Chain-of-Thought와 Program-aided Language Models을 이용한 전제-가설-라벨 삼중항 자동 생성

조희진[O1], 이창기[1], 배경만[2]
[1]강원대학교
[2]한국전자통신연구원
{thekey1027, leeck}@kangwon.ac.kr, kyoungman.bae@etri.re.kr

Generating Premise-Hypothesis-Label Triplet Using Chain-of-Thought and Program-aided Language Models

Hee-jin Cho[O1], Changki Lee[1], Kyoungman Bae[2]
[1]Kangwon National University
[2]Electronics and Telecommunications Research Institute

요 약

자연어 추론은 두 문장(전제, 가설)간의 관계를 이해하고 추론하여 함의, 모순, 중립 세 가지 범주로 분류하며, 전제-가설-라벨(PHL) 데이터셋을 활용하여 자연어 추론 모델을 학습한다. 그러나, 새로운 도메인에 자연어 추론을 적용할 경우 학습 데이터가 존재하지 않거나 이를 구축하는 데 많은 시간과 자원이 필요하다는 문제가 있다. 본 논문에서는 자연어 추론을 위한 학습 데이터인 전제-가설-라벨 삼중항을 자동 생성하기 위해 [1]에서 제안한 문장 변환 규칙 대신에 거대 언어 모델과 Chain-of-Thought(CoT), Program-aided Language Models(PaL) 등의 프롬프팅(Prompting) 방법을 이용하여 전제-가설-라벨 삼중항을 자동으로 생성하는 방법을 제안한다. 실험 결과, CoT와 PaL 프롬프팅 방법으로 자동 생성된 데이터의 품질이 기존 규칙이나 기본 프롬프팅 방법보다 더 우수하였다.

주제어: Natural Language Inference, Premise-Hypothesis-Label Triplet, Chain-of-Thought, Program-aided Language Models

1. 서론

자연어 추론(Natural Language Inference, NLI)은 주어진 두 문장(전제, 가설)간의 관계를 이해하고 추론하는 것을 목표로 하며, 다음의 세 가지 범주로 분류한다: 1. 함의(Entailment): 두 문장 중 하나가 다른 문장을 함축하거나 포함하는 경우; 2. 모순(Contradiction): 두 문장이 서로 모순된 정보를 포함하는 경우; 3. 중립(Neutral): 두 문장 간에 모순 관계가 없이 중립적인 관계에 있는 경우이다. 자연어 추론 작업은 주로 기계학습 및 딥러닝 모델을 사용하여 수행되며, 이러한 모델은 주어진 두 문장(전제, 가설)을 입력으로 받아 두 문장 간의 관계를 예측하도록 훈련된다.

이러한 자연어 추론 작업은 특정 도메인의 경우 학습 데이터가 존재하지 않거나 이를 구축하는 데 많은 시간과 자원이 필요하다는 문제가 있다. 이를 극복하기 위해 [1]에서는 문장 변환 규칙을 활용하여 자연어 추론 모델 학습에 사용할 수 있는 전제-가설-라벨 삼중항(Premise-Hypothesis-Label(PHL) Triplet)을 자동으로 생성하는 방법을 제안하였다.

본 논문에서는 학습 데이터가 존재하지 않는 신규 도메인에서 자연어 추론 학습 데이터를 거대 언어 모델과 In-context learning을 이용하여 생성할 수 있는 방법을 제안하고, 추가적으로 Chain-of-Thought(CoT)와 Program-aided Language Models(PaL) 방식을 사용하여 자연어 추론 학습 데이터 자동 생성의 정확도를 개선할 수 있음을 보인다.

2. 관련 연구

교사 학습 기반의 자연어 추론(Supervised NLI)에서는 주로 전제(P), 가설(H) 그리고 레이블(L: E, C, N)로 이루어진 전제-가설-라벨(PHL) 데이터셋을 활용하여 모델을 학습한다. 그러나 새로운 도메인에 자연어 추론을 적용할 경우 학습 데이터가 존재하지 않거나 이를 구축하는 데 많은 시간과 자원이 필요하다는 문제가 있다. 이러한 문제를 해결하기 위해, [1]에서는 문장 변환 규칙을 활용하여 자연어 추론 모델 학습에 사용할 수 있는 전제-가설-라벨 삼중항(PHL Triplet)을 자동으로 생성하는 방법을 제안하였다.

[2]에서는 기존 문장 변환 규칙을 추가 및 수정하여 자연어 추론 근거 데이터를 함께 추출할 수 있도록 하였고 거대 언어 모델과 In-context learning을 이용하여 자연어 추론 학습 데이터와 근거를 자동으로 추출하는 방법을 제안하였다.

최근 거대 언어 모델을 활용한 다양한 연구가 진행되었고, 여러 단계의 추론 과정을 생성하도록 유도하여 언어 모델의 수행 능력을 크게 향상시키는 Chain-of-Thought(CoT) 방법이 제안되었다[3]. 또한 Program-aided Language Models(PaL)에서는 자유 형식 텍스트

(Free-form Text)를 사용하는 CoT와는 다르게 자연어 문제를 읽고 중간 단계에서 Python 인터프리터와 같은 프로그래밍 런타임 방식으로 문제를 해결하는 방법을 제안하였다[4].

본 논문에서는 기존 연구[2]에 추가적으로 CoT와 PaL 기법을 적용하여 자연어 추론 학습 데이터의 자동 생성 방법을 다양화하고 자동 생성 정확도를 개선한 방법을 제안한다.

3. 전제-가설-라벨 삼중항 자동 생성

본 논문에서는 [1]에서 사용한 문장 변환 규칙 대신에 거대 언어 모델과 CoT, PaL 등의 프롬프팅(Prompting) 방법을 이용하여 자연어 추론 학습 데이터인 전제-가설-라벨 삼중항을 자동으로 생성하는 방법을 제안한다.

[1]에서 사용한 MS-COCO 텍스트 데이터에서 문장만을 추출하여 전제(Premise) 문장으로 사용하고, 문장 변환 기법에 CoT와 PaL 등의 프롬프팅 방법을 적용하여 가설(Hypothesis) 문장과 라벨(Label)을 추출한다.

3.1 Chain-of-Thought를 이용한 전제-가설-라벨 삼중항 자동 생성

Chain-of-Thought(CoT) 프롬프팅은 언어 모델이 추론을 할 수 있도록 논리적인 과정을 중간 단계에 추가하여 질의하는 방식이다[3]. 문제 해결 과정의 중간 단계를 명시적인 자연어로 제공함으로써 작업을 단계적으로 이해하여, 더 복잡한 작업에 대한 수행 능력을 향상시키는 결과를 얻는다.

다음은 자연어 추론 라벨에 대해 [1]에서 제안한 문장 변환 규칙을 CoT 프롬프트(Prompt)로 변경한 예시이다. 문장 변환 규칙에 품사나 의존 구문 분석 정보가 필요한 경우에는 CoT 프롬프트에만 품사 태거나 의존 구문 분석기의 결과를 넣어주었으며, 실제 CoT 추론 과정에서 거대 언어 모델이 품사 태거와 의존 구문 분석 결과를 올바르게 생성하여 그 결과를 사용함을 확인하였다.

1) 함의(Entailment)

- **Hypernym Substitution**: 문장의 명사를 상위어로 대체하여 가설을 만든다. CoT 프롬프트는 다음과 같다: "Q: In a sentence {a black dog is sleeping}, a noun with a parent word is changed to a parent word. A: The result of the pos_tag is [('a', 'DT'), ('black', 'JJ'), ('dog', 'NN'), ('is', 'VBZ'), ('sleeping', 'VBG')], The noun with hypernym in the sentence is 'dog' and the parent word of the word 'dog' is 'animal'. So The answer is {a black animal is sleeping}.
- **Pronoun Substitution**: 문장의 명사를 대명사로 대체하여 가설을 만든다. CoT 프롬프트: "Q: In a sentence {Two men sitting on top of a blue truck}, After changing the nouns to personal pronouns (He, She, It, They, etc.) in the input sentence. A: The word with a personal pronoun in the sentence is 'Two men', and the personal pronoun of 'Two men' is 'They'. So The answer is {They sitting on top of a blue truck}.
- **Counting**: 문장에서 공통 상위어를 가진 명사의 개수를 세고 "There are {count} {상위어}s present"와 같은 여러 템플릿을 사용하여 가설을 만든다. CoT 프롬프트: "Q: In a sentence {A motorbike and a car are parked}, Execute pos_tag to count the number of 'NN' tags with common hypernym, then change it to a sentence containing the hypernym and the number of 'NN' tags. A: The result of the pos_tag is [('A', 'DT'), ('motorbike', 'NN'), ('and', 'CC'), ('a', 'DT'), ('car', 'NN'), ('are', 'VBP'), ('parked', 'VBN')], The word that is the 'NN' tag in the sentence is 'motorbike','car' and the number is two. A common hypernym is 'automobiles'. So The answer is {two automobiles are parked}.
- **Paraphrasing**: 다른 단어를 사용하여 문장의 의미를 동일하게 표현한다. CoT 프롬프트: "Q: In a sentence {A brown purse is sitting on a green bench}, print paraphrased input sentences. A: 'purse' is changed to 'bag'. 'sitting on' changed to 'perched atop'. 'bench' is changed to 'seat'. So The Answer is {A brown bag is perched atop a green seat}.
- **Extracing Snippets**: 문장에서 핵심 의미만 남긴다. CoT 프롬프트: "Q: In a sentence {A person with red shirt is running near the garden}, Create sentences with only the core meaning left and the rest of the modifiers removed in the input sentence. A: A modifier that does not have a core meaning in a sentence is 'with red shirt'. So The answer is {A person is running near the garden}.

2) 모순(Contradiction)

- **Contradictory_words_adj**: 문장의 형용사를 반의어로 대체하여 가설을 만든다. CoT 프롬프트: "Q: In a sentence {He lives in a big house}, Run the pos_tag and change the word with the 'JJ' tag to contradictory word. A: The result of the pos_tag is [('He', 'PRP'), ('lives', 'VBZ'), ('in', 'IN'), ('a', 'DT'), ('big', 'JJ'), ('house', 'NN')], The word that is the 'JJ' tag in the sentence is 'big', if you change 'big' to contradictory 'JJ' tag word, it becomes 'small', So The answer is {He lives in a small house}.
- **Contradictory_words_noun**: 문장의 명사를 다른 명사로 대체하여 가설을 만든다. CoT 프롬프트: "Q: In a sentence {She is playing the piano}, Replaces a noun with another noun that contradicts it. A: The result of the pos_tag is [('she', 'PRP'), ('is', 'VBZ'), ('playing', 'VBG'), ('the', 'DT'), ('piano', 'NN')], The noun in the sentence is 'piano', and the contradictory word is 'violin'. So The answer is {She is playing the violin}.
- **Contradictory_verb**: 문장의 동사를 반의어로 대체하여 가설을 만든다. CoT 프롬프트: "Q: In a sentence {A girl is walking}, Replace the verb with another contradictory verb. A: The result of

the pos_tag is [('a', 'DT'), ('black', 'JJ'), ('dog', 'NN'), ('is', 'VBZ'), ('sleeping', 'VBG')], The verb in the sentence is 'walking', and the contradictory verb is 'driving'. So The answer is {A girl is driving}.

- **Number Substitution**: 문장에서 숫자에 해당하는 단어를 다른 숫자로 변경하여 가설을 만든다. CoT 프롬프트: "Q: In a sentence {two cars parked on the sidewalk on the street}, Run the pos_tag and change the word with the 'CD' tag to another word. A: The result of the pos_tag is [('two', 'CD'), ('cars', 'NNS'), ('parked', 'VBD'), ('on', 'IN'), ('the', 'DT'), ('sidewalk', 'NN'), ('on', 'IN'), ('the', 'DT'), ('street', 'NN')], The word that is the 'CD' tag in the sentence is 'two', if you change 'two' to another 'CD' tag word, it becomes 'seven', So The answer is {seven cars parked on the sidewalk on the street}.
- **Subject Object Swap**: 문장의 subject와 object에 해당하는 단어의 자리를 변경하여 가설을 만든다. CoT 프롬프트: "Q: In a sentence {A clock is standing on top of a concrete pillar}, Exchange the position of the word corresponding to the subject and object in the sentences. A: The result of dependency parsing of the input sentence is [['A', 'det'], ['clock', 'nsubj'], ['is', 'aux'], ['standing', 'ROOT'], ['on', 'prep'], ['top', 'pobj'], ['of', 'prep'], ['a', 'det'], ['concrete', 'amod'], ['pillar', 'pobj']], The subject of the sentence is 'clock' and the object is 'pillar', So The answer is {A pillar is standing on top of a concrete clock}.
- **Irrelevant Hypothesis**: 전제 문장과 전혀 관련 없는 다른 문장을 가설로 샘플링 한다. 이 경우는 CoT가 적용되지 않는다. 프롬프트: "Q: In a sentence {Sign for an ancient monument on the roadside}, Generates a completely unrelated sentence. A: {A man goes to strike a tennis ball}
- **Negation Introduction**: 문장에 부정을 적용한 문장을 가설로 사용한다. CoT 프롬프트: "Q: In a sentence {Empty fog covered streets in the night}, Apply negation to the sentence. A: The result of the pos_tag is [('Empty', 'NNP'), ('fog', 'NN'), ('covered', 'VBD'), ('streets', 'NNS'), ('in', 'IN'), ('the', 'DT'), ('night', 'NN')], Applying the negation to the verb 'covered' results in 'did not cover', So The answer is {Empty fog did not cover streets in the night}.

3) 중립(Neutral)

- **Adding Modifiers**: 문장의 명사에 수식어를 추가하여 가설로 사용한다. CoT 프롬프트: "Q: In a sentence {This is a bird sitting on a twig}, Add modifiers to decorate nouns in sentences. A: The result of the pos_tag is [('This', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('bird', 'NN'), ('sitting', 'VBG'), ('on', 'IN'), ('a', 'DT'), ('twig', 'NN')], The noun in the sentence is 'bird'. A suitable modifier is 'small', So The answer is {This is a small bird sitting on a twig}.
- **ConceptNet**: 문장에서 위치, 관계 정보를 추가하여 가설로 사용한다. CoT 프롬프트: "Q: In a sentence {Three horses standing by a tree eating the grass}, Add modifiers (in/at [mask][mask]) to the sentence. A: The appropriate word for the context is 'in the yard', So The answer is {Three horses standing by a tree eating the grass in the yard}.
- **Same Subject but Non Contradictory Verb**: 문장의 동사를 동일한 의미를 갖는 다른 동사로 변경하고 임의의 명사를 추가한다. CoT 프롬프트: "Q: In a sentence {A small child is sleeping in a bed with a bed cover}, From the input sentence, print a sentence with the same subject and no contradictory verbs with nouns added. A: The result of the pos_tag is [('A', 'DT'), ('small', 'JJ'), ('child', 'NN'), ('is', 'VBZ'), ('sleeping', 'VBG'), ('in', 'IN'), ('a', 'DT'), ('bed', 'NN'), ('with', 'IN'), ('a', 'DT'), ('bed', 'NN'), ('cover', 'NN')], Replace 'sleeping' with another non-contradictory verb, 'laying', and add the noun 'chair', So The answer is { A child laying in bed sleeping with a chair nearby}

3.1 Program-aided Language Models를 이용한 전제-가설-라벨 삼중항 자동 생성

Program-aided Language Models(PaL) 프롬프팅(Prompting)은 거대 언어 모델을 사용하여 Python과 같은 프로그래밍 언어의 지식과 논리를 활용하는 방식이다 [4]. CoT 방식은 중간 추론 단계가 자연어인 반면 PaL의 중간 추론 단계는 프로그램 소스 코드로 이루어져 있다.

다음은 자연어 추론 라벨에 대해 [1]에서 제안한 문장 변환 규칙을 PaL 프롬프트(Prompt)로 변경한 예시이다. CoT 프롬프팅과 동일하게, 문장 변환 규칙에 품사나 의존 구문 분석 정보가 필요한 경우에는 PaL 프롬프트에만 품사 태거나 의존 구문 분석기의 결과를 넣어주었으며, 실제 PaL 추론 과정에서 거대 언어 모델이 품사 태거와 의존 구문 분석 결과를 올바르게 생성하여 그 결과를 사용함을 확인하였다.

1) 함의(Entailment)

- **Hypernym Substitution**: 문장의 명사를 상위어로 대체하여 가설을 만든다. PaL 프롬프트는 다음과 같다: "Q: In a sentence {a black dog is sleeping}, a noun with a parent word is changed to a parent word. A: pos_tag = [('a', 'DT'), ('black', 'JJ'), ('dog', 'NN'), ('is', 'VBZ'), ('sleeping', 'VBG')], noun = 'dog', hypernym = 'animal', answer = sentence.replace(noun, hypernym), print(answer) = {a black animal is sleeping}.
- **Pronoun Substitution**: 문장의 명사를 대명사로 대체하여 가설을 만든다. PaL 프롬프트: "Q: In a sentence {Two men sitting on top of a blue truck}, After changing the nouns to personal pronouns (He, She, It, They, etc.) in the input sentence. A: noun = 'Two men', personal_pronouns = 'They', answer = sentence.replace(noun, personal_pronouns), print(answer) = {They sitting

on top of a blue truck}.

- **Counting**: 문장에서 공통 상위어를 가진 명사의 개수를 세고 "There are {count} {상위어}s present"와 같은 여러 템플릿을 사용하여 가설을 만든다. PaL 프롬프트: "Q: In a sentence {A motorbike and a car are parked}, Execute pos_tag to count the number of 'NN' tags with common hypernym, then change it to a sentence containing the hypernym and the number of 'NN' tags. A: pos_tag = [('A', 'DT'), ('motorbike', 'NN'), ('and', 'CC'), ('a', 'DT'), ('car', 'NN'), ('are', 'VBP'), ('parked', 'VBN')], noun = {'motorbike','car'}, noun_num = len(noun), hyper = hypernym(noun), print(answer) = {two automobiles are parked}.
- **Paraphrasing**: 다른 단어를 사용하여 문장의 의미를 동일하게 표현한다. PaL 프롬프트: "Q: In a sentence {A brown purse is sitting on a green bench}, print paraphrased input sentences. A: answer = sentence.replace('purse','bag'), answer = sentence.replace('sitting on','perched atop'), answer = sentence.replace('bench','seat), print(answer) = {A brown bag is perched atop a green seat}.
- **Extracing Snippets**: 문장에서 핵심 의미만 남긴다. PaL 프롬프트: "Q: In a sentence {A person with red shirt is running near the garden}, Create sentences with only the core meaning left and the rest of the modifiers removed in the input sentence. A: modifier = 'with red shirt', answer = sentence.remove(modeifier), print(answer) = {A person is running near the garden}.

2) 모순(Contradiction)

- **Contradictory_words_adj**: 문장의 형용사를 반의어로 대체하여 가설을 만든다. PaL 프롬프트: "Q: In a sentence {He lives in a big house}, Run the pos_tag and change the word with the 'JJ' tag to contradictory word. A: pos_tag = [('He', 'PRP'), ('lives', 'VBZ'), ('in', 'IN'), ('a', 'DT'), ('big', 'JJ'), ('house', 'NN')], adjective = 'big', contradictory_words = 'small', answer = sentence.replace(adjective, contradictory_words), print(answer) = {He lives in a small house}.
- **Contradictory_words_noun**: 문장의 명사를 다른 명사로 대체하여 가설을 만든다. PaL 프롬프트: "Q: In a sentence {She is playing the piano}, Replaces a noun with another noun that contradicts it. A: pos_tag = [('she', 'PRP'), ('is', 'VBZ'), ('playing', 'VBG'), ('the', 'DT'), ('piano', 'NN')], noun = 'piano', contradictory_words = 'violin', answer = sentence.replace(noun, contradictory_words), print(answer) = {She is playing the violin}.
- **Contradictory_verb**: 문장에서 동사를 반의어로 대체하여 가설을 만든다. PaL 프롬프트: "Q: In a sentence {he is playing}, Replace the verb with another contradictory verb. A: pos_tag = [('he', 'PRP'), ('is', 'VBZ'), ('playing', 'VBG')], verb = 'playing', contradictory_words = 'sleeping', answer = sentence.replace(verb, contradictory_words), print(answer) = {he is sleeping}.
- **Number Substitution**: 문장에서 숫자에 해당하는 단어를 다른 숫자로 변경하여 가설을 만든다. PaL 프롬프트: "Q: In a sentence {two cars parked on the sidewalk on the street}, Run the pos_tag and change the word with the 'CD' tag to another word. A: pos_tag = [('two', 'CD'), ('cars', 'NNS'), ('parked', 'VBD'), ('on', 'IN'), ('the', 'DT'), ('sidewalk', 'NN'), ('on', 'IN'), ('the', 'DT'), ('street', 'NN')], CD = 'two', another_nummod = random.randrange(1,11) answer = sentence.replace(CD, another_nummod), print(answer) = {seven cars parked on the sidewalk on the street}.
- **Subject Object Swap**: 문장의 subject와 object에 해당하는 단어의 자리를 변경하여 가설을 만든다. PaL 프롬프트: "Q: In a sentence {A clock is standing on top of a concrete pillar}, Exchange the position of the word corresponding to the subject and object in the sentences. A: dependency_parsing = [['A', 'det'], ['clock', 'nsubj'], ['is', 'aux'], ['standing', 'ROOT'], ['on', 'prep'], ['top', 'pobj'], ['of', 'prep'], ['a', 'det'], ['concrete', 'amod'], ['pillar', 'pobj']], subject = 'clock', object = 'pillar', answer = sentence.replace(subject,object), print(answer) = {A pillar is standing on top of a concrete clock}.
- **Irrelevant Hypothesis**: 전제 문장과 전혀 관련 없는 다른 문장을 가설로 샘플링 한다. 이 경우는 PaL이 적용되지 않는다. 프롬프트: "Q: In a sentence {Sign for an ancient monument on the roadside}, Generates a completely unrelated sentence. A: {A man goes to strike a tennis ball}
- **Negation Introduction**: 문장에 부정을 적용한 문장을 가설로 사용한다. PaL 프롬프트: "Q: In a sentence {Empty fog covered streets in the night}, Apply negation to the sentence. A: pos_tag = [('Empty', 'NNP'), ('fog', 'NN'), ('covered', 'VBD'), ('streets', 'NNS'), ('in', 'IN'), ('the', 'DT'), ('night', 'NN')], verb = 'covered', negation = 'did not cover', answer = sentence.replace(verb, negation), print(answer) = {Empty fog did not cover streets in the night}.

3) 중립(Neutral)

- **Adding Modifiers**: 문장의 명사에 대한 수식어를 추가하여 가설로 사용한다. PaL 프롬프트: "Q: In a sentence {This is a bird sitting on a twig}, Add modifiers to decorate nouns in sentences. A: pos_tag = [('This', 'DT'), ('is', 'VBZ'), ('a', 'DT'), ('bird', 'NN'), ('sitting', 'VBG'), ('on', 'IN'), ('a', 'DT'), ('twig', 'NN')], noun = 'bird', suitable_modifier = 'small bird', answer = sentence.replace(noun, suitable_modifier), print(answer) = {This is a small bird sitting on a twig}.

- **ConceptNet**: 문장에서 위치, 관계 정보를 추가하여 가설로 사용한다. PaL 프롬프트: "Q: In a sentence {Three horses standing by a tree eating the grass}, Add modifiers (in/at [mask][mask]) to the sentence. A: appropriate_word = 'in the yard', answer = sentence + appropriage_word, print(answer) = {Three horses standing by a tree eating the grass in the yard}.
- **Same Subject but Non Contradictory Verb**: 문장에서 동사를 동일한 의미를 갖는 다른 동사로 변경하고 임의의 명사를 추가한다. PaL 프롬프트: "Q: In a sentence {A small child is sleeping in a bed with a bed cover}, From the input sentence, print a sentence with the same subject and no contradictory verbs with nouns added. A: pos_tag = [('A', 'DT'), ('small', 'JJ'), ('child', 'NN'), ('is', 'VBZ'), ('sleeping', 'VBG'), ('in', 'IN'), ('a', 'DT'), ('bed', 'NN'), ('with', 'IN'), ('a', 'DT'), ('bed', 'NN'), ('cover', 'NN')], verb = 'sleeping', non_contradictory_verb = 'laying', add_noun = 'chair', answer = sentence.replace(verb, non_contradictory_verb), print(answer+add_noun) = {A child laying in bed sleeping with a chair nearby}

4. 실험 및 결과

본 논문에서는 거대 언어 모델로 GPT 3.5 모델 중 'text-davinci-003' 모델을 사용하였다. [1]에서 사용한 MS-COCO 데이터셋에서 문장을 추출하여 전제(Premise) 문장으로 사용하고 CoT와 PaL 등의 프롬프팅을 이용하여 전제-가설-라벨 삼중항 학습 데이터를 자동 생성하였으며, 한 규칙 당 40개, 한 프롬프팅 방법 마다 총 600개를 생성하였다. 자동 생성한 데이터는 수작업으로 정확도를 측정하였다.

표 1은 [1]에서 제안한 규칙 기반 변환 방법과 거대 언어 모델 기반의 기본적인 프롬프팅, CoT 프롬프팅, PaL 프롬프팅 방법의 전제-가설-라벨 삼중항 자동 생성 정확도를 측정한 것으로, 이를 통해 CoT와 PaL 프롬프팅 방법이 높은 정확도를 보임을 알 수 있고, 특히 CoT 프롬프팅 방법으로 전제-가설-라벨 삼중항 데이터를 생성한 것이 가장 높은 정확도를 보임을 알 수 있다. 그러나, 함의(Entailment)의 COUNT 규칙에서 CoT의 성능이 하락하였는데, 이는 문맥 정보를 고려하지 않고 품사 태깅을 통한 품사 정보만 이용했기 때문이라고 판단된다.

표 2는 규칙 기반, 기본 프롬프팅, CoT 프롬프팅, PaL 프롬프팅에 따른 Hypernym Substitution(HS) 규칙에서의 출력 결과이다. 규칙 기반 방법의 경우, 단어는 잘 변환하나 전반적으로 문맥에 맞게 유연하게 대처하지 못하는 경향을 보인다. 기본 프롬프팅의 경우, 문장이 올바르게 출력될 수 있도록 유연하게 대처하지만 변경이 필요한 대상 단어를 잘 인식하지 못하거나 일부 단어가 삭제되는 등 정확하지 못한 출력을 내는 경향을 보인다. CoT와 PaL의 경우, 추리(Reasoning) 과정을 통해 문장의 명사를 정확하게 인식하고 적절한 상위어로 대체하여 가설 문장을 만든다. 또한 경우에 따라 문장 수정이 필요한 경우 올바른 문장으로 수정되어 출력되는 경향을 보인다.

표 1. PHL 데이터 생성 정확도 성능

학습 데이터 생성 정확도		규칙(base)	기본 프롬프팅 (3-shot)	CoT (3-shot)	PaL (3-shot)
Entailment	HS	87.5	80	100	100
	PS	35	95	100	100
	COUNT	80	80	65	77.5
	PA	-	100	100	100
	ES	-	100	100	87.5
Contradiction	CW_adj	82.5	75	100	100
	CW_noun	70	75	100	100
	CV	72.5	87.5	100	85
	NS	97.5	50	100	100
	SOS	67.5	50	97.5	95
	IH	100	100	100	100
	NI	100	87.5	100	100
Neutral	AM	97.5	85	100	90
	CON	85	95	97.5	100

	SSNCV	-	62.5	100	95
학습 데이터 생성 평균 정확도		81.25	81.5 (+0.25)	**97.3 (+16.05)**	95.3 (+14.05)

표 2. 제안 방법 별 Hypernym Substitution 출력 결과 예시

규칙(base) 오답 예시	전제 문장: A young boy with a spoon looking at a birthday cupcake 가설 문장: A young boy with a container looking at a birthday cupcake
기본 프롬프팅 오답 예시	전제 문장: A man sits with a traditionally decorated cow 가설 문장: A man sits with a traditionally decorated animal cow
CoT 정답 예시	전제 문장: A man sits with a traditionally decorated cow Reasoning: The result of the pos_tag is [('A', 'DT'), ('man', 'NN'), ('sits', 'VBZ'), ('with', 'IN'), ('a', 'DT'), ('traditionally', 'RB'), ('decorated', 'VBN'), ('cow', 'NN')] The noun with hypernym in the sentence is 'cow' and the parent word of the word 'cow' is 'animal' So The answer is {A man sits with a traditionally decorated animal} 가설 문장: A man sits with a traditionally decorated animal
PaL 정답 예시	전제 문장: A man sits with a traditionally decorated cow Reasoning: pos_tag = [('A', 'DT'), ('man', 'NN'), ('sits', 'VBZ'), ('with', 'IN'), ('a', 'DT'), ('traditionally', 'RB'), ('decorated', 'VBN'), ('cow', 'NN')] noun = 'cow', hypernym = 'animal' answer = sentence.replace(noun, hypernym), print(answer) 가설 문장: A man sits with a traditionally decorated animal

5. 결론

본 논문에서는 자연어 추론을 위한 학습 데이터인 전제-가설-라벨 삼중항을 자동 생성하기 위해 [1]에서 제안한 문장 변환 규칙 대신에 거대 언어 모델과 CoT, PaL 등의 프롬프팅(Prompting) 방법을 이용하여 전제-가설-라벨 삼중항을 자동으로 생성하는 방법을 제안하였다. 실험 결과, CoT와 PaL 프롬프팅 방법 모두에서 생성된 데이터 품질이 더 우수하였으며 기존의 규칙이나 기본 프롬프팅에서 사용하기 어려웠던 품사 및 구문 정보를 추리(Reasoning) 과정에서 사용할 수 있음을 보였다.

향후 연구로는, 전제-가설-라벨 삼중항을 생성하기 위한 문장 변환 휴리스틱을 기존의 자연어 추론 학습 데이터로부터 자동으로 생성할 수 있는 기술에 대해 연구할 계획이다.

감사의 글

이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No.2023-0-00369, (4세부) 전문지식 대상 판단 결과의 이유/근거를 설명 가능한 전문가 의사결정 지원 인공지능 기술 개발)

참고문헌

[1] Neeraj Varshney et al., “Unsupervised Natural Language Inference Using PHL Triplet Generation”, ACL,pp.2003-2016, 2022.
[2] 조희진, “거대 언어 모델과 In-context learning을 이용한 자연어 추론 학습 데이터 자동 생성”, 한국정보과학회 학술발표논문집, pp.373-375, 2023.
[3] Jason Wei et al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”, NeuraIPS, 2022
[4] Luyu Gao et al., “PAL: Program-aided Language Models”, arXiv preprint arXiv:2211.10435, 2022
[5] S Bird et al., “NLTK: the natural language toolkit”, COLING/ACL, 2006