kt GenieLabs

Dev-Challenge 2022

과제 1 : T5 모델 기반 NER

Team : 왜 오수새인가

고려대학교 산업경영공학부 석사과정 오수지

세종대학교 데이터사이언스학과 박새란

00 목차

kt GenieLabs Dev-Challenge 2022

1

EDA

1. 통계적 분석
2. Named Entity 분석
   ① NER 가설 1
   ② NER 가설 2

2

학습 데이터 생성

1. 데이터 전처리
2. 학습 데이터 - 검증 데이터 분리

3

모델 학습 방식

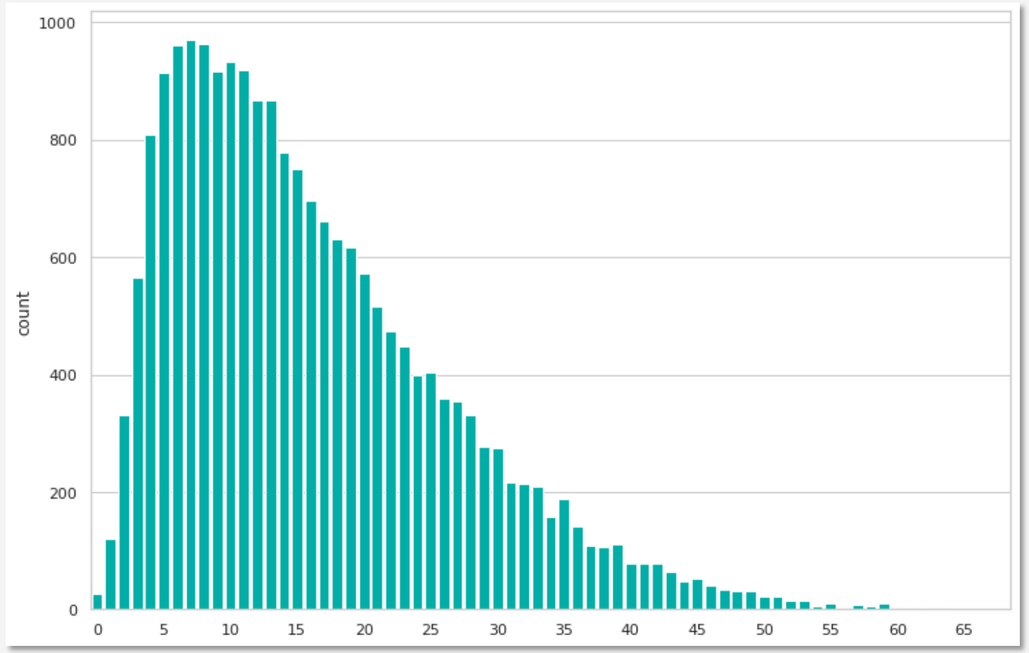
1. Text-to-text Multi-task Learning
2. POS Embedding
3. Model Lightweight
   ① Knowledge Distillation
   ② Vocab Pruning

1. Raw text 분석

1. 데이터 개수 확인
   - train : 20802개
   - test : 5201개
2. 데이터 길이 확인 : 평균 54.4자, 최대 148자로 구성
   - 분위수에 따른 데이터 길이 : 25%: 32.0 || 50%: 48.0 || 75%: 72.0 || 99%: 130.0

2. 토큰의 개수

1. 데이터별 토큰 개수 확인 : 평균 23개, 최대 89개로 구성
   - 분위수에 따른 토큰 개수 : 25%: 15.0 || 50%: 21.0 || 75%: 29.0 || 99%: 54.0
2. 데이터별 토큰 길이 확인 : 평균 2.5, 최대 12
   - 분위수에 따른 토큰 길이 : 25%: 1.0 || 50%: 3.0 || 75%: 3.0 || 99%: 6.0



[그래프1] 토큰 개수 별 데이터 분포

1. Entity Tag 종류 : 'DT', 'LC', 'OG', 'PS', 'QT', 'TI'

- person(PS), location(LC), organization(OG), date(DT), time(TI), quantity(QT)
- 가장 자주 나타나는 Tag: PS
- 가장 적게 나타나는 Tag: TI

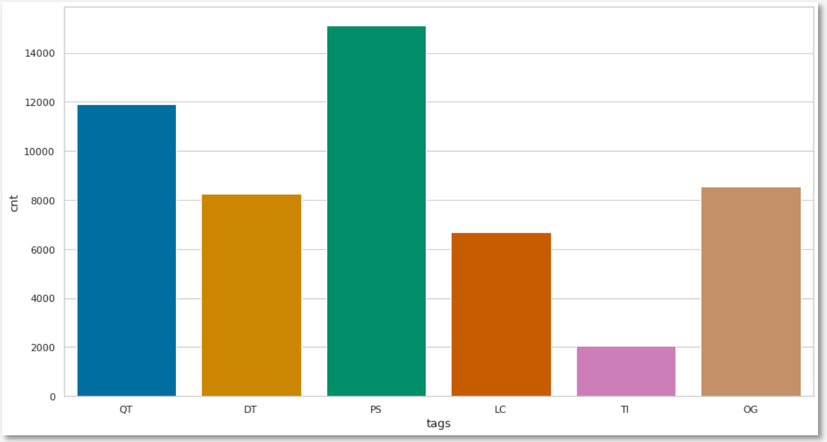
2. NE 개수 확인

- 전체 데이터셋 내 NE 총 개수 : 52562개
- 전체 데이터셋 내 중복 제거 후 NE 총 개수 : 17872개
- 반복 등장하는 NE가 다수 존재하는 것을 확인할 수 있음

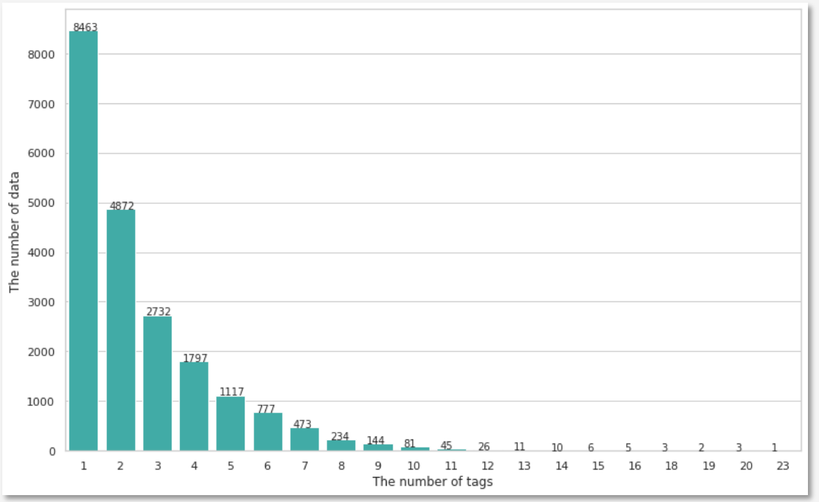
3. Tag 개수에 따른 데이터 분포

- 데이터별 텍스트에 포함된 Tag 개수가 1~23개로 다양하지만 균일하게 분포하지 않음

  → 학습 데이터와 검증 데이터를 나눌 때 Tag 개수 분포도 고려해야 함



[그래프2] 각 Tag별 Entity 분포



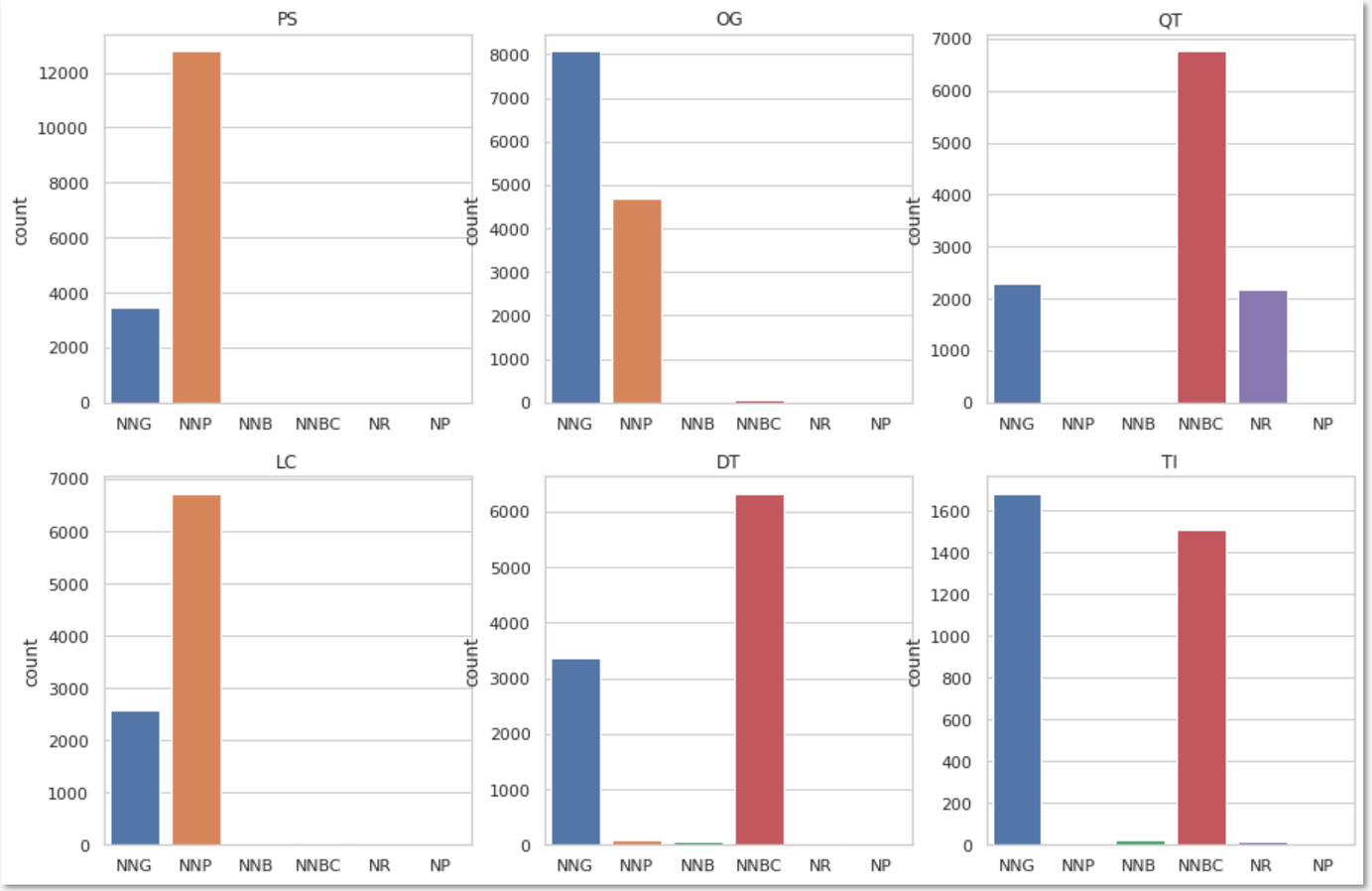
[그래프3] Tag 개수에 따른 데이터 분포

NER 가설 1. Named Entity의 Tag에 따라 자주 속하는 명사가 다르다.

- 명사 종류

품사 태그	설명
NNG	일반 명사
NNP	고유 명사
NNB	의존 명사
NNBC	단위를 나타내는 명사
NR	수사
NP	대명사

- PS, OG, LC는 일반명사와 고유명사가 많음
- QT, DT, TI는 단위 명사가 가장 많이 나타남
- 각 Entity마다 분포의 특징이 있으므로, Entity의 **명사 종류를 반영한 POS Embedding**을 통해 Entity의 추가적인 정보를 학습할 수 있음



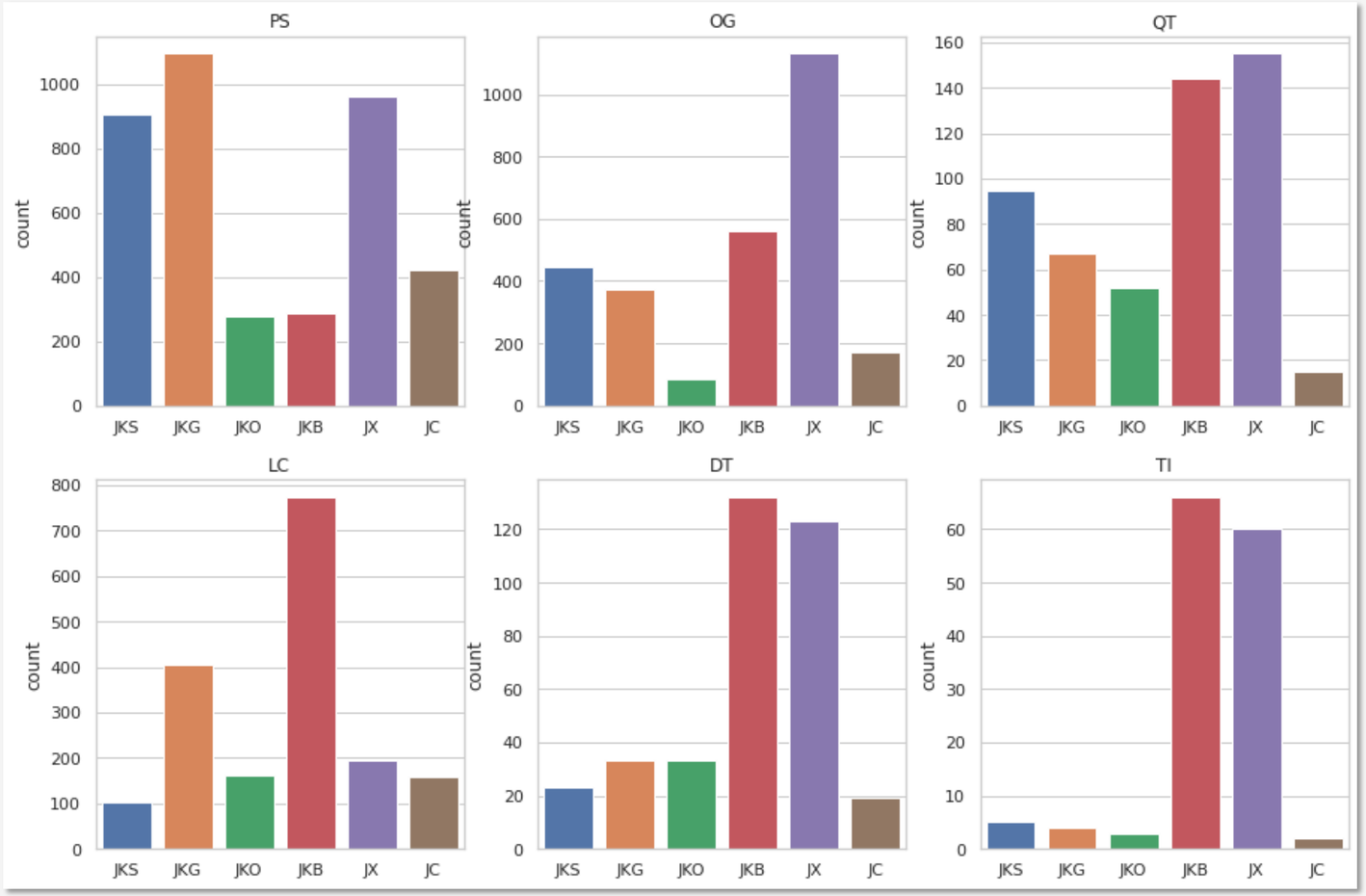
[그래프4] 각 Tag 별 명사 분포

NER 가설 2. Named Entity의 Tag에 따라 자주 함께 사용되는 조사가 다르다.

- 조사 종류

품사 태그	설명
JKS	주격 조사
JKG	관형격 조사
JKO	목적격 조사
JKB	부사격 조사
JX	보조사
JC	접속 조사

- PS는 주격, 관형격, 보조사와 자주 함께 사용됨
- QT, DT, TI는 부사격 조사, 보조사와 자주 함께 사용됨
- 각 Entity마다 분포의 특징이 있으므로, **조사의 종류를 반영한 POS Embedding**을 통해 Entity의 추가적인 정보를 학습할 수 있음



[그래프5] 각 Tag 별 조사 분포

Data Description

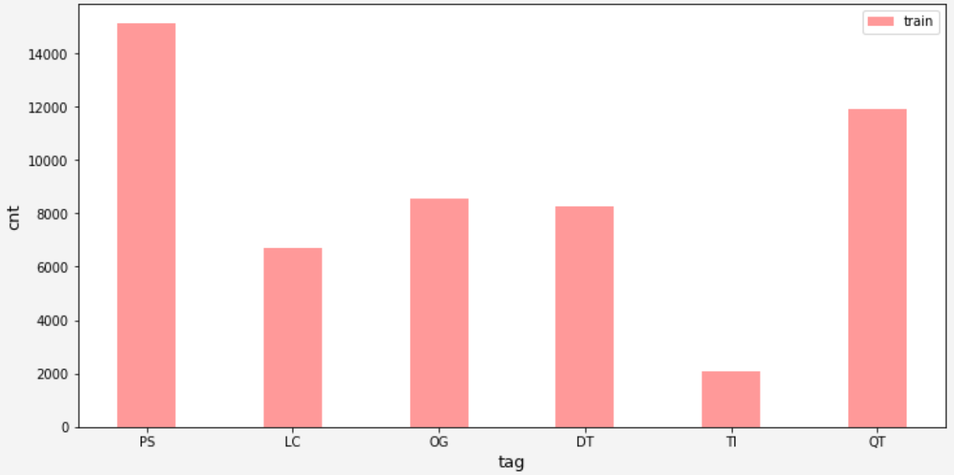
- **text** : 제공 받은 raw text 데이터
- **preprocessed_text** : NER tag, NER 구분자 (‘<’,’>’) , ‘₩n’ 구분자가 제거된 text 데이터
- **entities** : 각 text에 포함된 entity 리스트
- **tags** : 각 text에 포함된 entity가 해당되는 tag 리스트 (PS, LC, OG, DT, TI, QT)
- **cnt** : 각 text에 포함된 entity의 개수

	text	preprocessed_text	entities	tags	cnt
0	아름다운 <첫걸음:QT> . 과정없이 자라는 나무는 없어요~\n	아름다운 첫걸음 . 과정없이 자라는 나무는 없어요~	[첫걸음]	[QT]	1
1	TV상영종료후 <1년뒤:DT>에 극장판이 나올정도의 초 대작.TV편의 설정오류 수정...	TV상영종료후 1년뒤에 극장판이 나올정도의 초 대작.TV편의 설정오류 수정과 새로운...	[1년뒤]	[DT]	1
2	지루할 틈없이 웃음포인트들이 있고, <4년전:DT> 영화지만 지금:봐도 재미있네요\n	지루할 틈없이 웃음포인트들이 있고, 4년전 영화지만 지금:봐도 재미있네요	[4년전]	[DT]	1
3	<고3:QT>때 개봉날 극장서 봤는데 당시 엄청 감동이였음.같이 본 여자애들 울고 ...	고3때 개봉날 극장서 봤는데 당시 엄청 감동이였음. 같이 본 여자애들 울고 그랬어요.	[고3]	[QT]	1
4	눈물 펑펑 역시 <병만:PS>삼촌은 기대를 져버리지 않았슴다!\n	눈물 펑펑 역시 병만삼촌은 기대를 져버리지 않았슴다!	[병만]	[PS]	1

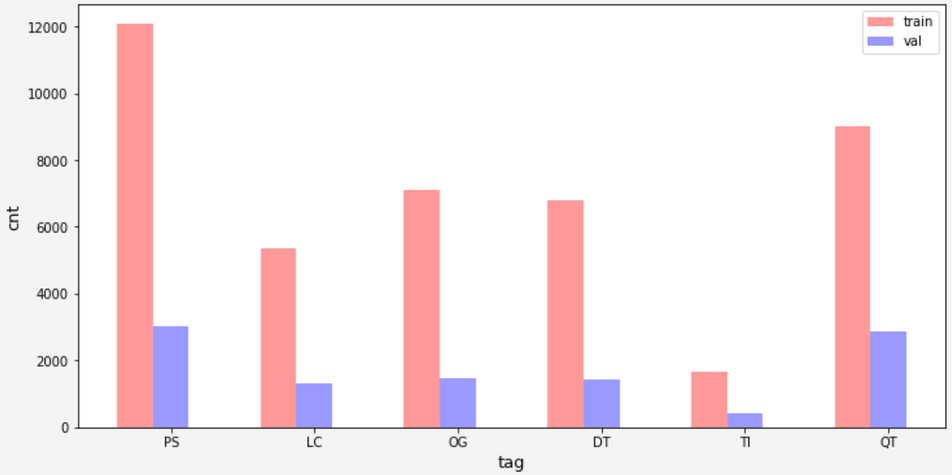
❖ 학습 데이터 - 검증 데이터 분리 시 고려한 점

1) 데이터셋 별 Tag 분포

Tag	Train Tag 개수	Val Tag 개수	Train Tag 비율	Val Tag 비율
PS	12091	3023	0.2875	0.2874
LS	5360	1341	0.1274	0.1275
OG	6839	1710	0.1626	0.1626
DT	6596	1650	0.1568	0.1569
TI	1649	413	0.0392	0.0392
QT	9512	2378	0.2262	0.2261



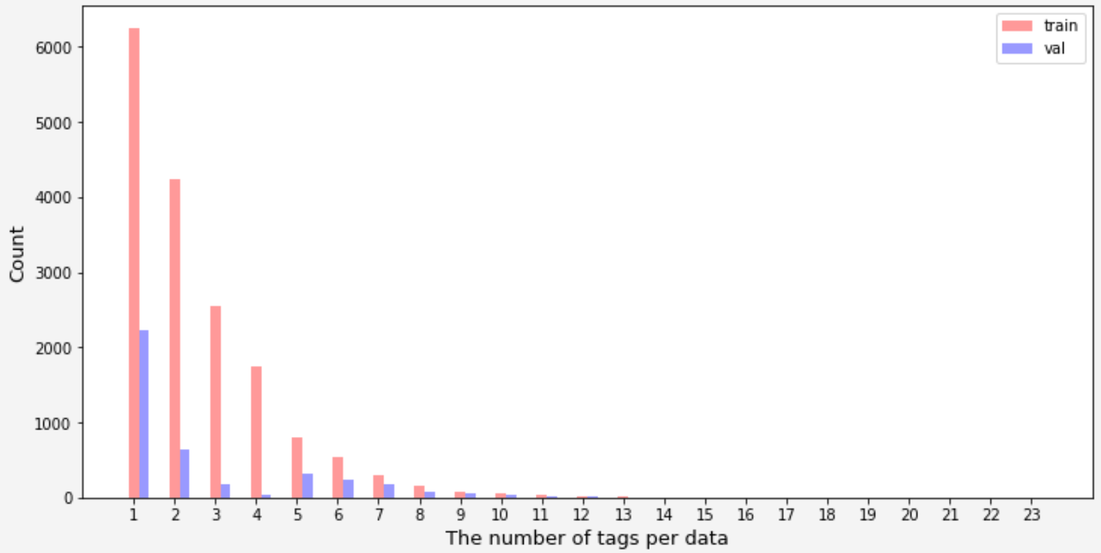
[그래프6] 기존 Train 데이터의 Tag 분포



[그래프7] Train / Val 분리 후 Tag 분포

❖ 학습 데이터 - 검증 데이터 분리 시 고려한 점

2) 데이터셋 별 Tag 개수 분포



# of Tags / Dataset	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Train	7345	4668	2654	1760	976	648	381	188	110	60	35	22	9	6	3	5	0	3	1	1	0	0	1
Val	1118	204	78	37	141	129	92	46	34	21	10	4	2	4	3	0	0	0	1	2	0	0	0

03 모델 학습 방식

| Text-to-text Multi-task Learning

kt GenieLabs Dev-Challenge 2022

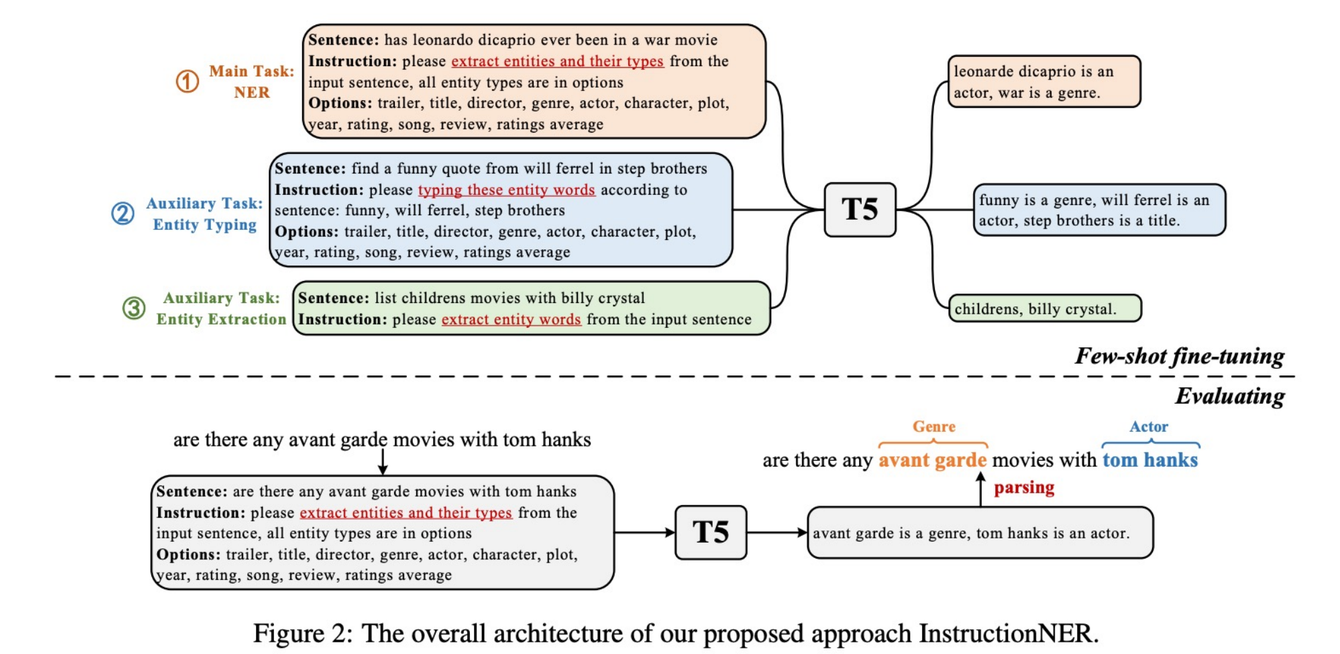


Figure 2: The overall architecture of our proposed approach InstructionNER.

InstructionNER: A Multi-Task Instruction-Based Generative Frameworkfor Few-shot NER (Wang et al., arXiv 22)

- T5의 대표적인 특징인 **Text-to-Text** 학습 방식과 NER 외에 2가지 Auxiliary Task를 동시에 학습하는 **Multi-task Learning** 방식으로 NER Task에서 높은 성능을 달성한 InstructionNER(Wang et al., arXiv 22)을 참고하여 한국어 데이터셋에 맞게 Input-Output을 수정했으며, 논문에서 제안한 3가지 Task로 학습을 진행할 예정
  - **Main Task : Named Entity Recognition**
  - **Auxiliary Task**
    1. **Entity Typing** : 문장과 문장에 존재하는 Entity가 주어진 상태에서 Entity Type을 예측하는 Task
    2. **Entity Extraction** : 문장이 주어졌을 때 문장에 존재하는 Entity를 찾아내는 Task
- Input은 입력 문장과 각 Task별 Instruction, 가능한 Entity Type으로 구성되며, Output은 각 Task에 대한 정답이 문장 형태로 구성됨

03 모델 학습 방식

| 입출력 설계

Raw Text : <중국:LC> 배우 <탕웨이:PS>가 팬 사인회에 모습을 드러냈다.

Main Task : NER

- **Input :** Sentence: 중국 배우 탕웨이가 팬 사인회에 모습을 드러냈다. Instruction: Input Sentence에서 찾을 수 있는 모든 Entity 및 그들의 Entity type을 출력하세요. 가능한 Entity type은 다음과 같습니다: 사람, 위치, 기관, 날짜, 시간, 수량
- **Output** : 중국은 위치이고, 탕웨이는 사람이다.

Auxiliary Task 1 : Entity Typing

- **Input :** Sentence: 중국 배우 탕웨이가 팬 사인회에 모습을 드러냈다. Instruction: Input Sentence에서 <중국, 탕웨이>에 해당하는 Entity 단어들의 Entity type을 출력하세요. 가능한 Entity type은 다음과 같습니다: 사람, 위치, 기관, 날짜, 시간, 수량
- **Output** : 중국은 위치이고, 탕웨이는 사람이다.

Auxiliary Task 2 : Entity Extraction

- **Input** : Sentence: 중국 배우 탕웨이가 팬 사인회에 모습을 드러냈다. Instruction: Input Sentence에서 Entity에 해당하는 단어를 모두 출력하세요.
- **Output** : 중국, 탕웨이.

03 모델 학습 방식

| Part-of-Speech Embedding

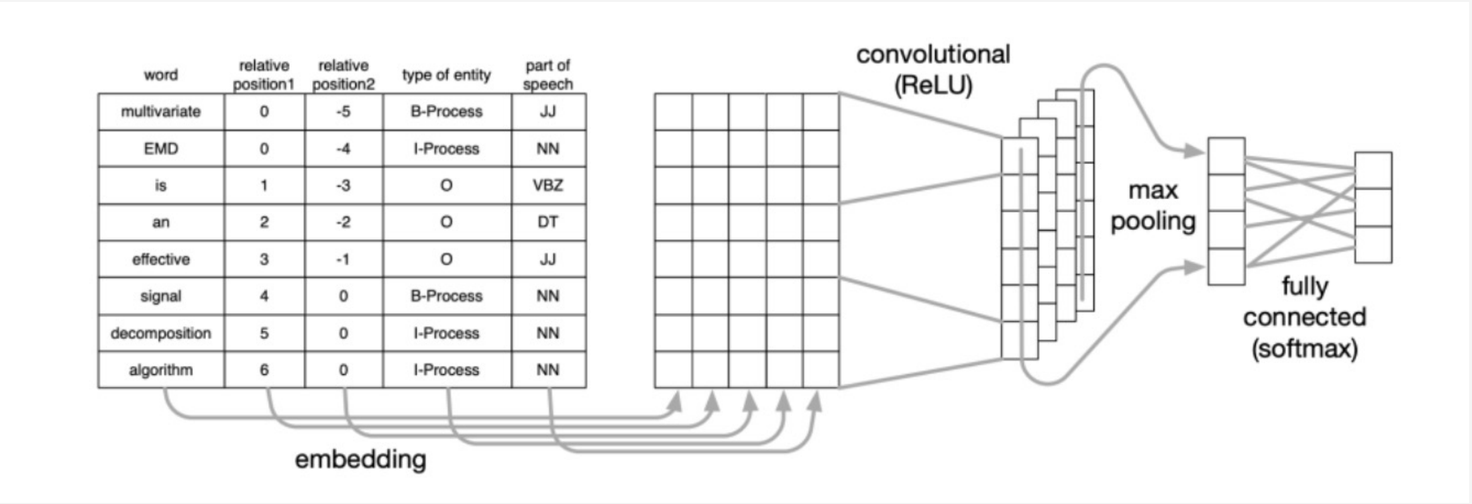
kt GenieLabs Dev-Challenge 2022

- Relation Extraction 분야의 논문에서 품사 태그 정보를 임베딩에 추가하는 방식으로 성능 향상을 보인 점에서 착안함
- POS Embedding을 구하기 위해 활용할 Mecab 라이브러리에 총 43개의 품사 태그가 있으므로 모든 품사 태그를 활용하기보단 2가지 가설을 통해 확인된 Entity별 차이가 뚜렷한 10가지의 품사만 활용할 예정
  - 가설 1 : Named Entity의 Tag에 따라 자주 속하는 명사가 다르다.

    → POS Embedding 활용할 명사 태그 4가지 : 일반 명사, 고유 명사, 단위 명사, 수사
  - 가설 2 : Named Entity의 Tag에 따라 함께 자주 사용되는 조사의 종류가 다르다.

    → POS Embedding 활용할 조사 태그 6가지 : 보조사, 주격 조사, 부사격 조사, 관형격 조사, 목적격 조사, 접속 조사

word	relative position1	relative position2	type of entity	part of speech
multivariate	0	-5	B-Process	JJ
EMD	0	-4	I-Process	NN
is	1	-3	O	VBZ
an	2	-2	O	DT
effective	3	-1	O	JJ
signal	4	0	B-Process	NN
decomposition	5	0	I-Process	NN
algorithm	6	0	I-Process	NN



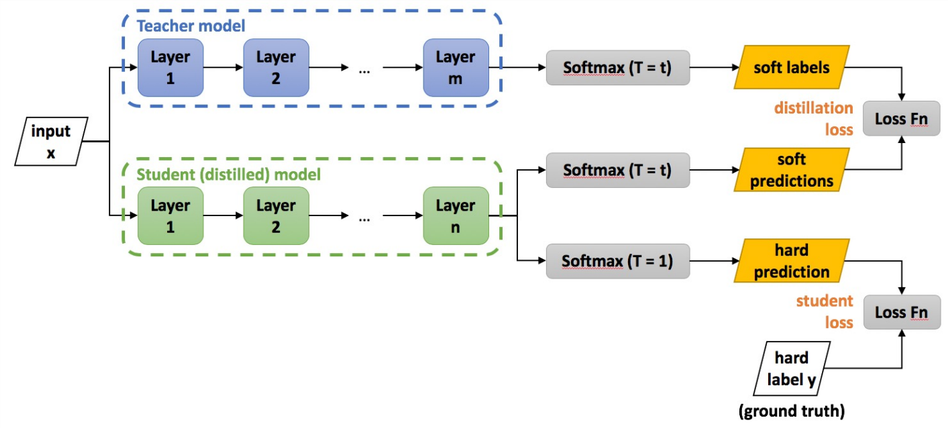
MIT at SemEval-2017 Task 10: Relation Extraction with Convolutional Neural Networks (Lee et al., SEMEVAL 17)

03 모델 학습 방식

| Model Lightweight

kt GenieLabs Dev-Challenge 2022

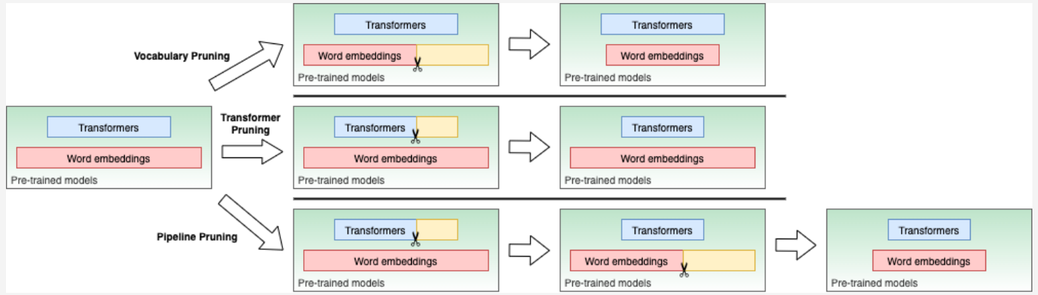
❖ Knowledge Distillation



https://intellabs.github.io/distiller/knowledge_distillation.html

- 성능이 좋고 무거운 모델을 가벼운 모델로 지식 증류하는 방식
  - Teacher 모델 : ke-t5 large/base 또는 KT AI ULM
  - Student 모델 : ke-t5 small
- TextBrewer(Yang et al., ACL 20 Demo Track)에서 공개한 Knowledge Distillation 오픈소스 라이브러리 활용 예정

❖ Vocab Pruning



TextPruner: A Model Pruning Toolkit for Pre-Trained Language Models (ACL 22 Demo)

- 다운스트림 태스크에 잘 등장하지 않는 단어를 사전에 vocab에서 제거하여 모델의 크기를 줄이는 방법
- TextPruner(Yang et al., ACL 22 Demo Track)에서 공개한 오픈소스 라이브러리를 활용해 한국어 T5 모델인 ke-t5 small 모델에 적용했을 때, 76M개의 파라미터에서 63M개의 파라미터로 약 1M개의 파라미터가 감소하는 점을 사전 확인함

LAYER NAME	#PARAMS
--model(partially shared):	76,895,616
--shared	
--weight:	32,833,536
--encoder(shared):	18,883,264
--embed_tokens(shared):	0
--block:	18,882,752
--final_layer_norm:	512
--decoder(shared):	25,178,816
--embed_tokens(shared):	0
--block:	25,178,304
--final_layer_norm:	512
--lm_head(shared)	
--weight(shared):	0

LAYER NAME	#PARAMS
--model(partially shared):	63,144,320
--shared	
--weight:	19,082,240
--encoder(shared):	18,883,264
--embed_tokens(shared):	0
--block:	18,882,752
--final_layer_norm:	512
--decoder(shared):	25,178,816
--embed_tokens(shared):	0
--block:	25,178,304
--final_layer_norm:	512
--lm_head(shared)	
--weight(shared):	0

감사합니다