대규모 언어 모델 및 인컨텍스트 러닝을 활용한 수치 추론 데이터셋 증강

황예찬°, 임진수°, 이영준, 최호진
한국과학기술원 전산학부
{yemintmint,j1n2u,yj2961,hojinc}@kaist.ac.kr

Numerical Reasoning Dataset Augmentation Using Large Language Model and In-Context Learning

Yechan Hwang°, Jinsu Lim°, Young-Jun Lee, Ho-Jin Choi
School of Computing, KAIST

요 약

본 논문에서는 대규모 언어 모델의 인컨텍스트 러닝과 프롬프팅을 활용하여 수치 추론 태스크 데이터셋을 효과적으로 증강시킬 수 있는 방법론을 제안한다. 또한 모델로 하여금 수치 추론 데이터의 이해를 도울 수 있는 전처리와 요구사항을 만족하지 못하는 결과물을 필터링 하는 검증 단계를 추가하여 생성되는 데이터의 퀄리티를 보장하고자 하였다. 이렇게 얻어진 증강 절차를 거쳐 증강을 진행한 뒤 추론용 모델 학습을 통해 다른 증강 방법론보다 우리의 방법론으로 증강된 데이터셋으로 학습된 모델이 더 높은 성능을 낼 수 있음을 보였다. 실험 결과 우리의 증강 데이터로 학습된 모델은 원본 데이터로 학습된 모델보다 모든 지표에서 2%p 이상의 성능 향상을 보였으며 다양한 케이스를 통해 우리의 모델이 수치 추론 학습 데이터의 다양성을 크게 향상시킬 수 있음을 확인하였다.

주제어: 데이터셋 증강, 수치 추론, 수학 문장제 문제, 대규모 언어 모델, 인컨텍스트 러닝

1. 서론

딥러닝을 활용한 모델 학습은 대개 학습 데이터의 다양성이 늘어남에 따라 성능이 높아지는 경향을 보인다.[1] 때문에 컴퓨터 비전 태스크에서는 학습 데이터의 다양성을 늘리기 위해 원본 이미지에 회전, 자르기, 색상 변환과 노이즈 추가 같은 다양한 종류의 변형을 가하는 데이터 증강(Data augmentation)이 매우 활발하게 사용되고 있다.[2] 학습 데이터의 다양성이 늘어남에 따라 높은 성능이 기대되는 것은 자연어 처리 태스크에서도 마찬가지이지만 자연어 문장 내에서의 배열된 단어의 순서가 바뀌거나, 하나의 단어만 삽입/삭제되어도 문장의 의미가 크게 달라질 수 있다는 등의 특성을 가진다. 자연어의 이러한 특성 때문에 자연어처리 태스크 데이터셋의 증강 방법론은 컴퓨터 비전 데이터셋의 증강 방법론보다 발전 속도가 더뎠다.

EDA(Easy data augmentation)[3]는 자연어처리 태스크의 한 분야인 텍스트 분류 데이터셋을 성공적으로 증강시킨 뒤 성능 향상을 이루어내는데에 성공한 선구적인 방법론이다. EDA는 문장 내 임의의 두 단어의 위치를 바꾸거나 임의의 단어를 삭제하는 등 비교적 방법론과 구현이 단순하고 어떤 자연어 문장에도 적용이 가능하다는 장점을 갖기 때문에 자연어 데이터셋의 대표적인 증강 방법으로 자리잡았다. 주어진 텍스트를 기존 언어에서 다른 언어로 번역한 뒤 이를 다시 원래 언어로 번역하여 데이터셋을 증강하는 역번역(Back translation)[4] 또한 번역 인공지능의 성능이 향상됨에 따라 자연어 증강을 위해 자주 사용되고 있다.

그러나 이러한 증강 기법들이 얼마나 효과적인지는 목표로하는 태스크에 따라 다를 수 있다. 질의 문답 태스크의 한 갈래인 수치 추론 태스크는 질문과 함께 주어진 설명으로부터 수치적인 추론을 진행한 뒤, 연산을 통해 최종 정답을 도출해야하는 태스크이다. 이때 EDA를 사용해 수치 추론 데이터 증강을 진행할 경우 여러 문제가 발생할 수 있다. 간단한 예시로, 여러 회사의 매출에 대한 설명과 *'A 회사와 B 회사의 매출액의 차이는 얼마인가?'*라는 질문이 주어졌을 때 A 또는 B 회사의 매출에 해당하는 수치가 EDA의 무작위 삭제(Random Deletion)에 의해 손실될 경우 올바른 추론을 진행할 수 없게 된다. 역번역 역시 수치 추론 데이터셋 증강에 적절할지는 미지수이다. 역번역을 사용하여 자연어 문장 증강을 진행할 경우 고유명사의 손실이 이루어 질 수 있으며, 새로 얻어진 데이터가 문제 해결에 필요한 조건들을 모두 온전히 유지하고 있다고 장담할 수 없기 때문이다.

이에 이 논문에서는 효과적인 수치 추론 태스크 데이터셋 증강을 위해 질문과 정답은 기존의 것을 그대로 유지하되, 질문 내에 함께 주어지는 설명을 체계적으로 재구성하여 새로운 질문을 만드는 방법론을 제안한다. 구체적으로, 대규모 언어 모델(Large language model)로 하여금 질문의 설명 텍스트 내에 있는 수치형 정보들을 이해시키고 이 정보들의 순서를 변경한 뒤 변경된 순서에 맞는 새로운 설명을 생성하는 프롬프트를 통해 [5]에서 공개된 한국어 수치 추론 데이터셋을 증강하였다.

이러한 방법론을 통해 한국어 수학 문장제 문제 데이터셋에 대해 증강을 진행한 뒤, 추론용 모델 학습을 통해 다른 자연어 데이터 증강 방법론보다 우리의 방법론으로 증강된 데이터셋으로 학습된 모델이 더 높은 성능을 낼 수 있음을 보였다.

표 1. 한국어 문장제 수학 문제 예시 (유형 : 산술 연산)

Type	Example
Question	학생 7명에게 사탕을 나누어 주려고 합니다. 한 사람당 3개씩 나누어 주었더니 7개가 남았습니다. 사탕을 학생들에게 똑같이 나누어 주려면 한 사람당 몇 개씩 주어야 합니까?
Numbers	"num0": 7, "num1": 1, "num2": 3, "num3": 7, "num4": 1
Equation	multiply(divide(add(multiply(num0, divide(num2, num1)),num3),num0),num4)
Answer	4

2. 제안하는 방법

EDA나 역번역 등 자연어 데이터셋 증강에 자주 사용되는 방법론들을 수치 추론 태스크 데이터셋에 적용할 경우, 증강된 데이터가 문제 해결에 반드시 필요한 정보를 손실할 가능성이 존재한다 (예시 : EDA에서의 무작위 삭제, 역번역에서의 고유명사 손실). 이러한 한계를 극복하기 위해 우리는 수치 추론 데이터셋의 원본 질문과 정답은 그대로 유지하되 질문과 함께 주어지는 설명을 체계적으로 재구성하는 과정을 거쳐 새로운 질문을 생성해낼 수 있는 방법론을 제시하였다. 구체적으로 우리는 CoT[6]와 G-Eval[7] 등의 선행 연구에서 대규모 언어 모델이 프롬프트 내부에서 추론을 진행하고 명시된 절차를 따르는 방법론으로부터 영감을 받아 인컨텍스트 러닝(In-Context learning) 기반의 프롬프팅으로 한국어 수학 문장제 문제 데이터셋[5]을 증강하는 프레임워크를 제안하였다. 수치 정보 이해, 수치 정보 순서 변경, 변경된 순서에 기반한 새로운 질문 생성을 수행하는 프롬프트를 이용하여 수치 추론 태스크 데이터셋을 증강한 뒤 새롭게 생성된 데이터가 유효한 데이터인지에 대한 몇 단계의 검증을 거쳐 새로운 데이터셋을 얻었다.

이 장에서 [5]에서 공개된 한국어 수학 문장제 문제 데이터셋에 대한 증강을 목표로 프롬프팅을 진행하여 기존 데이터셋을 증강하는 방법론에 대해 설명한다.

2.1 한국어 수학 문장제 문제

한국어 수학 문장제 문제 데이터셋(Ko-MWP)은 산술 연산, 크기 비교, 도형 서술 등 총 8가지 유형의 수학 문제들을 한국어로 서술한 데이터셋이다. 한국어 수학 문장제 문제 풀이용 모델은 표 1에서와 같이 (1) 질문(Question)을 입력으로 받은 뒤 (2) 풀이와 관련된 개체들로 부터 수치형 정보(Numbers)를 추출한 뒤 (3) 답을 도출하기 위한 식(Equation)을 생성한 뒤 식의 결과값을 계산하여 최종 정답(Answer)을 생성한다.

이러한 수학 문장제 문제 데이터셋을 증강시키기 위해 수치형 정보 혹은 식에서 사용되는 연산자를 변경하는 것은 기존의 식과는 전혀 다른 식 및 정답을 도출하게 된다. 이러한 방식으로 변화된 식과 기존 질문과의 관계를 체계적으로 검증하는 것은 어렵기 때문에, 본 논문에서는 이러한 한국어 문장제 수학 문제에 대하여 주요 수치형 정보와 식에서 쓰이는 연산자를 변경시키지 않으면서도 의미적으로 다른 문장을 생성하는 것을 목표로 한다.

2.2 수치 추론 데이터셋 증강을 위한 프롬프트

본 연구에서는 체계적이고 정교한 수치 추론 데이터셋 증강을 위해 다음의 세부 지시들을 수행할 수 있도록 프롬프트의 흐름을 구성하였다. 동시에 이후 과정에서 단계별 생성 결과들에 대해 검증을 진행하고 최종 결과 생성물을 추출할 수 있도록 하기 위하여 출력에 몇 가지 제약사항을 두어 정해진 형식에 맞추어 생성을 진행하도록 하였다.

2.2.1 수학 문장제 문제에서 수치정보에 대한 모델의 이해

효과적인 수치 추론 데이터셋 증강은 질문 내의 각 수치형 정보들이 나열된 순서와 식을 구성하는 연산자에 대한 깊은 이해를 필요로한다. 따라서 우리는 먼저 대규모 언어 모델로 하여금 질문의 텍스트 내에서 수치형 정보들이 갖는 의미를 잘 이해시키는 것을 목표로 하였다. 이를 위해 모델이 생성하는 문장 내부에 원본 질문에서의 수치형 정보들의 등장 순서를 명시하도록 지시하였다.

2.2.2 수치형 정보들의 순서 변경

우리는 수치 추론 문제를 푸는데에 필요한 정보들을 모두 유지하면서 동시에 원래와는 다른 질문을 얻기 위해 수치형 정보들의 순서를 변경한 뒤 해당 순서에 맞는 새로운 질문을 생성하는 방법을 택하였다. 이를 위한 중간 과정으로 앞 단계에서 얻어진 수치형 정보들의 순서에 변형을 가해 새로운 수치 정보 순서를 생성하도록 하였다. 또한 기존 순서에 대해 수행된 변화를 출력하게 하였고, 출력된 변화 내용과 상응하는 정보가 최종 질문에 반영 되었는지 확인하였다.

2.2.3 새로운 수치형 정보들의 순서에 기반한 질문 생성

이 단계에서는 앞 단계에서 얻어진 새로운 수치형 정보들의 순서를 바탕으로 새로운 질문을 생성하는 것을 목표로 하였다. 이때 수치형 정보의 순서는 다르지만 논리적으로는 동일한 질문을 생성하라는 제약사항을 통해 원래 질문에 담겨있는 정보들을 최대한 손실하지 않도록 유도하였다.

제시된 세 가지 세부 지시들을 바탕으로 프롬프트를 설명부, 처리 과정, 과정 수행시 제약사항으로 구성하였다 (표 2). 특히 모델의 입력으로 질문의 풀이과정이라고 볼 수 있는 식이 함께 주어지는 만큼, 문제 풀이에 도움이 될 수 있는 내용은 생성에서 제외하도록 하여 해답의 유출을 방지하고자 하였다.

표 2. 제안한 프롬프트 및 ChatGPT[8]에서 입출력 예시

프롬프트

너는 산술 문제를 제출하고 주어진 Answer에 대하여 평가를 하는 도우미이다. 기존 문제를 기반하여 새로운 문제를 제출하는 Task가 주어진다. 아래의 Steps에 따라 새로운 문제 생성을 하여라.

Process Steps:
Step 1: 주어진 Equation과 Numbers의 key와 value를 참조하여 Question에서 언어적 패턴에 대하여 Numbers의 순서를 출력을 하라.
Step 2: 생성한 Numbers의 순서를 새로운 언어적 생성 패턴으로 변형을 해라.
Step 3: 변형된 Numbers 생성 패턴에 따라, Answer와 New Equation에 답할 수 있는 주어진 Question과 다른 New Question을 생성해라.

Constraint:
1. 기존 Question과 논리적으로 동일한 New Question을 생성해라.
2. new Question에서 문제를 풀기위한 힌트를 제공하지 마라.
3. Step2의 변형에서 Numbers change 는 주어진 Numbers에 새로운 패턴으로 변함에 따른 변수의 변화 기록해라. 각 변수에 대하여 "original number" → "new mapping number" 로 표현해라.
4. numbers에서 등장한 변수를 한번씩 전부 사용해라.

입력 예시

Equation: fun_add(num0, num1)
Numbers: "nae0": "석진", "num0": 9, "num1": 5
Answer: 14
Question: 상자안에 9개의 공이 있습니다. 석진이가 5개의 공을 상자 안에 더 넣었습니다. 상자 안에 있는 공은 모두 몇 개입니까?

출력 예시

Generation Steps:
Step 1: 주어진 Equation과 Numbers의 key와 value를 참조하여 Question에서 언어적 패턴에 대하여 Numbers의 순서를 출력을 하라.
주어진 Equation은 "func_add(num0, num1)"이고, Numbers에는 "nae0"에 석진, "num0"에 9가, "num1"에 5가 대응됩니다. Question에서 "9개의 공"과 "5개의 공"이 언급되었으므로 순서를 유지한 채로 출력합니다.
Question에서 기존 Numbers의 순서: 석진, 9, 5

Step 2: 생성한 Numbers의 순서를 새로운 언어적 생성 패턴으로 변형을 해라.
순서를 변형하는 새로운 패턴을 사용하여 순서를 변경합니다. 예를 들어, 현재 순서가 [9, 5]이면 이를 [5, 9]로 변경할 수 있습니다.
New Numbers 순서: 석진, 5, 9
기존 numbers에 적용된 변수의 변화 **Numbers change: "nae0" → "nae0" , "num1" → "num0" , "num0" → "num1"**
New Numbers: "nae0": "석진", "num0": 5, "num1": 9

Step 3: 변형된 Numbers 생성 패턴에 따라, Answer와 Equation에 답할 수 있는 주어진 Question과 다른 언어적 패턴 형태의 문제인 New Question을 생성하라.
New Numbers 순서 와 New Equation에 답할 수 있도록 주어진 Question과 다른 New Question을 생성합니다.
New Question: 석진이는 5개의 공을 상자 안에 넣었습니다. 상자 안에 원래부터 9개의 공이 있었습니다. 상자 안에 있는 공은 모두 몇 개입니까?

2.3 프롬프트를 통한 질문 증강 및 검증

우리는 2.2에서 구성한 프롬프트를 사용하여 한국어 수학 문장제 문제 데이터셋에 대해 증강을 진행하기로 하였다. 이때 모델이 원본 질문에서 등장하는 암시적인 수치 정보들을 더 잘 이해할 수 있도록 하기 위한 전처리를 과정(2.3.1)을 추가하였다. 또한 새롭게 생성된 질문이 의도대로 생성되었는지를

Algorithm 1: 한국어 수학 문장제 데이터셋 증강 절차

Input: question, equation, numbers, answer, few-shot example, n_retry
Output: new_question, new_equation, new_numbers

```
n = 0
while n < n_retry do
    n += 1
    prep_question = replace_hangul_to_num(question)
    prompt = generate_prompt(prep_question, equation, numbers, answer, few_shot_example)
    response = LLM(prompt)
    change_history, new_numbers, new_question = parse_result(response)
    validation_of_change_history(numbers, change_history, new_numbers)
    validation_of_equation(equation, change_history, new_numbers, answer)
    validation_of_question(new_question, new_numbers)
    if prep_question == new_question then
        new_question = null
        continue
    else
        break
return new_question, new_equation, new_numbers
```

프롬프트 내부에서 검증(2.3.2-2.3.4)하여 요구사항을 만족하지 못하는 데이터셋은 새로 증강을 진행하도록 하여 최종 데이터셋의 퀄리티를 보장하고자 하였다. 원본 데이터에 대해 증강을 진행하여 결과물을 얻는 전체 과정은 알고리즘 1과 같다.

2.3.1 입력 질문에 대한 전처리

한국어는 수치와 관련된 표현이 매우 다양하다는 언어적 특성을 가진다. 예를 들어 숫자 3을 표현하는 문장 작성시 3을 제외하고도 '세', '셋', '석' 등으로 표현이 가능하다. 이렇듯 명시적으로 숫자를 통해 표현되지 않은 수치 정보가 모델의 입력으로 주어졌을 경우 수치 정보를 곧바로 이해하기 어려울 수 있다. 따라서 모델로 하여금 이러한 암시적인 수치 정보들을 더 잘 이해할 수 있도록 하기 위해 질문이 프롬프트에 입력으로 들어가기 전, 질문 내에서 한글로 표현된 수 관형사를 숫자로 변환하는 전처리를 수행하였다. 예를 들어 '전체 길이가 15cm일 때 삼각형의 한변의 길이는?'라는 질문의 경우, '삼각형'이라는 단어 내의 '삼'을 아라비아 숫자 3으로 변경하게 된다. 문장 내의 수 관형사를 판단하기 위한 방법으로 한국어 수학 문제 풀이 모델인 Ko-Graph2Tree[5]에서 사용한 Komoran[9] 형태소 분석기 및 사전 기반 매칭을 활용하였다.

2.3.2 검증 1: 수치형 정보의 순서 변화 기록 검증

한국어 수학 문장제 문제 데이터셋에서는 등장하는 수치형 정보의 순서가 변경되었을 경우 변경된 순서 정보를 반영하여 정답 식을 수정하는 과정이 필요하다. 이를 위해 우리는 모델이 변경을 가한 수치 정보의 순서를 생성 문장 내에 출력하게 하였다. 또한 생성 문장 내에 추론 과정을 출력시킴으로써 생성하는 문장들의 일관성이 깨지는 것을 제한하고자 하였다. 이후 모델이 생성한 변화 기록을 기존 질문의 수치형 정보에 적용한 결과가 모델이 생성한 출력과 일치하는지 검증하였다.

표 2의 출력 예시에서 수치형 정보의 순서를 변경하는 Step 2의 출력을 파싱한 결과를 바탕으로 기존 수치형 정보들에 순서적인 변화가 적용되었음을 확인할 수 있으며, 이 변경된 정보를 모델이 새롭게 생성한 New Numbers와 비교하게 된다.

2.3.3 검증 2: 변경된 변수를 사용하여 New Equation 생성 및 실행 결과 검증

모델에서 가해진 변형에 의해 정답 식 및 계산 결과가 바뀔 수 있기 때문에, 우리는 수치형 정보의 변화 기록을 사용하여 Equation을 업데이트하고 새로 얻어진 식의 실행 결과가 기존 Answer와 일치하는지 검증하였다. 이를 통하여 모델이 가한 변형으로 인해 최종 정답이 변경되는 것을 방지하였다.

2.3.4 검증 3: 증강된 질문과 변경된 수치형 정보에서 추출된 수치형 정보간 일치 여부 검증

증강된 질문과 새로 계산된 식 간의 매핑 여부 유지를 위해 증강된 질문(New Question)에서의 수치형 정보와 모델의 중간 단계에서 얻어진 수치형 정보(New Numbers) 사이의 일치 여부를 검증을 진행하였다. 구체적으로, 증강된 질문으로 부터 수치형 정보를 추출하기 위하여 기존 한국어 수학 문장제 데이터셋 생성에서 사용하는 전처리 로직을 활용하였고 여기서 추출된 정보들과 변경된 수치형 정보의 일치도를 검증하였다.

3. 실험 세팅

3.1 데이터셋 구성

본 실험에서는 실험 데이터셋으로 한국어 수학 문장제 문제 데이터셋을 사용하였다. 구체적으로, 미리 8개의 유형 중 하나의 유형으로 분류 되어있는 총 1,732개의 문제에 대해 각 유형마다 무작위로 80%의 데이터를 학습 데이터, 나머지 20%의 데이터를 평가 데이터로 나누었고 최종적으로 1,382개의 학습 데이터와 350개의 평가 데이터를 얻었다. 본 논문에서 제안된 방법론의 성능을 비교하기 위한 베이스라인 증강 방법론으로 EDA와 역번역을 택하여 증강을 진행하였고, 증강 과정에서 원본 질문에 존재하던 개체명이나 수치형 정보가 손실된 경우는 최종 데이터셋에서 제외하였다.

- EDA[3]
  EDA를 활용한 데이터셋 증강을 위해 원본 EDA 논문에서 사용된 네 가지 기법(Synonym Replacement, Random Insertion, Random Swap, Random Deletion)을 한국어 데이터에 대해 구현한 KoEDA[1]를 활용하였다. EDA를 통한 변화 정도를 조정하는 초매개변수인 α를 각각 0.1과 0.3으로 설정하여 증강을 진행하였으며, 이러한 절차를 통해 최종적으로 α=0.1에서 1,119개, α=0.3에서 849개의 데이터를 증강하였다.
- 역번역[4]
  역번역을 통한 증강에서는 기존의 한국어로 이루어진 질문을 영어로 번역한 뒤 다시 한국어로 재번역하는 방법을 택하였다. 본 실험에서는 구글의 T5[10] 모델을 한국어와 영어 말뭉치를 이용하여 재학습한 ke-t5 모델[2]을 사용하였다. EDA에서와 마찬가지로 증강된 문장에서 핵심 정보가 손실된 증강 데이터를 제외하였고 최종적으로 1,007개의 증강 데이터를 얻었다.
- Ours
  앞선 2장에서 제안한 우리의 방법론을 통해 증강을 진행하였고, 프롬프트 내부에서 few-shot 예시를 함께 사용하였다. 모델로는 gpt-3.5-turbo-0301[8]을 사용하였으며 n_retry=5로 설정하여 최종 1,072개의 증강 데이터를 얻었다.

3.2 실험 환경

세 증강 방법론으로 얻어진 증강 데이터셋을 원본 학습 데이터셋과 합쳐 최종 학습 데이터셋으로 사용하였으며, 모델로는 원본 데이터셋[5]의 베이스라인 모델인 Graph2Tree[11]의 한국어 구현체[3]를 사용하였다. 초매개변수로는 해당 모델의 기본 세팅인 배치 사이즈 32, 학습률을 0.001으로 세팅한 Adam 옵티마이저를 사용하였다. 또한 학습 횟수에 의한 유불리를 없애기 원본 데이터셋 학습용 모델의 학습 에포크를 75로 설정한 뒤, 증강된 데이터셋들에 대해서는 (데이터셋의 크기)*(에포크)가 원본 데이터셋용 모델과 비슷한 값을 갖도록 에포크를 75보다 작게 설정하였다. 각 데이터셋 별로 5번의 모델 학습을 진행한 뒤 350개의 평가 데이터 대해 얻어진 지표들을 평균내어 최종 평가 성능으로 기록하였다. 평가 지표로는 (1) 모델이 예측한 식으로부터 계산된 정답이 실제 정답과 일치하는지를 평가하는 Answer Accuracy와 (2) 평가 데이터의 식과 모델이 예측한 식의 일치 정도를 평가하는 Equation Accuracy를 사용하였다.

[1] https://github.com/toriving/KoEDA
[2] https://github.com/AIRC-KETI/ke-t5
[3] https://github.com/sogang-isds/Korean-MWPS

표 3. 한국어 수학 문장제 문제 평가 데이터셋에 대한 증강 기법 별 성능

학습 데이터셋	평균 Answer Acc	최대 Answer Acc	평균 Equation Acc	최대 Equation Acc
Ko-MWP	65.03%	66.57%	60.63%	61.71%
Ko-MWP + EDA(α=0.1)	64.97% (↓0.06)	66.28% (↓0.29)	60.57% (↓0.06)	61.71% (0.00)
Ko-MWP + EDA(α=0.3)	64.46% (↓0.57)	65.43% (↓1.14)	60.11% (↓0.52)	61.43% (↓0.28)
Ko-MWP + Back Translation	66.80% (↑1.77)	68.57% (↑2.0)	61.89% (↑1.32)	63.14% (↑1.43)
Ko-MWP + Ours	**67.31% (↑2.28)**	**68.57% (↑2.0)**	**62.91% (↑2.28)**	**63.71% (↑2.0)**

4. 실험 결과

4.1 정량적 평가

증강 기법 별로 얻어진 데이터셋으로 학습한 모델들에 대해 평가를 진행한 결과는 표 3에서 확인할 수 있다. 모델별 평균 Answer Accuracy, 최대 Answer Accuracy, 평균 Equation Accuracy 및 최대 Equation Accuracy를 백분율로 기록하였으며, 증강 없이 원본 학습 데이터셋을 학습한 모델 대비 성능의 증감을 함께 표기 하였다.

EDA 증강 데이터셋을 학습한 모델은 원본 데이터셋을 학습한 모델에 비해 비슷하거나 더 하락한 성능을 보였다. 구체적으로, α=0.1일 경우 원본 데이터셋용 모델과 비교하여 평균과 최대 성능수치에서는 큰 차이가 없었으며 α=0.3일때는 많은 단어들이 변경이 되면서 전체적으로 성능이 떨어지는 결과를 보였다. 이는 핵심 정보가 손실이 되는 케이스에 대하여 필터링을 진행하였음에도 단어 수준에서 무작위 삭제 혹은 변화를 가하는 EDA가 수치 추론에서 성능 하락을 야기할 수 있음을 시사한다. 역번역을 활용한 증강에서는 원본 데이터용 모델 대비 1.3%p 이상의 성능 향상을 보여 수치 추론 데이터셋 증강에 역번역이 활용될 가능성을 보였다.

이 논문의 방법론을 활용하여 얻어진 증강 데이터로 학습된 모델은 원본 데이터셋용 모델과 비교했을 때 모든 지표에서 2%p 이상의 성능 향상을 보였다. 이는 다른 방법론 대비 가장 큰 성능의 향상이며 우리의 방법론이 성공적으로 수치 추론 데이터셋의 다양성을 늘렸음을 시사한다.

4.2 정성적 평가

표 4 에서는 우리의 방법론으로 증강된 데이터의 예시이다. 성공 케이스들의 경우, 본 논문에서 제안한 검증 절차에 의해 원본 질문에서 등장한 수치형 정보들이 증강된 질문에서도 잘 포함되어 있음을 확인할 수 있다. 또한 여러 종류의 서술 형태 변화를 확인할 수 있는데, 성공 케이스 1, 2에서와 같이 원래 존재하던 단어를 삭제하거나, 원래의 단어(홀수)를 동일한 의미를 갖는 다른 단어(짝수만을 제외한 수)로 설명하거나, 성공 케이스 3과 같이 수치형 정보들의 순서를 바꾸는 것이 그 예시이다.

그러나 제안한 검증 절차들로는 모델이 질문 내의 주요 단어들과 수치형 정보들의 관계를 잘못 이해한 케이스를 필터링 할 수 없음을 확인하였다 (실패 케이스 1, 2). 또한 실패 케이스 3과 같이 증강된 질문이 원본 질문과 매우 유사한 형태와 의미를 갖는 경우도 존재하였다. 이러한 사항들은 추가적인 검증 절차들을 추가하여 극복해야하는 한계점이다.

표 4. 제안한 방법론을 통해 증강된 질문 예시

성공 케이스 1) 문장 내 단어 및 구분 변경
Original Question: 지구 환경의 날을 맞아 전교 학생 다모임에서 채소 씨앗을 나눠주었어요. 상추 씨앗을 475명이 받았고 부추 씨앗을 492명이 받았다면 씨앗을 받은 학생은 모두 몇 명인가요? Augmented Question: 지구 환경의 날을 맞아, 학생들에게 채소 씨앗을 나눠주었습니다. 상추 씨앗을 475명이, 부추 씨앗을 492명이 받았다면, 씨앗을 받은 학생은 총 몇 명인가요?
성공 케이스 2) 추론 문제 변경
Original Question: 2부터 100까지의 수 중에서 홀수만을 합한 값을 구하시오. Augmented Question: 2부터 100까지의 모든 수 중에서 짝수만을 제외한 수들의 합은 얼마인가요?
성공 케이스 3) 추론 순서 변경
Original Question: 5개의 수 1,5,2,9,7가 있습니다. 내림차순으로 나열했을 때 5 번째 수와 오름차순 나열했을때 2 번째 있는 수의 곱은 얼마입니까? Augmented Question: 5개의 수 1, 5, 2, 9, 7가 있습니다. 오름차순으로 나열했을 때 2 번째 있는 수와 내림차순으로 나열했을 때 5 번째 있는 수의 곱은 얼마입니까?
실패 케이스 1) 문제의 핵심 단어 변화
Original Question: 직육면체를 위에서 본 도형의 꼭짓점의 수는 몇 개입니까? Augmented Question: 정육면체를 위에서 본 도형의 꼭짓점의 수는 몇 개인가요?
실패 케이스 2) 심볼 간 관계 매칭 실패
Original Question: 혜진이의 나이는 12살이고, 어머니의 나이는 혜진이의 나이의 3배보다 4살이 더 많습니다. 아버지의 나이는 어머니의 나이보다 2살이 더 많은 동시에 동생의 나이에 6배일 때, 동생의 나이는 혜진이의 나이보다 몇 살이 더 적습니까? Augmented Question: 혜진이는 12살이고, 그녀의 어머니는 혜진이의 나이보다 3배에 4살을 더한 만큼 나이가 많습니다. 아버지의 나이는 어머니보다 2살 많으며, 동생의 나이는 혜진이의 나이의 6분의 1입니다. 그렇다면, 동생의 나이는 혜진이보다 몇 살 덜까요?
실패 케이스 3) 증강으로 인한 변화가 작음
Original Question: 둘레가 400m인 마름모 모양의 공원의 한 변의 길이는 몇 m입니까? Augmented Question: 둘레가 400m인 마름모 모양의 공원의 한 변의 길이는 몇 미터인가요?

5. 결론 및 향후계획

본 연구에서는 대규모 언어 모델과 인컨텍스트 러닝을 활용한 프롬프팅을 통해 한국어 수치 추론 데이터셋을 효과적으로 증강시킬 수 있는 방법론을 제안하였다. 또한 한국어의 언어적 특성을 고려하여 모델의 입력을 전처리하였고 3단계의 필터링 단계를 통해 생성된 데이터를 검증함으로써 생성되는 데이터의 퀄리티를 보장하였다. 실험 결과 기존에 자연어 데이터 증강을 위해 널리 쓰이던 EDA를 수치 추론 태스크에 활용하여 추론 모델 학습을 진행할 경우 증강 전보다 성능이 떨어짐을 확인하였고, 기존보다 성능을 향상시킬 수 있는 역번역을 통한 증강 역시 우리의 방법론보다 효과가 미미함을 확인하였다. 정성적 평가에서 확인한 몇 가지 문제점을 해결할 경우 우리의 방법론을 통한 증강의 효과는 더 높아질 것으로 예상한다. 추후 연구에서는 정성적 평가에서 확인한 한계를 극복하고, 추론 과정은 유지한 채로 문맥의 도메인을 변화시켜 더욱 다양한 데이터셋 구축할 수 있는 증강 방법론을 개발하고자 한다.

감사의 글

이 논문은 2023년도 정부(산업통상자원부)의 재원으로 한국산업기술평가관리원의 지원을 받아 수행된 연구임. (No. 1415184727, 전문개인투자자 맞춤형 투자 정보 제공을 위한 실시간 금융 텍스트 심층 이해 및 투자 정보 지원 서비스 개발)

참고문헌

[1] S. Y. Feng, V. Gangal, J. Wei, S. Chandar, S. Vosoughi, T. Mitamura, and E. Hovy, "A survey of data augmentation approaches for NLP," *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 968–988, Aug. 2021. [Online]. Available: https://aclanthology.org/2021.findings-acl.84

[2] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, Vol. 6, No. 1, pp. 1–48, 2019.

[3] J. Wei and K. Zou, "Eda: Easy data augmentation techniques for boosting performance on text classification tasks," *arXiv preprint arXiv:1901.11196*, 2019.

[4] R. Sennrich, B. Haddow, and A. Birch, "Improving neural machine translation models with monolingual data," *arXiv preprint arXiv:1511.06709*, 2015.

[5] 김동근, 이나연, 심현우, and 구명완, "Graph2tree 모델을 이용한 한국어 수학 문장제 문제 풀이," *정보과학회논문지*, Vol. 49, No. 10, pp. 807–815, 2022.

[6] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, Vol. 35, pp. 24 824–24 837, 2022.

[7] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "Gpteval: Nlg evaluation using gpt-4 with better human alignment," *arXiv preprint arXiv:2303.16634*, 2023.

[8] OpenAI, "Openai: Introducing chatgpt," https://openai.com/blog/chatgpt, 2022.

[9] G. L. Junsoo Shin, Junghwan Park, "komoran," https://github.com/shineware/KOMORAN, [Online; accessed 11-Sep-2023].

[10] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, Vol. 21, No. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[11] J. Zhang, L. Wang, R. K.-W. Lee, Y. Bin, Y. Wang, J. Shao, and E.-P. Lim, "Graph-to-tree learning for solving math word problems," *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 3928–3937, Jul. 2020.