

# Seq2seq 기반 한국어 추상 의미 표상(AMR) 파싱 연구

HUANG HAO<sup>0</sup>, 박혜진<sup>1</sup>, 김한샘

연세대학교 언어정보학협동과정, 연세대학교 국어국문학과<sup>1</sup>  
alvis\_hh@yonsei.ac.kr, hyejin110062@naver.com, khss@yonsei.ac.kr

## A Study for Sequence-to-sequence based Korean Abstract Meaning Representation (AMR) Parsing

Hao Huang<sup>0</sup>, Hyejin Park, Hansaem Kim  
Institute of Language and Information Studies, Yonsei University

### 요약

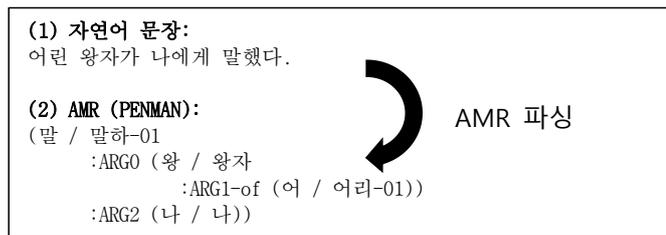
본 연구에서는 한국어 AMR 자동 파싱을 하기 위해 seq2seq 방법론을 적용하였다. Seq2seq 방법론은 AMR 파싱 태스크를 자연어 문장을 바탕으로 선형화된(linearization) 그래프의 문자열을 번역해내는 과정을 거친다. 본고는 Transformer 모델을 파싱 모델로 적용하여 2020년 공개된 한국어 AMR와 자체적으로 구축된 한국어 <어린 왕자> AMR 데이터에서 실험을 진행하였다. 이 연구에서 seq2seq 방법론 기반 한국어 AMR 파싱의 성능은 Smatch F1-Score 0.30으로 나타났다.

**주제어:** AMR, 한국어 <어린 왕자> AMR, 한국어 AMR 파싱, seq2seq 방법론.

### 1. 서론

추상 의미 표상(Abstract Meaning Representation, AMR)[1]은 루트(root)가 있는 유향 비순환 그래프(Directed Acyclic Graph; DAG)로 문장 단위의 의미를 표상한다. 그래프에서 노드(node)는 의미 개념을 나타내고 간선(edge)은 의미 개념 간의 관계를 나타낸다.

영어를 위해 개발한 의미 표상 체계인 AMR를 한국어로 적용하기 위한 기초 연구[2] 단계를 거쳐 한국어 AMR 가이드라인1.0[3]과 한국어 AMR 데이터를 공개된 바 있다[4]. 그림1은 자연어 문장(1)과 PENMAN 형식으로 주석된 한국어 AMR(2)의 예시이다.



예1: AMR 파싱; (1)amr 주석 대상 자연어 문장;  
(2) (1)에 대응하는 AMR 주석

그림1에서 제시하는 것과 같이 AMR 파싱(parsing)은 입력받은 자연어 문장을 자동으로 AMR로 변환시키는 자연어 처리 과제이다. 의미 파싱은 정보 추출, 자동 요약, 기계 독해 등 비정형 데이터로부터 중요한 정보를 얻거나 그 의미를 이해하여 적절한 대답을 생성하는 질의응답 기술 등 자연어 처리 분야에 두루 응용될 수 있다[4].

본 연구는 seq2seq 방법론을 적용하여 한국어 AMR 파싱을 시도한다. 본 연구는 Seq2seq 방법론 기반 한국어 AMR 파싱의 첫걸음을 내딛는 연구로서 파싱 결과에 대한 성능 평가, 오류 분석을 통해 후행 개선 방안을 논의한

다.

### 2. 관련 연구

#### 2.1. AMR 파싱 방법론

자연어 문장을 자동으로 AMR로 파싱하는 연구가 활발하게 진행되어 왔다. 초기의 AMR 파싱은 통계 방법론(statistical methods)을 활용했고, 대표적으로는 그래프 기반(graph-based) 방법론과 전이 기반(transition-based) 방법론을 기반으로 한 연구가 있었다. 최초의 작동 AMR 파서인 JAMR는 그래프 기반(graph-based) 방법론을 사용한 것이다[5]. JAMR은 통계 방법론을 기반으로 개념 식별과 관계 식별 절차를 거쳐 자연어를 AMR 그래프로 전환한다. 전이 기반 AMR 파서 CAMR도 제안되었다[6]. CAMR은 의존 구문 분석 파서(dependency parser)를 통해 자연어 문장을 분석 트리(수형도)로 전환하고 정해진 규칙에 따라 다시 분석 트리를 AMR 그래프로 전환한다.

딥러닝이 발전하면서 AMR 파싱을 위한 모델도 다수 발표되었다. 딥러닝 기반 AMR 파싱 모델은 다음과 같이 정리할 수 있다.

- 1) 신경망 전이 기반 파서[7][8][9][10][11].
  - 2) sequence-to-graph 기반 파서[12][13][14].
  - 3) sequence-to-sequence 기반 파서[15][16][17][18].
- 최근에 pre-training 기술을 활용해서 AMR 파싱 성능을 높이는 연구도 많아지고 있다. [12][14] 등의 연구에서는 BERT[19]를 인코더 계층에서 적용하였으며 [18][20] 등의 연구에서는 BART[21]를 적용하여 AMR 파싱을 수행했다.

#### 2.2. 한국어 AMR 파싱

한국어 AMR 파싱에 대한 선행연구는 [22]에서 의존 구

문 분석 트리로부터 AMR 그래프로 전환하는 graph-to-graph translation 모델을 제안하였고 그 성능으로 0.23의 Smatch F1-Score[24]을 보고한 바 있다. [4]에서는 영어 AMR 파싱 모델인 seq2graph[12]를 한국어 AMR 파싱을 적용해서 0.44의 Smatch F1-Score를 보고했다. [23]에서는 한국어 AMR 데이터 자동 증강 방안을 논의했다. 해당 연구에서는 Semi-AMR 데이터를 변환해내는 알고리즘을 제시하였으며 자동 변환 결과는 Gold-standard 데이터(인공 주석된 gold데이터와 10어절 이하의 Semi-AMR 데이터)에 대해 Smatch F1-Score가 0.46인 것으로 보고했다.

Seq2seq 모델은 그래프 기반 방법론이나 전이 기반 방법론에 비해 복잡한 데이터를 전처리 및 후처리할 필요가 없다는 장점을 가지고 있으며 보조적 태스크[17]와 pre-trained 언어 모델[18]을 잘 적합할 수 있다는 강점을 보여준다. 영어 AMR 파싱 연구에서 성능이 좋은 것으로 나타난 seq2seq 기반 알고리즘을 한국어에 적용해보는 선행 연구가 부재하기 때문에 본고는 한국어 AMR 파싱을 seq2seq 기반 방법론을 적용하여 baseline을 산출해 보고자 한다.

### 3. 한국어 AMR 데이터

데이터 셋	총	Train	Validation	Test
KAMR.2020	1253	1001	126	126
KAMR.tlp	1562	1274	145	143
본 연구	2781	2247	268	266

단위:문장

표1:데이터 셋 Split 분포

뉴스 기사, 위키 백과 문서, 한국어 기초 사전, 소설 <어린 왕자>(일부)등 장르의 텍스트를 주석 대상으로 구축된 1,253문장 규모의 한국어 AMR 말뭉치<sup>1</sup>(이하:KAMR.2020)가 공개된 바 있다[4]. 이 데이터는 1,001 문장 규모의 훈련 데이터, 126 문장 규모의 검증 데이터, 126 문장 규모의 평가 데이터로 분할된다[4].

이외에 본고에서는 자체적으로 1,562 문장 규모의 한국어 <어린 왕자> AMR<sup>2</sup>(이하:KAMR.tlp)를 구축하여 파싱 데이터셋으로 사용한다. AMR의 체계에 익숙한 주석자 3명이 34문장을 기준으로 산출한 주석자간 일치도(IAA; Inter-annotator agreement) Smatch F1-score는 0.75이다. [4]에서 보고된 KAMR.2020의 IAA가 0.75인점을 고려하여 구축된 데이터는 Gold AMR로 간주하여 사용할 수 있다. 본 연구의 데이터 또한 영어, 중국어 등 다른 언어의 <어린 왕자> AMR 데이터들의 Standard Split에 따라 1,274 문장의 훈련 데이터, 145 문장의 검증 데이터,

143 문장의 평가 데이터로 분할했다.

KAMR.2020에서 포함한 34문장의 <어린 왕자> 텍스트에 대한 AMR를 제외하여 최종적으로 본 연구에서 사용한 데이터셋은 표1와 같다. 데이터셋은 2,781 문장의 데이터를 2,247 문장의 훈련 데이터, 268 문장의 검증 데이터, 266 문장의 평가 데이터로 분할하였다.

## 4. Seq2seq 기반 한국어 AMR 파싱 방법론

### 4.1. 파싱 모델

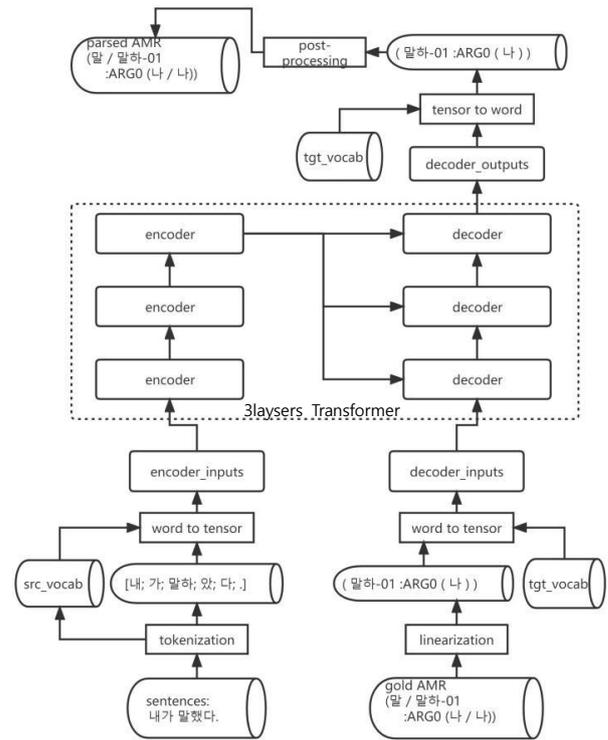


그림2: AMR 파싱 과정

Seq2seq 방법론은 AMR 파싱 태스크를 자연어 문장을 바탕으로 선형화된 그래프의 문자열을 번역해내는 과정을 거친다. 본 연구는 seq2seq 모델인 Transformer[25]를 파싱 모델의 기본적인 프레임으로 삼는다. 그림2에서 제시한 것처럼 자연어 문장을 encoder input으로 하고 선형화된 AMR를 decoder input과 decoder output으로 한다. 구체적인 실험 단계는 다음과 같다.

(1) **자연어 문장 토큰화.** 자연어 문장을 토큰화하여 토큰화된 형태소(사전 key)에게 유일한 인덱스(사전 val ue)를 부여해서 원천 사전(src\_vocab)을 만든다.

(2) **AMR 선형화.** 선형화(linearization) 과정에서는 PENMAN 형식으로 주석한 AMR 데이터를 한 줄의 AMR 문자열로 전환한다. 예2에서는 더 복잡한 문장과 그에 대응하는 선형화된 AMR 문자열을 제시한다. 선형화된 AMR에서 ‘달-01’, ‘편지’ 등의 개념 노드; ‘:ARG-i’, ‘:topic’ 등 관계 표지를 목표 사전(tgt\_vocab)의 key

<sup>1</sup> 한국어 AMR 말뭉치(2020) 데이터: <https://github.com/choe-hyonsu-gabrielle/korean-amr-corpus>

<sup>2</sup> 영어, 중국어, 브라질-포르투갈어 등 언어와 병렬적으로 구축된 AMR 말뭉치이다. 곧 공개할 예정이다.

로 삼고 key에게 인덱스를 부여해서 tgt\_vocab를 만든다.

```

자연어 문장:
그 애가 돌아왔다고 이내 편지를 보내 달라…….

AMR (PENMAN 그래프):
(달 / 달-01 :mode imperative
  :ARG1 (보 / 보내-01
    :ARG1 (편 / 편지
      :topic (돌 / 돌아오-01
        :ARG0 (애 / 애
          :mod (그 / 그))))
    :manner (이 / 이내)))

선형화된 AMR:
( 달-01 :mode_imperative :ARG1 ( 보내-01 :ARG1 ( 편지 :topic ( 돌
아오-01 :ARG0 ( 애 :mod ( 그 ) ) ) ) :manner ( 이내 ) ) ) )
    
```

예2: AMR 선형화

(3) **모델 변수 조정.** 2,247 문장의 훈련 데이터, 268 문장의 검증 데이터를 통해 모델 변수(parameter)를 조정하여 최고 성능의 모델을 저장한다.

(4) **파싱 및 후처리.** 최고 성능의 모델에 266개의 테스트 문장을 입력해서 파싱된 AMR 데이터를 얻는다. 파싱된 AMR 또한 선형 문자열 형태이므로 후처리 과정을 통해 PENMAN 형식의 AMR로 전환한다.

(5) **파싱 성능 보고.** Test 데이터의 gold AMR과 모델이 산출한 parsed AMR 간의 일치도를 Smatch 점수로 보고한다.

#### 4.2. 모델 변수(parameter)

실험과정 중에 최고 성능을 보인 모델의 변수는 다음과 같다.

- Embedding Size=512;
- FeedForward dimension=2048;
- Dimension of K(=Q),V=64;
- Encoder Layers=Decoder Layers=3;
- Multi-Head Attention=8;
- Batchsize=16
- Encoder input max sequence length=118;
- Decoder input max sequence length=380.

학습 과정에서는 SGD Optimizer로 최적화를 수행하며 (learning rate=0.001, momentum=0.995), 50 epoch 동안 학습을 진행하였다. 평균 한 epoch의 소요 시간은 31.4 s다. 전체 50 epoch 동안의 학습 과정에서 검증 데이터 loss가 가장 낮은 지점은 24 epoch였다(Train loss=1.29 6; Val loss=2.301).

#### 4.3. 파싱 결과

모델의 학습 결과 전체 266 문장의 평가 데이터에 대한 SMATCH score는 Precision 0.28, Recall 0.32, F1-score 0.30의 낮은 점수를 기록했다. KAMR.2020 123문장의

평가 데이터에 대한 Smatch F1-score는 0.27이다. KAMR.tlp 143문장의 평가 데이터에 대한 Smatch F1-score는 0.34로 기록했다.

	P	R	F1
전체 test data(266문장)	0.28	0.32	0.30
KAMR.2020 test data(123문장)	0.30	0.25	0.27
KAMR.tlp test data(143문장)	0.32	0.37	0.34
Unlabled	0.41	0.35	0.37
No WSD	0.31	0.26	0.28
Concepts	0.21	0.17	0.19
Named Ent.	0.17	0.23	0.20
Negations	0.00	0.00	0.00
Wikification	0.13	0.17	0.15
SRL	0.25	0.21	0.23

표2: AMR 파싱 성능

이외에 amr-evaluation[26]에 따라 Unlabled(관계 표지 labels 제외하는 smatch 점수), No WSD(‘하-01’, ‘하-02’와 같은 Propbank 용언 프레임의 하위 frameset를 구별하지 않는 Smatch 점수), Concepts(개념 인식 점수), Named Ent.(개체명 인식 점수), Negations(‘:polarity’ 부정에 대한 인식 점수), Wikification(‘:wiki’ 위키 개념에 대한 인식 점수), SRL(‘:ARG-i’ 의미역에 대한 인식 점수)를 기록하였다. 이 중에 Negations와 Wikification 점수가 가장 낮은 것으로 보인다. 본 모델은 부정과 개체명(AMR 주석에서 ‘:wiki’는 무조건 개체명 표상과 같이 나타난다.)에 대한 인식 성능이 부족하다고 할 수 있다.

#### 4.4. 오류 분석

파싱 결과에서 보인 것처럼 본 모델은 부정과 개체명에 대한 인식 성능이 부족하다. 특히 부정 인식에 대한 점수가 0.00이므로 266개의 평가 문장에 대한 ‘:polarity’, 즉 부정에 대한 파싱은 전부 잘못 주석되었다.

- (1) ‘:polarity’ 부정에 대한 인식 오류

```

자연어 문장:
그러나 어린 왕자는 대답이 없었다.

Gold AMR:
(c / contrast-01
  :ARG2 (대 / 대답-01 :polarity -
    :ARG0 (왕 / 왕자
      :ARG1-of (어 / 어리-01))))

Parsed AMR
(c / contrast-01
  :ARG2 (대 / 대답-01
    :ARG0 (왕 / 왕자
      :ARG1-of (어 / 어리-01))))
    
```

위에서 제시한 것과 같이 파싱된 AMR 데이터에서 부정

을 표상하는 ‘:polarity -’가 누락되었다. 훈련 데이터에서 ‘대답-01’이 56번이 나타나는데 그중에 10번만 ‘polarity -’가 ‘대답-01’ 뒤에 나타나는 불균형적인 상황이다. 훈련 데이터가 2,247 문장밖에 없는 적은 규모이기 때문에 그 중에서 부정이 나타나는 문장이 더욱 적을 수밖에 없었다. 학습 데이터의 절대적인 양이 부족하기 때문에 본 모델에서는 ‘대답이 없었다’가 부정으로 인식되지 않는 것으로 판단된다. 한국어의 부정은 문법적 부정, 어휘적 부정 등의 유형이 있고 표지가 다양하기 때문에 주석의 정확도, 주석자 간 일치도가 낮다.

(2) ‘:wiki’ 및 개체명에 대한 인식 오류

```

자연어 문장(일부)
.....주변 4장(미국·중국·일본·러시아).....

Gold AMR
(강 / 강
  :mod 4
  :mod (주 / 주변)
  :example (c2 / country :wiki "미국" :name (이2 / 이름 :op1
"미국")))
  :example (c3 / country :wiki "중화인민공화국" :name (이3 /
이름 :op1 "중국"))
  :example (c4 / country :wiki "일본" :name (이4 / 이름 :op1
"일본"))
  :example (c5 / country :wiki "러시아" :name (이5 / 이름 :o
p1 "러시아"))))

Parsed AMR
(i / include-91
  :ARG1 (등 / 등
  :op1 (c / country :wiki "미국" :name (이 / 이름 :op1 "미국
"))
  :op2 (a2 / and
  :op1 (c2 / company :wiki "중국" :name (이2 / 이름 :op1
"중국"))
  :op2 (c3 / company :wiki "미국" :name (이3 / 이름 :op1
"미국")))))
    
```

파싱된 AMR 데이터에서는 훈련 데이터에서 가장 많이 나타난 ‘미국’과 ‘중국’만 개체명으로 인식되었다. 개체명에 대한 표상은 일반적으로 ‘(개체명범주 :wiki '개체명' :name (이름 :op1 '개체명'))’와 같은 긴 문자열로 이루어져야 하기 때문에 파싱 모델에서 쉽게 해결하기 어려운 문제가 될 수 있다. [15]에서 개체명 익명화(named entity anonymization) 작업이 seq2seq 기반 AMR 파싱 성능을 향상할 수 있다는 결과를 보여주었다. 선형화된 AMR 중에 ‘(country :wiki "미국" :name (이름 :op1 "미국"))’와 같은 개체명 표상을 ‘country\_i’로 익명화 처리를 하여 파싱 난도를 낮추는 것이다.

5. 결론

본 논문에서는 이전에 한국어 AMR 파싱 연구에서 시도되지 않았던 seq2seq 방법론을 AMR 파싱 모델에 적용하

였다. Transformer 모델을 기준 파싱 모델로 삼아 Smatch Precision 0.28, Recall 0.32, F1-score 0.30의 성능을 기록하였다. 또한 후속 연구를 위해 seq2seq 방법론 기반 한국어 AMR 파싱 baseline을 산출하였다. 현재 유의미한 수준의 파싱 성능을 달성하지는 못했으나 앞으로 충분한 규모의 gold AMR 데이터셋이 확보된다면 복잡한 전처리와 후처리를 할 필요가 없는 seq2seq 파싱 모델로 바로 적용할 수 있다는 것에서 본 연구의 의의를 찾을 수 있다.

본고는 Seq2seq 모델 파싱 성능이 훈련 데이터로서의 인공 주석 데이터 규모와 밀접한 관계가 있는 점을 고려하여 1,253문장 규모의 KAMR.2020 데이터 외에 직접 구축한 1,562문장 규모의 KAMR.tlp 데이터를 사용하였다. 다만 2,247문장의 훈련 데이터 규모 또한 충분하지 않기 때문에 파싱 성능이 낮은 수준으로 제한되었다. [4]에서 보고한 것과 같이 현 단계에서 한국어 AMR 파싱 성능을 더 향상시키기 위해 데이터 규모를 확충하는 것이 최우선적인 과제이다. 한국어 AMR 데이터 인공 주석 도구 개발과 데이터 자동 증강 방법론을 향후 연구로 추진할 필요가 있다.

본 연구에서 사용하는 데이터셋이 한국어 AMR 파싱 선행 연구[4][22]와 다르기 때문에 모델 간에 성능 비교가 불가능한 것으로 어쩔 수 없는 한계가 있으므로 향후 연구에서 해결할 예정이다.

참고문헌

[1] Banarescu, Laura, et al., “Abstract meaning representation for sembanking”, *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 2013.

[2] 최현수, 한지윤, 오태환, 김한샘., “한국어 추상 의미 표상(AMR)을 위한 기초 연구.” 언어. 제44권-4호. 한국어언어학회. pp.943-969, 2019.

[3] 최현수 외, “문장 의미의 그래프 구조 표상을 위한 한국어 abstract meaning representation 가이드라인,” 제31회 한글 및 한국어 정보처리 학술대회 논문집, pp. 252- 257, 2019.

[4] 최현수, “한국어 추상 의미 표상(AMR) 및 의미 파싱 연구,” Master’s thesis, 연세대학교 일반대학원, 서울, 2020.

[5] Flanigan, Jeffrey, et al. "A discriminative graph-based parser for the abstract meaning representation." *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2014.

[6] Wang, Chuan, et al. "CAMR at SemEval-2016 task 8: An extended transition-based AMR parser." *Proceedings of the 10th international workshop on semantic evaluation (semeval-2016)*. 2016.

[7] Ballesteros, Miguel, and Yaser Al-Onaizan. "AMR parsing using stack-LSTMs." *arXiv preprint arXiv:1707.07755*, 2017.

[8] Liu, Yijia, et al. "An AMR aligner tuned by tran

- sition-based parser." arXiv preprint arXiv:1810.03541, 2018.
- [9] Astudillo, Ramón Fernandez, et al. "Transition-based parsing with stack-transformers." arXiv preprint arXiv:2010.10669, 2020.
- [10] Zhou, Jiawei, et al. "AMR parsing with action-pointer transformer." arXiv preprint arXiv:2104.14674, 2021.
- [11] Drozdov, Andrew, et al. "Inducing and Using Alignments for Transition-based AMR Parsing." arXiv preprint arXiv:2205.01464 (2022).
- [12] Zhang, Sheng, et al. AMR Parsing as Sequence-to-Graph Transduction. arXiv preprint arXiv:1905.08704, 2019.
- [13] Lyu, Chunchuan, Shay B. Cohen, and Ivan Titov. "A differentiable relaxation of graph segmentation and alignment for amr parsing." arXiv preprint arXiv:2010.12676, 2020.
- [14] Cai, Deng, and Wai Lam. "Amr parsing via graph-sequence iterative inference." arXiv preprint arXiv:2004.05572, 2020.
- [15] Konstas, Ioannis, et al. "Neural amr: Sequence-to-sequence models for parsing and generation." arXiv preprint arXiv:1704.08381, 2017.
- [16] Peng, Xiaochang, et al. "Sequence-to-sequence models for cache transition systems." Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2018.
- [17] Xu, Dongqin, et al. "Improving AMR parsing with sequence-to-sequence pre-training." arXiv preprint arXiv:2010.01771, 2020.
- [18] Bevilacqua, Michele, Rexhina Blloshmi, and Roberto Navigli. "One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. No. 14. 2021.
- [19] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805, 2018.
- [20] Bai, Xuefeng, Yulong Chen, and Yue Zhang. "Graph Pre-training for AMR Parsing and Generation." arXiv preprint arXiv:2203.07836, 2022.
- [21] Lewis, Mike, et al. "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." arXiv preprint arXiv:1910.13461, 2019.
- [22] 김영. "Graph-based neural AMR parser for Korean semantic parsing." Master's thesis, 연세대학교 일반대학원, 서울, 2020.
- [23] Choe, Hyonsu, et al. "Automatic Data Augmentation for Korean AMR Sembanking & Parsing." Annual Conference on Human and Language Technology. Human and Language Technology, 2020.
- [24] S. Cai and K. Knight, "Smatch: an evaluation metric for semantic feature structures," Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 748-752, 2013.
- [25] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30, 2017.
- [26] Damonte, Marco, Shay B. Cohen, and Giorgio Satta. "An incremental parser for abstract meaning representation." arXiv preprint arXiv:1608.06111, 2016.