 것으로 수적 추론 능력을 직접 주입하고자 했던 실험이 있었다[16]. 사칙연산의 학습으로 인해 덧셈, 뺄셈과 같은 연산이 가능하면서도 DROP[17]과 같은 일반 질의응답에서 준수한 성능을 기록할 수 있다는 것을 확인할 수 있었다. 두 실험 모두 언어 모델을 다양하게 활용하여 수학적 추론에서 효과적이라는 점을 시사하고 있지만, 언어 모델이 질의응답의 성능을 유지하면서도 수적 추론 능력을 갖도록 하는 방법론이기에 여러 번의 사칙연산을 포함한 완전한 문장제 수학 문제 풀이에는 걸맞지 않다.

본 논문에서는 [4]에서 소개하는 트리 디코더를 기반으로, [5]에서 소개한 그래프 인공 신경망을 적용하여 한국어 문장제 수학 문제를 해결하고자 한다. 또한 사전학습 언어모델이 언어적 자질을 추출할 수 있는 것에 반해, 부족한 수리적 추론 능력에 대해서 그래프 네트워크의 활용이 문제 속 개체들의 관계와 수리적 추론 능력에 확인하고자 하였으며, 결과적으로 그래프와 트리 구조를 사용한 한국어 수학 문제 풀이 방법에 대한 가능성과 언어 모델이 추가되었을 때 더욱 성능이 개선될 수 있는지에 검증하였다.

2.2 한국어 문장제 수학 문제 풀이

국내에서의 선행 연구들은 공개된 영어권 문장제 수학 문제들을 한국어로 번역하여 사용하였다. 이는 한국어 도메인에서도 비슷한 유형의 문제들이 제안된 방법으로 풀이 가능하다는 점을 보여주었다. [7]은 영어권 선행연구에서 사용된 템플릿기반 Log-linear 분류 모델이 한국어로 번역된 데이터셋에서 유효하다는 것을 확인하였다. 비슷한 연구인 [8]에서는 템플릿을 사용한 분류 모델 방법 기반인 것은 동일하나, 문제의 자질 추출 역할을 BERT[13]를 도입함으로써 해결하였다. 그리고 [7]의 후속 연구에서 제안된 koEPT 모델[9]에서는 기존 템플릿기반 분류모델에서 결정경계가 모호한 경우를 지적하며 Expression Pointer Transformer(EPT)[18] 구조를 채택한다. 연산자와 피연산자를 한 단위로 나누어 묶어 사용되리라고 예상되는 묶음을 제공하고, 문맥적 정보를 반영한 피연산자-문맥 포인터를 통해 후보를 선택하게 되어 식을 생성한다. 위와 같은 선행 연구들을 통해

영어권 수학 데이터를 번역하여 사용하는 것과 Seq2seq 구조의 모델이 문제 풀이에 있어 가용한 것을 확인하였으나, 트리나 그래프를 사용하는 것과 같이 문제의 수식 구조, 수에 대한 이해와 같은 문맥 정보를 목적에 맞게 재구성하여 학습하는 모델의 연구는 진행되지 않았다. 본 논문에서 진행되는 실험은 [9]에서 사용한 사전학습 언어모델을 임베딩 단계에 적용하는 것으로 문제에서 포함하고 있는 여러가지 언어적 자질을 학습할 수 있다. 또한 [7,8]에서 문제에 대한 유형을 템플릿으로 분류한 것과는 달리 트리 구조의 수식 생성 모듈을 사용하여 학습된 언어 자질이 각 트리 수식 토큰으로 생성될 수 있어, 더욱 효과적으로 정답을 유추할 수 있다.

3. 모델

본 절 3.1에서는 한국어 문장제 수학 문제 풀이의 정의 및 이론을 설명한다. 이어서 한국어 사전 학습 언어 모델을 활용하여 생성된 임베딩이 그래프 구조의 정보로 변환되는 과정과 트리 구조의 디코딩이 어떻게 진행되는 지를 기술한다. [5]에서 소개하는 그래프 모델의 구조에 3.2의 언어 모델을 거친 임베딩이 변환되는 것을 3.3에서 수식으로 표현하였다. 3.4의 트리구조의 디코더는 [4]에서 소개한 것과 같은 메커니즘을 사용하기 때문에 방법을 간단하게 서술하는 것으로 대체한다.

3.1 문제 정의

표 1 한국어 문장제 수학 문제 풀이 정의

Table 1 Definition of Solving Korean Math Word Problems

Type	Example
Problem	학교에서 비안카가 14 개의 컵케이크를 만들었다. 그녀가 6 개를 팔아서 17 개를 더 만들었다면, 그녀는 몇 개의 컵케이크를 가지고 있을까?
Answer	25
Equation	$14 - 6 + 17$
Converted Equation	$\text{number0}(14) - \text{number1}(6) + \text{number2}(17)$

한국어 문장제 수학 문제 풀이는 <표 1>의 예시와 같이 한국어로 주어진 수학 문제 Problem에 대해 정답 Answer를 유추하는 것을 목표로 한다. 자연어 처리에서의 질의 응답 모델을 도입하여 문제를 질의로, 문제에 대한 정답을 응답으로써 학습하는 것으로는 수, 개체, 연산식에 대한 이해를 이끌어 내기 어렵기 때문에 문제를 해결할 수 있는 수식 Equation을 맞추는 방법으로 발전하였다. 또한 모든 수를 토큰으로 학습하는 것이 학습에 있어 비효율적이었기 때문에, <표 1>의 Converted Equation과 같이 수 토큰을 따로 지정하여 수식에 나타낸다. 문제에서 등장하는 순서에 맞게 수 토큰

의 순번을 지정하여 학습되도록 한다.

3.2 사전 학습 언어 모델 임베딩

한국어 문장제 수학 문제 P 는, 문장 토큰 $V_p = \{v_1, \dots, v_m\}$ 과 수 토큰 $n_p = \{n_1, \dots, n_l\}$ 의 연속으로 이루어져 있다(l 과 m 은 각각 문제에 나타난 수의 개수 및 단어의 개수). 이를 사전 학습 언어 모델 M_{pre} 의 입력으로 사용하면, 언어 모델에 의해 수학 문제의 언어적 특성이 분석된 임베딩인 $H_{emb} \in R^{N \times d}$ 을 얻을 수 있다 (여기서 N 은 $m+l$, d 는 hidden vector의 차원 수를 나타냄).

$$H_{emb} = M_{pre}(P) \quad (1)$$

3.3 그래프로의 표현

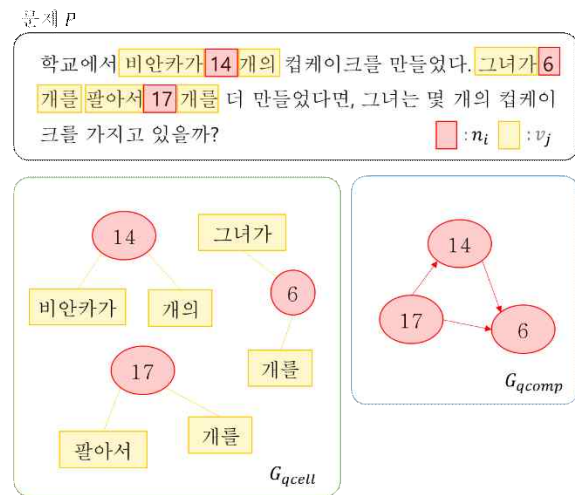


그림 1 문제로부터의 그래프 구축 예시

Fig. 1 Construction example of graph from problem

문제를 그래프로 표현하기 위해, 문제에 나타난 모든 수 토큰 n_p 와 단어 토큰 V_p 를 그래프의 노드로 참조한다. 그 다음 문제 P 를 $QC = \{Q_1, Q_2, \dots, Q_l\}$ 인 수량 셀로 변환한다. 각 수량 셀 $Q_i \in QC$ 는 수 토큰 $\{n_i\}$ 와 그 수 토큰에 관련된 속성 값 $\{v_{1i}, v_{2i}\}$ 을 포함한다. 속성 값으로 수 토큰 위치의 전후 위치에 있는 단어 토큰을 사용한다.

그림 1과 같이 수량 셀들을 통해 수량 셀 그래프 G_{qcell} 과 수량 비교 그래프 G_{qcomp} 를 구축할 수 있다. 각 수량 셀 $Q_i = \{n_i\} \cup \{v_{1i}, v_{2i}\}$ 에 대하여 n_i 과 $v_j \in \{v_{1i}, \dots, v_{qi}\}$ 사이의 방향이 지정되지 않은 간선 e_{ij} 을 G_{qcell} 에 추가한다. 이 때, v_j 는 문제에서 나타난 모든 수 토큰 전후 위치의 단어 토큰들의 집합이다. 한편 두 수 토큰 $n_i, n_j \in n_p$ 에 대해 $n_i > n_j$ 인, 방향이 있는 간선 $e_{ij} = (n_i, n_j)$ 를 G_{qcomp} 에 추가한다. 그리고 각 그래프 G 의 인접행렬 $A \in R^{N \times N}$ 을 각 노드에 맞

는 간선이 존재할 경우에 1로, 아닐 경우 0으로 초기화해준다. 이로써 G_{qcomp} 의 인접행렬 A_{qcomp} 와 G_{qcell} 의 인접행렬 A_{qcell} 을 구할 수 있다.

생성된 인접 행렬 $\{A_k\}_{k=1}^K$, $A_k \in \{A_{qcomp}, A_{qcell}\}$ 과 초기 노드 임베딩 H_{emb} 를 그래프 변환 모듈의 입력으로 사용한다. 여기서 K 는 그래프의 개수이며 $A_k \in \mathbb{R}^{N \times N}$ 은 k 번째 인접행렬이다. 그래프 노드의 특징을 학습하기 위해 GCNs(Graph Convolutional Networks) (Kipf and Welling, 2017)을 사용하며, 여러 그래프를 위해 K -head 그래프 합성 곱을 이용한다. 단일 GCN은 파라미터 $W_{gk} \in \mathbb{R}^{d \times d_k}$ 를 갖는다($d_k = d/K$). 그래프 구조를 나타내는 인접 행렬 A_k 과 X (처음에는 H_{emb} 를 사용한다)를 사용하여 GCN 학습을 다음과 같이 정의한다.

$$GCN(A_k, X) = GCConv_2(A_k, GCConv_1(A_k, X)) \quad (2)$$

GCN은 2개의 그래프 합성 곱 연산을 포함한다.

$$GCConv(A_k, X) = \text{relu}(A_k X^T W_{gk}) \quad (3)$$

$\{A_k\}_{k=1}^K$ 의 각 그래프를 GCN을 사용해 병렬로 학습시켜 d_k -차원의 출력 값을 산출한다. 이를 연결하여 투영시키면, 최종 값은 다음과 같다.

$$Z = \text{concatenate}(GCN(A_k, H_{emb}))_{k=1}^K \quad (4)$$

이 결과를 순방향 신경망, 층간 정규화 레이어, 잔차 연결(residual connection)을 통해 증축한다.

$$\hat{Z} = Z + \text{LayerNorm}(Z) \quad (5)$$

$$\bar{Z} = \hat{Z} + \text{LayerNorm}(FFN(\hat{Z})) \quad (6)$$

여기서, $FFN(x)$ 는 relu 함수를 사이에서 사용하는 순방향 신경망 두 층으로 구성되어 있다.

$$FFN(x) = \max(0, x W_{f1} + b_{f1}) W_{f2} + b_{f2} \quad (7)$$

이로써 노드 표현 \bar{Z} 는 수 토큰, 개체들과 관계들을 나타낸다. 마지막으로 전역적 그래프 간 문맥을 학습하기 위해 원소 기준 최소 풀링(Element-wise min-pooling) 연산을 학습된 모든 노드 표현들에 적용시켜준다. 그리고 완전 연결 계층을 통해 그래프 표현 z_g 를 도출하게 된다.

$$z_g = FC(\text{MinPool}(\bar{Z})) \quad (8)$$

3.4 트리 구조의 디코더

트리 디코더의 목적은 전위 표기된 트리 구조 q_t 의 연산식을 학습하고 예측하는 것이다. 문제 P 의 타

겟이 되는 연산 수식 어휘들을 $V_{dec} = V_{op} \cup V_{con} \cup n_P$ 라고 할 때, 특정 토큰 임베딩 $e(y|P)$ 는 다음과 같이 정의된다(여기서 V_{op} 는 연산자, V_{con} 은 $\pi, 1$ 과 같은 상수를 나타낸다).

$$e(y|P) = \begin{cases} e_{(y, op)} & \text{if } y \in V_{op} \\ e_{(y, con)} & \text{if } y \in V_{con} \\ \bar{z}_{loc(y, P)}^p & \text{if } y \in n_P \end{cases} \quad (9)$$

연산자는 단말 노드가 될 수 없으며 n_P 는 항상 단말 노드에 위치하게 된다. \bar{Z} 로부터 P 에서 n_P 의 위치 y 에 맞는 $\bar{z}_{loc(y, P)}^p$ 가 수 토큰 위치의 임베딩으로 사용된다. 전위 순회 과정으로 트리의 노드를 생성한다. 초기 q_{root} 로 z_g 를 사용한다. 예를 들어 부모 노드로부터의 왼쪽 노드는 그림 2의 ①②와 같이 부모 노드의 벡터 q_y 와 그 예측된 토큰 임베딩 $e(\hat{y}|P)$ 를 사용하여 생성된다. 오른쪽 노드를 생성하는 경우는 그의 쌍둥이 왼쪽 노드가 완성된 후라고 할 수 있다. 오른쪽 노드는 그림 2의 ④⑦에서 나타난 것과 같이 부모 노드와 쌍둥이 왼쪽 노드의 정보를 모두 사용하여 생성하게 된다. 단말 노드가 아닌 트리의 정보를 압축(encode)하기 위해, 그림 2의 ⑥과 같이 재귀 신경망이 정의되어 연산자 토큰의 임베딩과 왼쪽, 오른쪽 임베딩을 융합(fuse)할 수 있다. 단말 노드는 그 토큰의 임베딩을 나타낸 것을 트리 임베딩으로 사용한다(그림 2의 ③⑤).

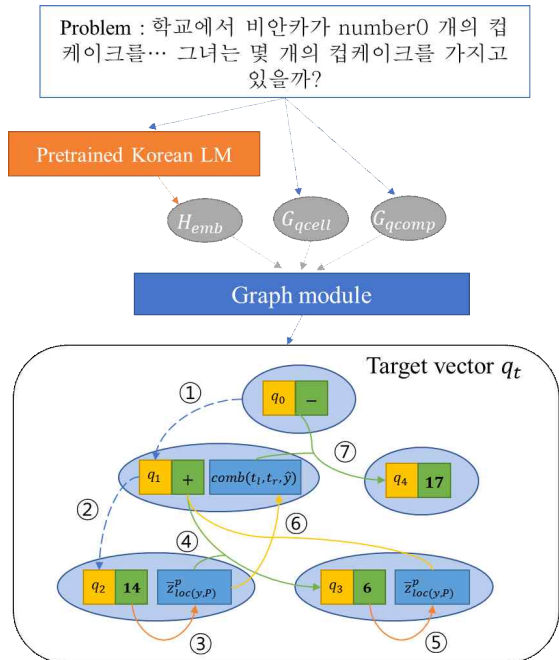


그림 2 graph module을 포함한 트리 구조 디코더의 수식 예측 과정

Fig. 2 The process of predicting equation of tree structure decoder including graph module

표 2 MAWPS 번역본 예시

Table 2 An example of MAWPS translation

Language	Korean	English
Question	학교에서 비안카가 number0 개의 컵케이크를 만들었다. 그녀가 number1 개를 팔아서 number2 개를 더 만들었다면, 그녀는 몇 개의 컵케이크를 가지고 있을까?	For the school bake sale Bianca made number0 cupcakes. If she sold number1 of them and then made number2 more, how many cupcakes would she have?
Numbers	14, 6, 17	14, 6, 17
Equation	- + number0 number2 number1	- + number0 number2 number1
Group_nums	[1,2,3,6,7,8,9,10,11]	[6, 7, 8, 12, 13, 14, 18, 19, 20]

위 과정을 통해 구할 수 있는 목표 벡터 q_t 와, G_c 를 \bar{Z} 와, q_t 간의 주의(Attention)로 계산된 전역 그래프 벡터라고 할 때, 문제-트리 연산식 구조 (p, T) 의 손실 함수 $L(T, P)$ 는 t -노드 토큰 y_t 의 음의 로그 가능도 확률의 합(sum of the negative log-likelihoods of probabilities)으로 정의된다 (여기서 E 는 T 의 토큰 개수이고 $prob$ 은 GTS(Goal-driven tree-structured neural model) [4] 논문에서 사용된 확률 함수를 지칭한다).

$$L(T, P) = \sum_{t=1}^E -\log prob(y_t | q_t, G_c, P) \quad (10)$$

4. 실험 및 분석

4.1 데이터셋

MAWPS[10]은 워싱턴 대학교에서 제안한 문제 데이터셋으로 Illinois[19], CommonCore[20]보다 더욱 다양한 형태의 방정식을 제공하고 있으며 총 1843개의 일차 방정식 문제로 구성되어 있다. 각 문제는 문제 내용과 매핑된 숫자 값, 정답, 방정식, 답, 수의 위치 정보 (group_nums), 설명문, 질문으로 이루어져 있다. 문제의 내용은 대체적으로 초등학교 고학년 수준에 해당하는 일차 방정식 내용으로 이루어져 있다. 본 논문에서는 MAWPS를 번역하여 제안하는 모델에 대한 실험을 진행한다. 번역된 MAWPS의 예시는 <표 2>와 같다. 기존 MAWPS에서는 제공하지만 본 논문에서 사용하지 않는 정답, 답, 설명문, 질문은 제외하였다.

표 3 일부 양식 수정 예시

Table 3 Modification examples for some patterns

Type	Modification example
Proper noun	스키틀즈 -> skittles 상호명 -> 레스토랑, 회사 등... 원, 엔 -> 달러
Measure	킬로미터, 센티미터 -> km, cm 다스 -> 12개
Situation	수확하다 -> 따다 돈을 맡기다 -> 예금하다

번역 작업은 1차로 구글에서 지원하는 번역기를 사용하여 MAWPS 데이터를 전체적으로 변환하는 과정을 거쳤다. 2차 번역은 대학생 이상의 학력을 갖춘 2명의 인원을 모집하여 번역오류와 어색한 문장을 교정하는 수작업을 수행하는 것으로 완료하였다. 데이터셋을 전체 인원으로 나누어 할당하여 진행한 후, 1달의 기간을 걸쳐 모든 인원이 전체 데이터를 5차례에 걸쳐 상호 검증하여 번역의 완성도를 높였다. 구글번역기로 익숙하지 않은 언어와 영어로 된 일부 양식은 <표 3>과 같이 동일하지만 익숙한 단어로 수정을 진행하였다.

4.2 실험

학습은 30 에폭, 배치사이즈는 8로 학습을 수행하였고 3겹 교차검증(3-fold cross validation)을 실시한다. MAWPS의 데이터 수는 1843개로 많지 않기 때문에, 교차검증으로 모든 데이터를 평가에 사용할 수 있게 하여 정확도를 향상시키고자 하였다. 영어 데이터에 대한 실험을 모든 모델에 함께 수행하여서 언어 도메인의 차이에 따른 결과 비교가 가능하도록 한다. 영어 데이터에 대한 사전학습 언어모델은 공개되어 있는 BERT[13], Electra[14]를 사용하였다. 마찬가지로 한국 데이터에 대한 사전학습 언어모델도 공개되어 있는 KoBert[21]와 KoElectra[22]를 사용하였다. 그리고 한국 데이터에 한해 트리 구조의 연산식을 Seq2seq 기법으로 학습하였을 때와의 차이를 나타내기 위하여 OpenNMT[11]를 사용해 학습하고 평가한 결과를 나타내고 있다.

실험에서는 3.3의 트리 구조 디코더를 공통적으로

포함하는 모델을 기반으로 하여, 3.2에서 설명한 그래프로의 변환 모듈의 유무에 따라 TO(Tree only, 그래프 모듈 미포함), TWG(Tree with graph, 그래프 모듈 포함)로 구분한다. 위 두가지 모델에서 사전 학습 언어 모델의 차이 및 유무에 따른 학습 결과를 보이고자 한다. 위 실험 구성에서는 그래프 모듈을 단독으로 한 세팅을 실험하지 않는다. 그래프를 사용함으로써 얻을 수 있는 개체 간의 정보가 순차적인 RNN의 디코더보다 상호관계가 명확한 트리 디코더에서 더욱 유용할 것으로 판단하였기 때문이다.

그림 3과 그림 4는 본 논문에서 실험한 모든 한국어 모델의 학습 정확도 그래프를 나타낸다. 결과는 Seq2seq 모델이 5 epoch 내로 60%로 수렴하였고, 언어 모델을 사용한 TO에서 15 epoch 때 81% 정확도로, TWG에서 6 epoch 때 85%의 정확도로 수렴하였다.

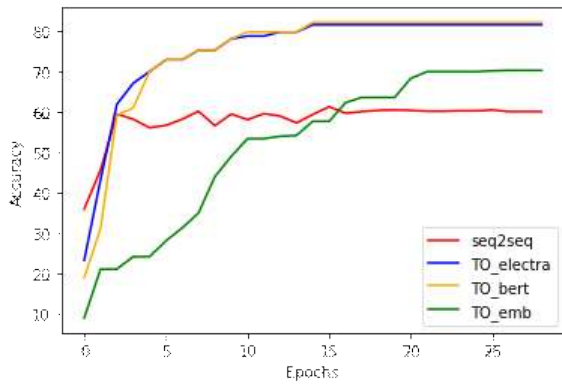


그림 3 TO 모델의 학습 정확도

Fig. 3 Training accuracy of the TO model

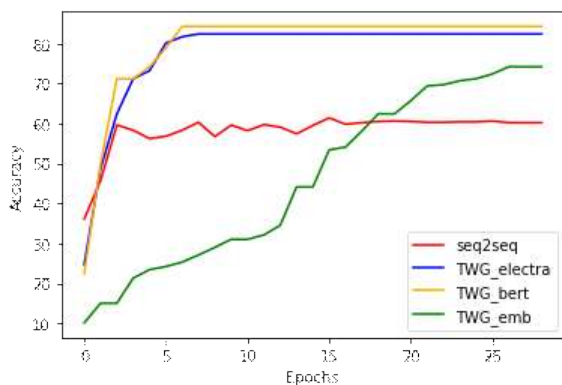


그림 4 TWG 모델의 학습 정확도

Fig. 4 Training accuracy of the TWG model

기본 임베딩을 사용한 TO_emb, TWG_emb 모델들은 전체 epoch에서 꾸준한 학습을 보여주다가 20 epoch 정도에서 75% 내외의 정확도로 수렴하였다. 모든 학습 정확도 그래프 결과에서 90%의

높은 결과가 나오지 않은 것은 데이터가 1843개 정도로 비교적 부족하기 때문이라고 추측된다. 학습 결과, OpenNMT로 실험한 Seq2seq의 학습 정확도가 가장 낮았다. 하지만 TWG와 TO에서 언어 모델이 사용되지 않은 경우 초반 학습이 느린 편이라는 것을 알 수 있는데, 이는 언어 모델을 사용한 다른 모델과는 달리 언어적 특성을 학습하는데 걸리는 시간이 포함되어 있기 때문이라고 추측할 수 있다. 반면 초반 학습이 가파른 Seq2seq는 비교적 단순한 모델 구조를 갖기에 빠르게 최대치까지 학습되는 모습을 확인할 수 있었다.

사전학습 언어모델을 사용하지 않은 기본 모델들의 결과는 <표 4>와 같다. 한국어 수학 문제 풀이의 정확도는 TO, TWG 각각 71.12%, 72.98%의 결과가 나왔고 Seq2seq 기법을 활용한 정확도는 55.48%를 기록하였다. 영어 수학문제 풀이의 정확도는 TO, TWG 각각 81.34%, 83.30%의 결과가 나왔다.

<표 5>는 BERT, KoBert, Electra, KoElectra를 사용한 학습 및 평가의 결과이다. 한국어 수학 문제 풀이의 정확도는 KoBert에서 TO, TWG 각각 76.13%, 78.94%의 결과가 나왔고, 영어 수학문제 풀이의 정확도는 BERT에서 TO, TWG 각각 86.72%, 88.92%의 결과가 나왔다. 또한 KoElectra에서의 한국어 수학 문제 풀이의 정확도는 TO, TWG 각각 75.82%, 77.64%의 결과가 나왔고 Electra에서의 영어 수학문제 풀이의 정확도는 TO, TWG 각각 88.94%, 89.81%의 결과가 나왔다.

표 4 그래프 혹은 트리를 사용한 기본 모델과 일반 Seq2seq 모델의 학습 결과

Table 4 Training results of the base model using the graph or tree and general seq2seq model

Model	Korean	English
Seq2seq	55.48%	68.71%
TO	71.12%	81.34%
TWG	72.98%	83.30%

표 5 사전학습 언어모델을 포함한 학습 결과

Table 5 Training results of including pre-trained language models

Language	LM type	Model	
		TO	TWG
English	BERT	86.72%	88.92%
	Electra	88.94%	89.81%
Korean	KoBert	76.13%	78.94%
	KoElectra	75.82%	77.64%

4.3 분석

표 6 한국어 데이터에서의 각 모델의 예측 결과

Table 6 Prediction results of each model in Korean data

case 1: 어릿광대는 number0 개의 풍선을 가지고 있었다. 그는 파티 후에 number1 개를 남겼다. 그는 파티에서 얼마나 많은 풍선을 사용했는가?
Seq2seq, TWG_emb, TWG_electra: - number0 number1 (correct)
case 2: 트리비아 팀은 총 number0 명의 멤버가 있었지만 number1 명의 경기 동안 모습을 드러내지 않았다. 각 멤버가 number2 점을 기록했다면, 몇 점을 합산했을까?
Seq2seq: - - number0 number1 number2 (wrong) TWG_emb, TWG_electra: * - number0 number1 number2 (correct)
case 3: 에드워드는 국가 박람회에서 number0 장의 티켓을 구입했다. 그는 부스에서 number1 장의 티켓을 사용했고, 나머지 티켓은 기구 승차에 사용하기로 결정했다. 승차권마다 number2 장씩 비용이 든다면, 그가 승차에만 사용하는 비용은 얼마일까?
Seq2seq: - - number0 number1 number2 (wrong) TWG_emb: / - number0 number1 number2 (wrong) TWG_electra: * - number0 number1 number2 (correct)

실험 결과는 트리, 그래프와 사전 학습 언어 모델을 사용하였을 때, 영어 데이터는 물론 한국어 데이터에서도 가장 우수한 성능을 기록하였다는 것을 나타낸다. 이를 자세하게 알아보기 위하여 모델별 예측에 대한 분석을 진행한다. <표 6>은 한국어 데이터에서 모델 별로, 비슷하지만 다른 예측 결과를 갖는 문제 예시들을 보여준다. case1은 연산자가 하나인 수식의 단순한 문제여서 seq2seq, TWG_emb, TWG_electra 모델 모두 정답을 맞추는 반면, case2에서 ‘각 멤버’가 수식에 직접적으로 곱연산이 필요한 등의 각 수 개체 간의 관계가 대소적인 비교 및 연산 순서 등으로 나타나야 하는 경우 seq2seq 모델은 정답을 맞추지 못하였다. 이는 그래프와 트리 구조의 도입이 수학적 관계에서 문제 풀이에 효과적으로 사용이 되었음을 시사하고 있다. 또한 case3은 TWG_electra 만 정답을 맞추었는데, 이는 포함되어 있는 언어모델이 데이터가 부족한 본 실험에서 문장이 긴 문제에 대해 정답을 맞추는 능력을 향상시켰기 때문이라고 유추할 수 있다. 마지막으로 모든 문제에서 보다 좋은 성능 지표 값을 갖는 모델이 틀린 문제를 맞추는 하위 성능 지표 모델은 없는 것으로 나타났다. 결국 그래프와 트리, 그리고 사전 학습 언어 모델을 포함하는 것이 한국어 문장제 수학 문제 풀이에 충분히 활용될 수 있음을 보이고 있으며, 문제에서 나타나는 다양한 정보를 재구성하여 학습하는 것이 한국어에서도 가능하다는 점도 시사한다.

또한, <표 4>와 <표 5>를 비교했을 때, 지표적으로도 사전학습 언어모델을 적용한 실험이 그렇지 않은 경우보다 평균 5%의 성능 개선이 있었다는 것을 확인하였다. 이와 같은 결과로 한국어에서도 트리, 그래프와 같은 구조의 모델이 언어모델을 통해 언어적 특성을 사용할 수 있을 때 성능이 좋아진다는 사실을 알 수 있었다. 특히, 그래프와 트리 구조를 같이 활용하는 모델이 동일함에도 불구하고 사

전학습 언어 모델 임베딩을 사용한 것만으로 성능이 개선되었다는 점에서 추후 실험에서의 활용 가능성이 높다는 것을 알 수 있다.

5. 결론

본 논문에서는 한국어 문장제 수학 문제 풀이를 위해 한국어 문제를 확보하였고, 그래프 및 트리 구조를 활용하는 모델들을 사용하여 한국어 문장제 수학 문제 풀이를 시도하였다. 그 결과 한국어 seq2seq 기반의 모델에 비해 그래프 및 트리 구조를 활용한 모델이 약 20% 정도 정확도가 높은 것을 확인하였다. 또한 단순한 임베딩 모델을 사용하는 것 보다 충분한 양으로 학습된 언어모델을 함께 사용하여 임베딩을 생성할 시 약 5% 정도 높은 정확도를 달성하였다. 이로써 한국어권 문장제 수학 문제 풀이에서도 그래프와 트리 구조를 활용한 모델과 언어 모델을 결합하였을 때, 기존 seq2seq 모델에 비해 좋은 성능을 보인다는 것을 알 수 있었다. 하지만 실험은 단 하나의 데이터셋 MAWPS 만으로 진행했기에 일반화 및 확장성에서 한계를 지닌다. 또한 MAWPS 데이터셋은 단 1843개의 문제 밖에 가지고 있지 않아 성능 측정을 하는데 문제가 있다. 문제의 난이도 또한 초등학교 수준의 문제로만 이루어져 있다는 한계가 있었다. 향후 그래프 및 트리 구조를 활용하면서 문장제 수학 문제가 갖는 개체간 특성을 언어 모델이 분석할 수 있는 연구 등이 진행될 수 있을 것이다.

References

- [1] Amini, A., Gabriel, S., Lin, S., Koncel-Kedziorski, R., Choi, Y., & Hajishirzi, H. “MathQA: Towards Interpretable Math Word Problem Solving with Operation-Based Formalisms,” *North American Chapter of the*

- Association for Computational Linguistics* pp. 2357-2367, 2019
- [2] Qin, J., Lin, L., Liang, X., Zhang, R. and Lin, L., "Semantically-Aligned Universal Tree-Structured Solver for Math Word Problems," *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 3780-3789, 2020.
- [3] L. Wang, Y. Wang, D. Cai, D. Zhang, and X. Liu, "Translating math word problem to expression tree," in *Proc. Conf. Empirical Methods Natural Language Processing*, pp. 1064-1069, 2018.
- [4] Xie, Z. and Sun, S., 2019. "A Goal-Driven Tree-Structured Neural Model for Math Word Problems," *International Joint Conferences on Artificial Intelligence Organization*, pp. 5299-5305, 2019.
- [5] Zhang, J., Wang, L., Lee, R.K.W., Bin, Y., Wang, Y., Shao, J. and Lim, E.P., "Graph-to-Tree Learning for Solving Math Word Problems," *Association for Computational Linguistics*, pp. 3928-3937, 2020.
- [6] Qin, J., Liang, X., Hong, Y., Tang, J. and Lin, L., "Neural-symbolic solver for math word problems with auxiliary tasks," *Association for Computational Linguistics*, pp. 5870-5881, 2021.
- [7] Changhyub Woo, Gahgene Gweon, "Solving Automatically Algebra Math Word Problem in Korean," *The 30th Annual Conference on Human & Cognitive Language Technology*, pp. 310-315, 2018.
- [8] Ki, K.S., Lee, D.G. and Gweon, G., "Kotab: Korean template-based arithmetic solver with bert," In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 279-282, 2020.
- [9] Rhim, S. K., Ki, K. S., Kim, B., & Gweon, G., "KoEPT: Automatically Solving Korean Math Word Problems using Generative Transformer," *Proceedings of the Korea Information Processing Society Conference*, pp. 362-365, 2021.
- [10] Koncel-Kedziorski, R., Roy, S., Amini, A., Kushman, N. and Hajishirzi, H., "MAWPS: A math word problem repository," In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1152-1157, 2016.
- [11] Klein, G., Kim, Y., Deng, Y., Senellart, J. and Rush, A.M., "Opennmt: Open-source toolkit for neural machine translation," In *Proceedings of ACL*, pp. 67-72, 2017.
- [12] Yan Wang, Xiaojiang Liu, and Shuming Shi, "Deep neural solver for math word problems," In *EMNLP*, pages 845-854, 2017.
- [13] Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K., "BERT: pre-training of deep bidirectional transformers for language understanding," In *Proc. 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171-4186, 2019.
- [14] K. Clark, M.-T. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: Pre-training text encoders as discriminators rather than generators," *ICLR*, 2020. [Online]. Available: <https://openreview.net/pdf?id=r1xMH1BtvB>
- [15] Jin, Y., Zheng, T., Gao, C. and Xu, G., "MTMSN: Multi-Task and Multi-Modal Sequence Network for Facial Action Unit and Expression Recognition," In *EMNLP-IJCNLP*, pp. 3597-3602, 2019.
- [16] Geva, M., Gupta, A. and Berant, J., "Injecting numerical reasoning skills into language models," *ACL*, 2020. [Online], Available: <https://openreview.net/forum?id=zifzkEup4th>
- [17] Dheeru Dua, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner, "Drop: A reading comprehension benchmark requiring discrete reasoning over paragraphs," *NAACL*, pp. 2368-2378, 2019.
- [18] B. Kim, K. Ki, D. Lee & G. Gweon,

“Point to the Expression: Solving Algebraic Word Problems Using the Expression-Pointer Transformer Model,” *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 3768–3779, 2020.

[19] S. Roy, T. Vieira & D. Roth, “Reasoning about quantities in natural language,” *Transactions of the Association for Computational Linguistics*, Volume 3, pp. 1–13, 2015.

[20] S. Roy & D. Roth, “Solving General Arithmetic Word Problems,” *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1743–1752, 2015.

[21] <https://github.com/monologg/KoBERT-Transformers>

[22] <https://github.com/monologg/KoELECTRA>

지능 기술 기반 응용

This work was supported by the Basic Research Program through the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) under Grant (2020R1A4A1017775).



배 광 호

2022년 국립금오공과대학교 컴퓨터공학과 졸업. 관심분야는 NLP, Deep Learning.



여 상 엽

2023년 국립금오공과대학교 컴퓨터공학과 졸업(학사) 예정. 관심분야는 NLP, Automatic Programming, Deep Learning



정 유 철

1996년~2003년 아주대학교 정보 및 컴퓨터공학과 졸업(학사). 2003년~2005년 한국과학기술원 정보통신공학(로봇 AI전공) 졸업(석사). 2005년~2011년 한국과학기술원 전산학 졸업(박사). 2009년~2013년 한국전자통신연구원(선임연구원). 2013년~2017년 한국과학기술정보연구원(선임연구원). 2017년~2022년 국립금오공과대학교 컴퓨터공학과 조교수. 2022년~현재 국립금오공과대학교 인공지능공학과 부교수. 연구분야는 정보 검색 및 자연어 처리 분야, 음성인식 및 음성합성, 인공