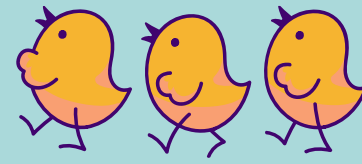


LangCon 2023



# 특정 도메인에 맞는 언어모델은 어떻게 만들까?

BHSN 박장원

## 박장원

([monologg](#)로 더 많이 아시는 거 같아요 😥)

• '한국어 언어모델'에 관심이 많습니다!

- KoELECTRA, KLUE RoBERTa 등

• **BHSN** 에서 Contract AI를 개발하고 있습니다.

- <https://www.bhsn.ai/>

Overview Repositories 31 Projects Packages Stars 1.2k

mono1ogg / README.md

**Things I do**

- NLP Engineer, contributing on Korean NLP with Open Source!

**Find me at**

LinkedIn Gmail Tech blog

**Jangwon Park**  
monologg

Follow

692 followers · 189 following

@bhsn-ai  
Seoul, Republic of Korea  
adieuju@gmail.com  
monologg.kr/about

**Highlights**  
Developer Program Member  
PRO

**Organizations**  
BHSN  
Block or Report

**Pinned**

- seanie12/mrqa** (Public)  
Code for EMNLP-IJCNLP 2019 MRQA Workshop Paper: "Domain-agnostic Question-Answering with Adversarial Training"  
Python 38 11
- KoELECTRA** (Public)  
Pretrained ELECTRA Model for Korean  
Python 523 138
- KoBigBird** (Public)  
Pretrained BigBird Model for Korean (up to 4096 tokens)  
Python 191 19
- ko\_lm\_dataformat** (Public)  
A utility for storing and reading files for Korean LM training  
Python 33 2
- transformers-android-demo** (Public)  
Transformers android examples (Tensorflow Lite & Pytorch Mobile)  
Java 62 14
- NER-Multimodal-pytorch** (Public)  
Pytorch Implementation of "Adaptive Co-attention Network for Named Entity Recognition in Tweets" (AAAI 2018)  
Python 50 11

2,057 contributions in the last year

2023  
2022  
2021  
2020  
...

Learn how we count contributions Less More

1. Intro
2. 도메인 특화 언어모델
3. Need to Consider (Data, Tokenizer, Difficulty of Task)
4. 저도...만들어 보고 싶어요!
5. 도메인 특화 언어모델이 도움이 되나요?

# 1. Intro



**'거대 모델을 대상으로 한 자연어처리 동향 이해' 로  
하러 해요!**

**후후 접수완료~ 걱정마세요~~**



**3 HOURS**

**LATER...**

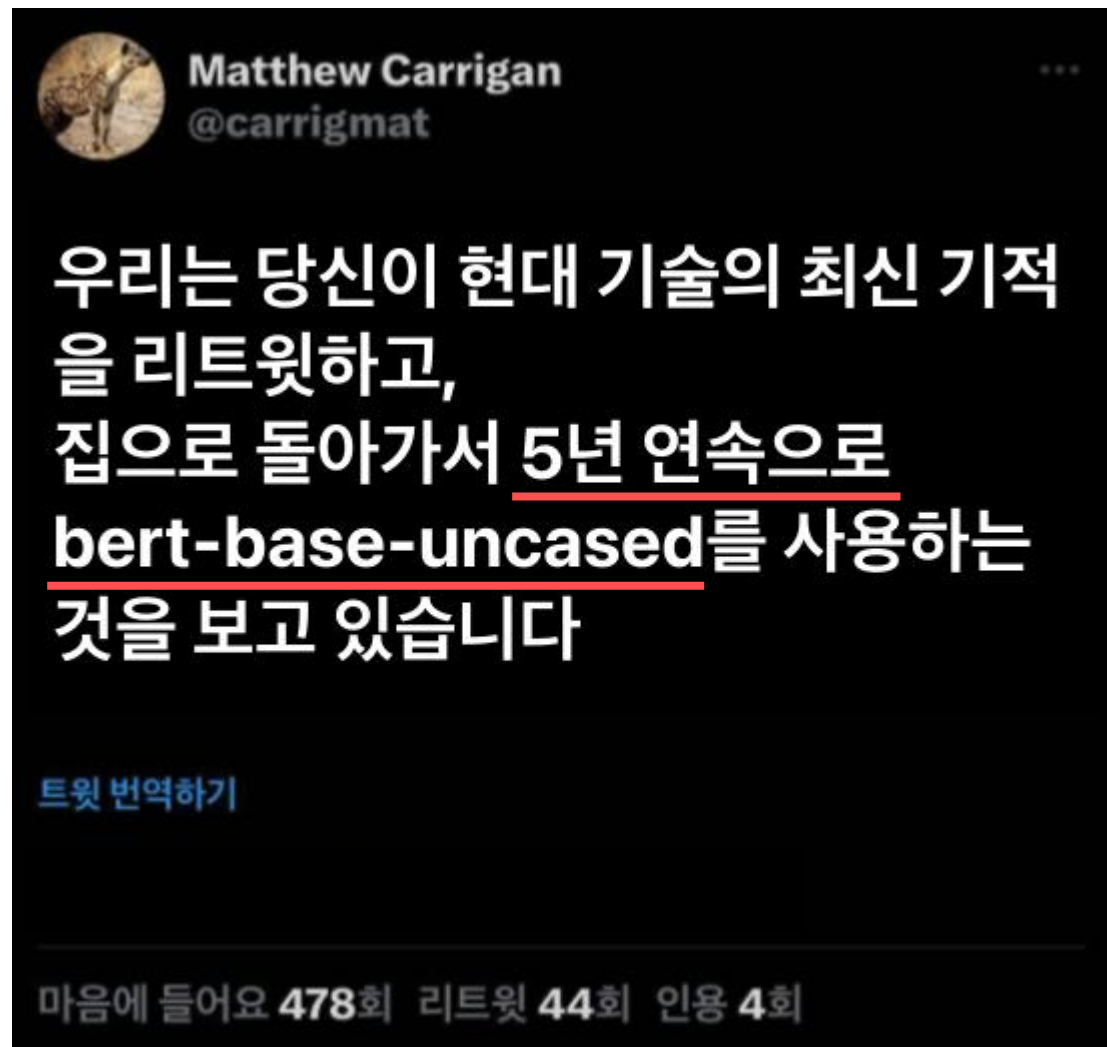


Matthew Carrigan  
@carrigmat

this is [@huggingface](#), we see you out there retweeting the latest state of the art miracle of modern technology and then going home and using bert-base-uncased for the fifth year in a row

트윗 번역하기

마음에 들어요 478회 리트윗 44회 인용 4회



Matthew Carrigan  
@carrigmat

우리는 당신이 현대 기술의 최신 기적을 리트윗하고,  
집으로 돌아가서 5년 연속으로 bert-base-uncased를 사용하는 것을 보고 있습니다

트윗 번역하기

마음에 들어요 478회 리트윗 44회 인용 4회

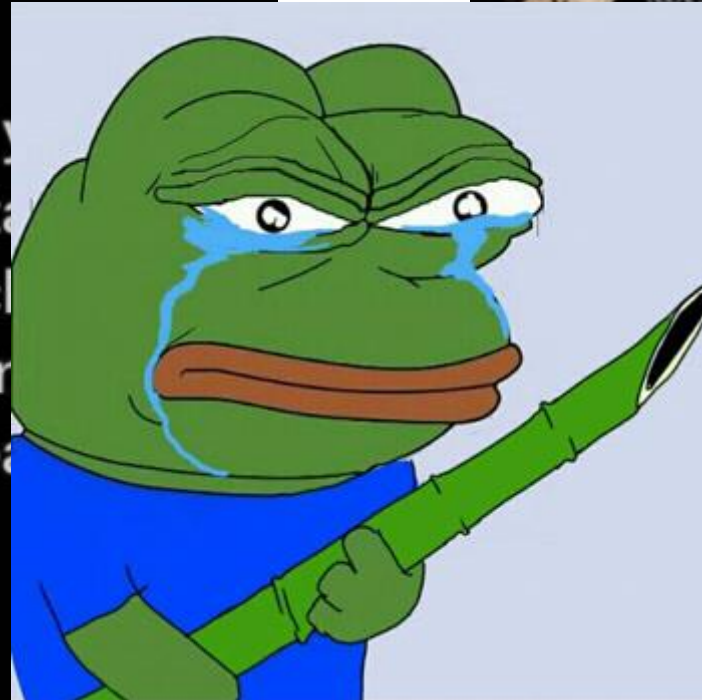
 **Matthew Carrigan**  
@carrigmat

this is [@huggingface](#), we see  
there retweeting the latest sta  
the art miracle of modern tech  
and then going home and usin  
base-uncased for the fifth year  
row

트윗 번역하기

---

마음에 들어요 478회 리트윗 44회 인용 4회



 **Matthew Carrigan**  
@carrigmat

당신이 현대 기술의 최신 기적  
릿하고,  
돌아가서 5년 연속으로  
base-uncased를 사용하는  
고 있습니다

---

마음에 들어요 478회 리트윗 44회 인용 4회



# OpenAI



**C.h.a.t.G.P.T.**

# Others



**나한테는 BERT도 빅모델이야ㅠ**

**99%의 현업자가**  
**바로 적용해보고 공감할 수 있는 것을**  
**이야기해볼까?**



**특정 도메인에서  
적당한 사이즈의 모델로도  
해결할 수 있는 것이 많다!**



## 2. 도메인 특화 언어모델

**General**  
(리뷰)

액션이랑 모든게 다 재미있는  
몇안되는 영화



Positive

**Domain**  
(계약서)

13.7 불가항력. 어떠한 당사자도 화재, 폭풍, 홍수, 지진, 사고, 전쟁(실제 발생 또는 선포 여부를 불문함), 천재지변, 법률의 규정, 정부기관의 조치 등 **자신의 통제를 벗어난 사유**에 의해 본 계약을 준수하지 못하는 경우 이에 대한 책임을 지지 아니한다. 다만, 해당 당사자는 불가항력이 발생한 경우 및 그러한 불가항력 상태로부터 벗어난 경우 가능한 한 빨리 이를 상대방에게 통지하여야 한다.

Q. Pandemic(COVID-19)를 이유로 계약의 이행을 거절할 수 있나?



**Biomedical**

**Legal**

**Finance**



**BioBERT, SciBERT, LegalBERT,  
FinBERT, BERTweet,  
PubMedBERT...**

Table 28: Statistics of the pretraining corpus.

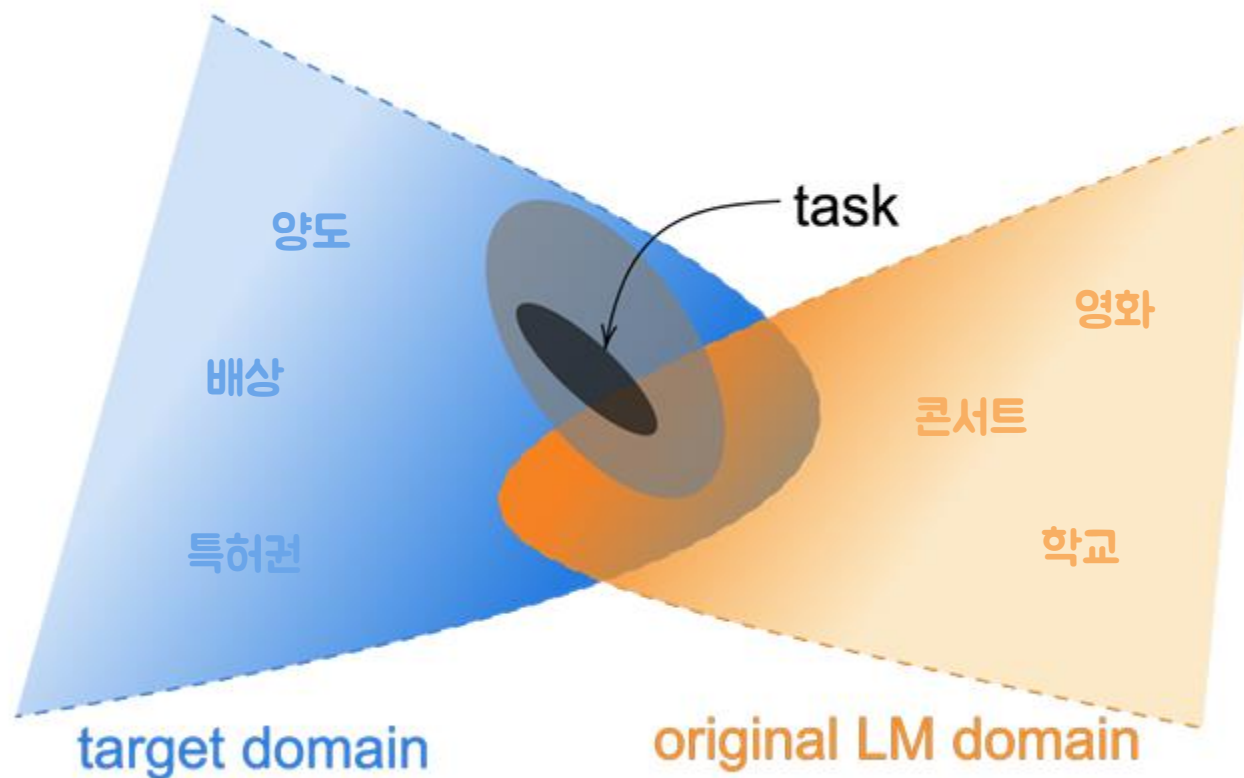
	MODU	CC-100-Kor	NAMUWIKI	NEWSCRAWL	PETITION	Total
# Sentences	167M	103M	14M	183M	5.2M	473M
# Words	1,892,814,395	1,593,887,022	265,203,602	2,716,968,038	50,631,183	6,519,504,240
size (GB)	18.27	15.46	2.52	25.87	0.53	62.65

**Pretraining Corpora** We gather the following five publicly available Korean corpora from diverse sources to cover a broad set of topics and many different styles. We combine these corpora to build the final pretraining corpus of size approximately 62GB. See Table 28 for overall statistics:

- **MODU** : *Modu*<sup>51</sup> Corpus [98] is a collection of Korean corpora distributed by National Institute of Korean Languages.<sup>52</sup> It includes both formal articles (news and books) and colloquial text (dialogues).
- **CC-100-Kor** : CC-100<sup>53</sup> is the large-scale multilingual web crawled corpora by using CC-Net [136]. This is used for training XLM-R [26]. We use the Korean portion from this corpora.
- **NAMUWIKI** : NAMUWIKI is a Korean web-based encyclopedia, similar to Wikipedia, but known to be less formal. Specifically, we download the dump created on March 2nd, 2020.<sup>54</sup>
- **NEWSCRAWL** : NEWSCRAWL consists of 12,800,000 news articles published from 2011 to 2020, collected from a news aggregation platform.
- **PETITION** : Petition is a collection of public petitions posted to the Blue House asking for administrative actions on social issues. We use the articles in the Blue House National Petition<sup>55</sup> published from August 2017 to March 2019.<sup>56</sup>

- News
- Wiki
- Book
- Web Crawl

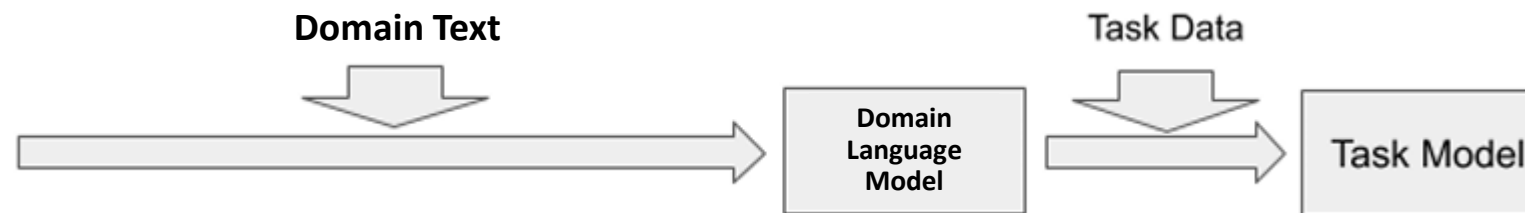




**In-Domain**

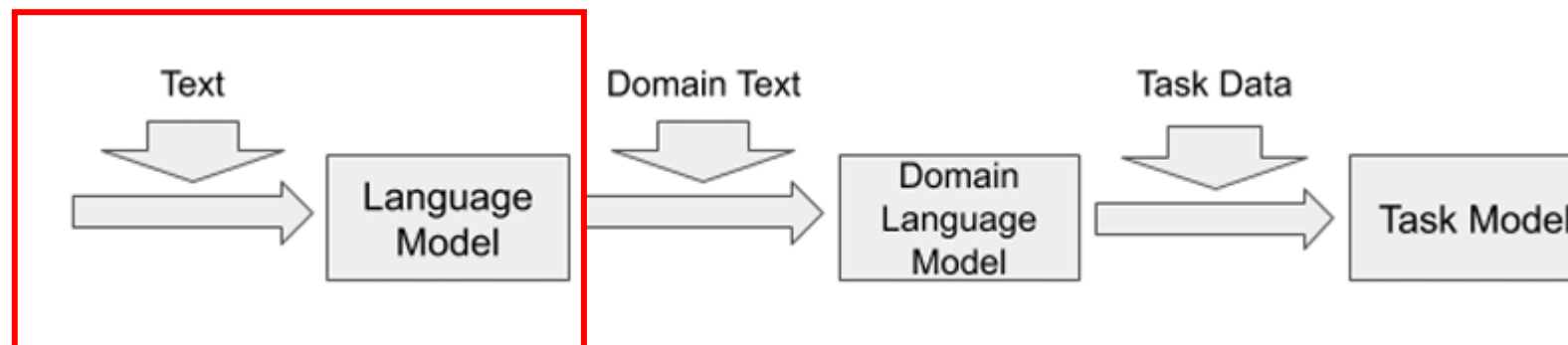
**Out-Domain**

## Pretraining From Scratch



<General LM (e.g. KoELECTRA)>

## Continual Pretraining



도메인 특화 언어모델을 만들 때  
무엇을 고려해야할까?



# 3. Need to Consider

## 3.1. Data

# 1. 그거 진짜 Domain-Specific Corpus는 맞아?



**금융 특화 언어모델을 만들고 싶어요!**

# 1. 그거 진짜 Domain-Specific Corpus는 맞아?



금융 특화 언어모델을 만들고 싶어요!

허허 그렇군요. 말씀치는 준비하셨나요?



# 1. 그거 진짜 Domain-Specific Corpus는 맞아?



금융 특화 언어모델을 만들고 싶어요!

허허 그렇군요. 말씀하시는 준비하셨나요?



옹! 경제 분야 뉴스를 가져왔어요!



# 1. 그거 진짜 Domain-Specific Corpus는 맞아?



금융 특화 언어모델을 만들고 싶어요!

허허 그렇군요. 말씀하시는 준비하셨나요?



옹! 경제 분야 뉴스를 가져왔어요!

(띠용?!)



# 1. 그거 진짜 Domain-Specific Corpus는 맞아?

Table 28: Statistics of the pretraining corpus.

	MODU	CC-100-Kor	NAMUWIKI	NEWSCRAWL	PETITION	Total
# Sentences	167M	103M	14M	183M	5.2M	473M
# Words	1,892,814,395	1,593,887,022	265,203,602	2,716,968,038	50,631,183	6,519,504,240
size (GB)	18.27	15.46	2.52	25.87	0.53	62.65

**Pretraining Corpora** We gather the following five publicly available Korean corpora from diverse sources to cover a broad set of topics and many different styles. We combine these corpora to build the final pretraining corpus of size approximately 62GB. See Table 28 for overall statistics:

- **MODU** : *Modu*<sup>51</sup> Corpus [98] is a collection of Korean corpora distributed by National Institute of Korean Languages.<sup>52</sup> It includes both formal articles (news and books) and colloquial text (dialogues).
- **CC-100-Kor** : CC-100<sup>53</sup> is the large-scale multilingual web crawled corpora by using CC-Net [136]. This is used for training XLM-R [26]. We use the Korean portion from this corpora.
- **NAMUWIKI** : NAMUWIKI is a Korean web-based encyclopedia, similar to Wikipedia, but known to be less formal. Specifically, we download the dump created on March 2nd, 2020.<sup>54</sup>
- **NEWSCRAWL** : NEWSCRAWL consists of 12,800,000 news articles published from 2011 to 2020, collected from a news aggregation platform.
- **PETITION** : Petition is a collection of public petitions posted to the Blue House asking for administrative actions on social issues. We use the articles in the Blue House National Petition<sup>55</sup> published from August 2017 to March 2019.<sup>56</sup>

• News

• Wiki

• Book

• Web Crawl



# 1. 그거 진짜 Domain-Specific Corpus는 맞아?

**N 뉴스** | 연예 | 스포츠 | 날씨 | 프리미엄

언론사별 | **정치** | **경제** | 사회 | 생활/문화 | IT/과학 | 세계 | 랭킹 | 신문보기 | 오피니언 | TV | 팩트체크

02.16(목)

경제  
금융  
증권  
산업/재계  
중기/벤처  
부동산  
글로벌 경제  
생활경제

① 헤드라인 뉴스

13 "폭탄 고지서" 공공요금 속도조절 · 전기가스로 추가인상 늦춘다지만 >

 가스전기로 인상 방침 고수한 정부...중산층 지원 없지만 尹 "속도조...  
정부가 서민부담 최소화를 목표로 폭과 속도를 조절하되 가스전기요금 정상화는 변  
함 없이 추진하겠다는 의지를 재차 분명히 했다. 취약계층에 대한 지원을 보 ...  
뉴스1

'폭탄 고지서'에 공공요금 속도조절...공기업 손실 더 커진다 중앙일보 | 10+

'폭탄 고지서'에 속도조절 "촉선 후 더 큰 폭탄" 우려도 JTBC | 30+

정치 + **경제** + 사회 + 생활/문화 + IT/과학 + 세계

# 1. 그거 진짜 Domain-Specific Corpus는 맞아?

(PT = RoBERTa)

PT	100.0	54.1	34.5	27.3	19.2
News	54.1	100.0	40.0	24.9	17.3
Reviews	34.5	40.0	100.0	18.3	12.7
BioMed	27.3	24.9	18.3	100.0	21.4
CS	19.2	17.3	12.7	21.4	100.0
	PT	News	Reviews	BioMed	CS

Figure 2: Vocabulary overlap (%) between domains. PT denotes a sample from sources similar to ROBERTA's pretraining corpus. Vocabularies for each domain are created by considering the top 10K most frequent words (excluding stopwords) in documents sampled from each domain.

## Vocabulary Overlap

- 총 4개의 도메인
  - News, Reviews, BioMed, Computer Science
- News, Reviews → 유사

# 1. 그거 진짜 Domain-Specific Corpus는 맞아?

Dom.	Task	ROBA.	DAPT	$\neg$ DAPT
BM	CHEMPROT	81.9 <sub>1.0</sub>	<b>84.2</b> <sub>0.2</sub>	79.4 <sub>1.3</sub>
	†RCT	87.2 <sub>0.1</sub>	<b>87.6</b> <sub>0.1</sub>	86.9 <sub>0.1</sub>
CS	ACL-ARC	63.0 <sub>5.8</sub>	<b>75.4</b> <sub>2.5</sub>	66.4 <sub>4.1</sub>
	SCIERC	77.3 <sub>1.9</sub>	<b>80.8</b> <sub>1.5</sub>	79.2 <sub>0.9</sub>
NEWS	HYP.	86.6 <sub>0.9</sub>	<b>88.2</b> <sub>5.9</sub>	76.4 <sub>4.9</sub>
	†AGNEWS	<b>93.9</b> <sub>0.2</sub>	<b>93.9</b> <sub>0.2</sub>	93.5 <sub>0.2</sub>
REV.	†HELPFUL.	65.1 <sub>3.4</sub>	<b>66.5</b> <sub>1.4</sub>	65.1 <sub>2.8</sub>
	†IMDB	95.0 <sub>0.2</sub>	<b>95.4</b> <sub>0.2</sub>	94.1 <sub>0.4</sub>

점수차가 크지 않음! 🙄🙄

Table 3: Comparison of ROBERTA (ROBA.) and DAPT to adaptation to an *irrelevant* domain ( $\neg$ DAPT). Reported results are test macro- $F_1$ , except for CHEMPROT and RCT, for which we report micro- $F_1$ , following Beltagy et al. (2019). We report averages across five random seeds, with standard deviations as subscripts. † indicates high-resource settings. Best task performance is boldfaced. See §3.3 for our choice of irrelevant domains.

## 2. Corpus는 어느 정도의 양이 적당한가?

일단 많으면 많을수록  
무조건 좋습니다!



## 2. Corpus는 어느 정도의 양이 적당한가?

**Corpus의 최소 요구량은 정답이 없습니다.**  
**다만 최근(2022.12)에 이런 논문은 있었어요.**



## 2. Corpus는 어느 정도의 양이 적당한가?



Dataset	BERT	PubMedBERT	4GB	8GB	12GB
NCBI-disease	84.3	87.8	87.7	87.9	88.0
HoC	79.0	82.3	81.1	82.5	81.4
PubMedQA	54.4	55.8	54.9	53.4	55.2

Table 2: Performance comparison of pre-trained language models. The models are evaluated on the tasks using the same fine-tuning process. All of our experimental models are pre-trained for 67K steps.



# 3. 데이터의 총 사이즈를 직접 계산해보세요

농민신문

## '임차 농지' 공익직불금 사각지대 여전

입력 2023.02.15 오전 5:02 | 기사원문

하지혜 기자

2 4

PICK 0 0 0 0 0

계약서 없는 농가 상당수...농식품부 예외 사항 제시  
농업경영체 등록정보엔 등재 못해 사실상 '신청 불가'



이미지투데이

"밭된 땅에서 농사짓고 세금까지 낸 세월이 얼마데, 임대차계약서 없이는 직불금을 못 준다니 포기했죠, 뭐."

전북의 시골에 농가 김모씨는 올해도 1322㎡(400평) 규모의 임차 농지에 대해 기본형 공익직불금을 신청하지 못했다. 직불금 신청에 필요한 임대차계약서가 없어서다. 그는 오래전 농지 소유주가 갑자기 사망한 후 땅을 물려받은 자녀들과 연락이 닿지 않아 임대차계약서를 쓰지 못했다. 그러다 2021년부터 임대차계약서를 제출해야 기본형 공익직불금을 신청할 수 있게 되면서 임차 농지에 대한 직불금을 아예 포기했다.

김씨는 "주위에 중중이나 소유주가 여러명인 땅은 임대차계약서를 쓰지 못한 농가가 적지 않다"며 "예전엔 임차 농지의 경작사실확인서를 신청 서류로 내면 직불금을 받을 수 있었는데, 이젠 똑같이 농사를 지어도 직불금을 받지 못하니 씁쓸하다"고 토로했다.

2023년 기본형 공익직불금 신청이 시작된 가운데 올해부터는 공익직불제의 사각지대가 해소될 것이란 기대감이 높다. 지난해 법 개정으로 올해부터 '2017~2019년 중 직불금을 받은 이력이 없는 농지'도 기본형 공익직불금을 신청할 수 있게 됐기 때문이다. 그러나 전체 농가의 절반에 달하는 임차농가들 사이에선 아직도 제도에 구멍이 있다는 의견이 제기된다.

문제는 임대차계약서다. 농지 소유자의 사망이나 행방불명, 해외 거주 등으로 소유권자가 불확실한 땅은 현실적으로 임대차계약을 맺기 어렵다. 소유관계가 복잡한 중중 땅이나 공동 소유 농지도 마찬가지다.

이런 문제가 제기되면서 농림축산식품부는 2021년부터 기본형 공익직불사업 시행지침에 나름의 해결책을 제시해왔다. 소유권자가 불확실한 농지는 경작자의 재산세납부서나 재산세납부자와의 계약서 등을 제출하면 임대차계약서를 갈음할 수 있게 해준 것이다. 중중 소유 농지도 실경 작자에게 땅을 임대한다는 내용이 담긴 중중 회의록을 임대차계약서 대신 제출할 수 있도록 했다.

이 지침대로라면 김씨 같은 임차농들도 직불금을 신청할 수 있지만 실제로 그렇지 않다. 사실상 임대차계약서가 없는 임차 농지는 농업경영체 등록정보에 등재할 수 없기 때문이다. 기본형 공익직불금 지급 대상은 '농업경영체 육성 및 지원에 관한 법률'에 따라 농업경영체 등록정보에 등록된 농지로 정해져 있다. 농업경영체 등록정보에 임차 농지를 등재하려면 임대차 현황 내용이 담긴 농지대장을 제출해야 한다. 그런데 이 농지대장에 임차 농지를 신고하려면 결국 임대차 계약서가 필요하다.

이에 대해 농업경영체 등록을 담당하는 국립농산품질관리원은 임대차계약서를 통해 '농지법'상 적법한 임차 농지만 농업경영체 등록정보에 등재하는 건 당연하다는 입장이다.

현행 '농지법'은 징집·질병 등 불가피한 상황을 제외한 사적인 임대차를 원칙적으로 금지한다. 1996년 1월1일 이후 취득한 농지는 한국농어촌공사를 통해 임대차계약을 맺어야 한다.

농관원 관계자는 "지난해 농식품부 국정감사에서조차 불법 임차 농지의 직불금 수급문제가 도마 위에 올랐다"면서 "불법 임대차와 직불금 부정수급을 막기 위해선 임대차계약서를 바탕으로 적법성을 따져야 한다"고 말했다.

현장에선 합법적인 임대차계약을 맺고 싶어도 현실적으로 힘들다는 목소리가 나온다.

한 면사무소의 직불금 담당자는 "농지은행을 통해 임대차계약을 맺는 데 번거로움을 느끼거나 8년 이상 자경 때 양도소득세를 감면받는 조항 때문에 임대차계약서를 써주지 않는 농지 소유자들이 여전히 있다"며 "불가피하게 불법 임차농으로 몰린 농민들은 억울할 수밖에 없다"고 지적했다.

## • UTF-8 기준

• 한글, 한자: 3 Byte

• 영어, 숫자, 공백: 1 Byte

## • 좌측의 뉴스는

# 4293Byte = 0.004 MB

## • 1GB를 모으려면 256000개의 뉴스가 필요함

• 그러나 모든 뉴스가 예시만큼 길지 않습니다ㅠ

## 2. Corpus는 어느 정도의 양이 적당한가?

**Domain-Specific Corpus를 모을 것이라면  
꼭 미리 계산하고 **각을 재보세요!****  
(생각보다 많이들 안 해보십니다ㅠ)



## 3.2. Tokenizer



데이터를 도저히 많이 못 모으겠네요 ㅠ.ㅠ

그러면 **Tokenizer**라도 고쳐볼까요?



# 1. 일단 UNK이 발생하는 것 먼저 보자!

- OOV (Out Of Vocabulary), UNK (Unknown Token)
- e.g. 나는 오늘 **똥교**에 갔다  
-> ['나', '##는', '오늘', '**[UNK]**', '갔', '##다']

# 1. 일단 UNK이 발생하는 것 먼저 보자!

```
from transformers import BertTokenizer
```

```
tokenizer = BertTokenizer.from_pretrained("klue/bert-base")
```

✓ 3.9s

Python

```
tokens = tokenizer.tokenize("상법의일부규정의시행에관한규정에 의하면 아래와 같습니다")  
print(tokens)
```

✓ 0.0s

Python

```
['상법', '##의', '##일', '##부', '##규', '##정', '##의', '##시', '##행', '##에', '##관', '##한', '##규', '##정', '##에', '의하', '##면', '아래', '##와', '같', '##습', '##니다']
```

```
tokens = tokenizer.tokenize("상법의일부규정의시행에관한규정에 의하면 아래와 같습니다")  
print(tokens)
```

✓ 0.0s

Python

```
[' [UNK]', '의하', '##면', '아래', '##와', '같', '##습', '##니다']
```



# 1. 일단 UNK이 발생하는 것 먼저 보자!

```
class WordpieceTokenizer(object):
    """Runs WordPiece tokenization."""

    def __init__(self, vocab, unk_token, max_input_chars_per_word=100):
        self.vocab = vocab
        self.unk_token = unk_token
        self.max_input_chars_per_word = max_input_chars_per_word

    def tokenize(self, text):
        """
        Tokenizes a piece of text into its word pieces. This uses a greedy longest-match-first algorithm to perform
        tokenization using the given vocabulary.

        For example, `input = "unaffable"` wil return as output `["un", "##aff", "##able"]`.

        Args:
            text: A single token or whitespace separated tokens. This should have
                already been passed through *BasicTokenizer*.

        Returns:
            A list of wordpiece tokens.
        """
```

# 1. 일단 UNK이 발생하는 것 먼저 보자!

```
output_tokens = []
for token in whitespace_tokenize(text):
    chars = list(token)
    if len(chars) > self.max_input_chars_per_word:
        output_tokens.append(self.unk_token)
        continue

    is_bad = False
    start = 0
    sub_tokens = []
    while start < len(chars):
        end = len(chars)
        cur_substr = None
        while start < end:
            substr = "".join(chars[start:end])
            if start > 0:
                substr = "##" + substr
            if substr in self.vocab:
                cur_substr = substr
                break
            end -= 1
        if cur_substr is None:
            is_bad = True
            break
        sub_tokens.append(cur_substr)
        start = end

    if is_bad:
        output_tokens.append(self.unk_token)
    else:
        output_tokens.extend(sub_tokens)
return output_tokens
```

최소 단위인 Character에서  
일치하는 게 없으면  
전부 [UNK] 처리가 됨ㄸㄸ





1. KoELECTRA Vocab을 그대로 쓴다
2. Vocab를 아예 새로 만든다 (새로운 도메인의 Corpus를 이용하여)
3. 기존 KoELECTRA Vocab에 새로운 단어를 추가

Continual Pretraining이 가능!

1. KoELECTRA Vocab을 그대로 쓴다
2. Vocab를 아예 새로 만든다 (새로운 도메인의 Corpus를 이용하여)
3. 기존 KoELECTRA Vocab에 새로운 단어를 추가

1. KoELECTRA Vocab을 그대로 쓴다

Pretraining From Scratch가 바람직함

2. Vocab를 아예 새로 만든다 (새로운 도메인의 Corpus를 이용하여)

3. 기존 KoELECTRA Vocab에 새로운 단어를 추가

1. KoELECTRA Vocab을 그대로 쓴다
2. Vocab를 아예 새로 만든다 (새로운 도메인의 Corpus를 이용하여)
3. 기존 KoELECTRA Vocab에 새로운 단어를 추가

### Vocabulary Expansion

-> 기존 35,000개인 KoELECTRA Vocab에 단어를 더 추가

그러면 **Vocab Size**는  
어디까지 키우는 게 제일 좋을까요?



# 3. Vocabulary Expansion

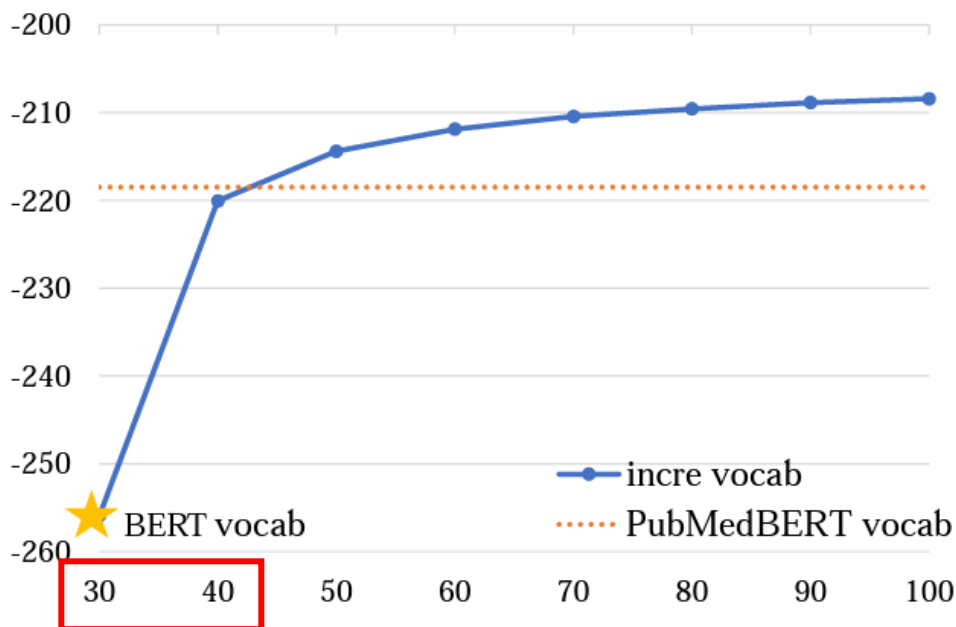


Figure 4: The  $P(D)$  of different vocab sizes under biomedical domain. We use the BERT’s vocabulary as the 30k vocabulary without vocabulary expanding. The PubMedBERT vocabulary is also 30k.

Given a domain-specific corpus  $D$ , the occurrence probability of corpus  $D$  is formulated as:

$$P(D) = \sum_x^{|D|} \log(P(\mathbf{x})), \quad (3)$$

where  $\mathbf{x}$  represents tokenized sentence in corpus  $D$ .

- Occurrence Probability라는 심플한 방법을 사용
  - 각 token의 출현빈도의 곱 -> 로그 합
- 30K -> 40K 로 갈 때 가장 효과적이고, 그 이후부터는 점차 줄어듦
  - 최적의 사이즈는 본인의 기준에 맞게 판단하면 됨!

Vocab Expansion만 하고  
Continual Pretraining을 하지 않아도  
성능이 오른 연구 결과가 있다.



# 3. Vocabulary Expansion

BioMed	CS	News	Reviews
[inc, ub, ated] → incubated	[The, orem] → Theorem	[t, uesday] → tuesday	[it, 's] → it's
[trans, fect] → transfect	[L, em, ma] → Lemma	[ob, ama] → obama	[that, 's] → that's
[ph, osp, ory] → phosphory	[vert, ices] → vertices	[re, uters] → reuters	[sh, oes] → shoes
[mi, R] → miR	[E, q] → Eq	[iph, one] → iphone	[doesn, 't] → doesn't
[st, aining] → staining	[cl, ust, ering] → clustering	[ny, se] → nyse	[didn, 't] → didn't
[ap, opt, osis] → apoptosis	[H, ence] → Hence	[get, ty] → getty	[can, 't] → can't
[G, FP] → GFP	[Seg, mentation] → Segmentation	[inst, agram] → instagram	[I, 've] → I've
[pl, asm] → plasm	[class, ifier] → classifier	[bre, xit] → brexit	[b, ought] → bought
[ass, ays] → assays	[Ga, ussian] → Gaussian	[nas, daq] → nasdaq	[you, 'll] → you'll
[ph, osp, ory, lation] → phosphorylation	[p, olyn] → polyn	[ce, o] → ceo	[kind, le] → kindle

Table 4: Samples of token sequences with large JSD between base and domain corpora sequence distributions; all of these sequences were added during AT to the Roberta-Base tokenizer.

**'incubated' 란 단어가 추가되면**

**-> 'inc', 'ub', 'ated' 의 embedding의 평균으로 초기화!**



# 3. Vocabulary Expansion

AT = Adaptive Tokenization

Domain	Task	RoBERTa	DAPT	TAPT	DAPT + TAPT	AT (Mean)	AT (Proj)	State-of-the-art (in 2020)
BioMed*	ChemProt	81.9 <sub>1.0</sub>	<u>84.2<sub>0.2</sub></u>	82.6 <sub>0.4</sub>	<b>84.4<sub>0.4</sub></b>	83.6 <sub>0.4</sub>	83.1 <sub>0.3</sub>	84.6
	RCT	87.2 <sub>0.1</sub>	<u>87.6<sub>0.1</sub></u>	87.7 <sub>0.1</sub>	<b>87.8<sub>0.1</sub></b>	87.5 <sub>0.4</sub>	<u>87.6<sub>0.3</sub></u>	92.9
CS*	ACL-ARC	63.0 <sub>5.8</sub>	<u>75.4<sub>2.5</sub></u>	67.4 <sub>1.8</sub>	<b>75.6<sub>3.8</sub></b>	70.1 <sub>2.0</sub>	68.9 <sub>1.6</sub>	71.0
	SciERC	77.3 <sub>1.9</sub>	<u>80.8<sub>1.5</sub></u>	79.3 <sub>1.5</sub>	81.3 <sub>1.8</sub>	<b>81.4<sub>0.4</sub></b>	81.2 <sub>1.2</sub>	81.8
News	HyperPartisan	86.6 <sub>0.9</sub>	88.2 <sub>5.9</sub>	90.4 <sub>5.2</sub>	90.0 <sub>6.6</sub>	<b>93.1<sub>4.2</sub></b>	91.6 <sub>5.5</sub>	94.8
Reviews	IMDB	95.0 <sub>0.2</sub>	95.4 <sub>0.1</sub>	95.5 <sub>0.1</sub>	<b>95.6<sub>0.1</sub></b>	95.4 <sub>0.1</sub>	<u>95.5<sub>0.1</sub></u>	96.2

Table 2: Results of different adaptive pretraining methods compared to the baseline RoBERTa. AT with mean subword and projective initializations are denoted as AT (Mean) and AT (Proj) respectively. Stddevs are from 5 seeds. Results for DAPT, TAPT, DAPT+TAPT, and state-of-the-arts are quoted from [Gururangan et al. \(2020\)](#). The highest non-state-of-the-art result is bolded, since the state-of-the-art functions as a performance ceiling, leveraging both domain-specific pretraining and an adapted tokenizer. The best of the three approaches which utilize only source and domain domain data before fine-tuning (i.e., DAPT and AT) is underlined. \*Due to restrictions on accessible papers in S2ORC, The BioMed and CS pretraining corpora used were respectively 33% and 74% smaller than the versions in [Gururangan et al. \(2020\)](#). Note that state-of-the-art numbers are current at the time of [Gururangan et al. \(2020\)](#), and are from the following works: ChemProt: S2ORC-BERT [Lo et al. \(2020\)](#), RCT: Sequential Sentence Classification [Cohan et al. \(2019\)](#), ACL-ARC: SciBert [Beltagy et al. \(2019\)](#), SciERC: S2ORC-BERT [Lo et al. \(2020\)](#), HyperPartisan: Longformer [Beltagy et al. \(2020\)](#), IMDB: XLNet Large [Yang et al. \(2019\)](#).

Tokenizer를 평가하는 방법은 다양하지만  
가장 쉬운 시작점을 소개하려 합니다



## 1) Subword Fertility

- Average Number of Subwords produced per words

- 단어마다 평균 몇 개의 subword로 쪼개지는가

e.g.

- '도시계획사업시행자'

-> ['도시', '##계', '##획', '##사업', '##시', '##행', '##자']

-> 총 7개

- '도시계획사업시행자'

-> ['도시', '##계획', '##사업', '##시행', '##자']

-> 총 5개

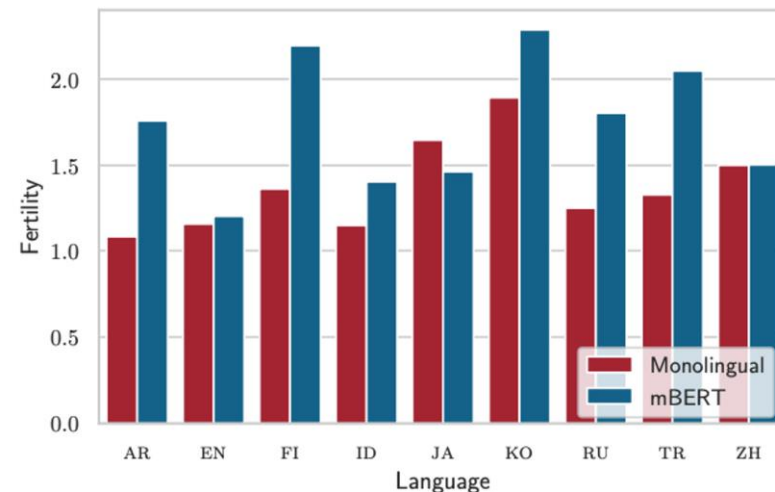


Figure 2: Subword fertility (i.e., the average number of subwords produced per tokenized word (Ács, 2019)) of monolingual tokenizers versus the mBERT tokenizer.

## 2) % of continued words

- The number of words that were split into at least two subwords
- 최소 2개 이상의 subword로 쪼개지는 단어 (전체 단어 중 몇 %)

e.g.

- 학교에 -> ['학교에'] ❌
- 학교에 -> ['학교', '##에'] ⓪

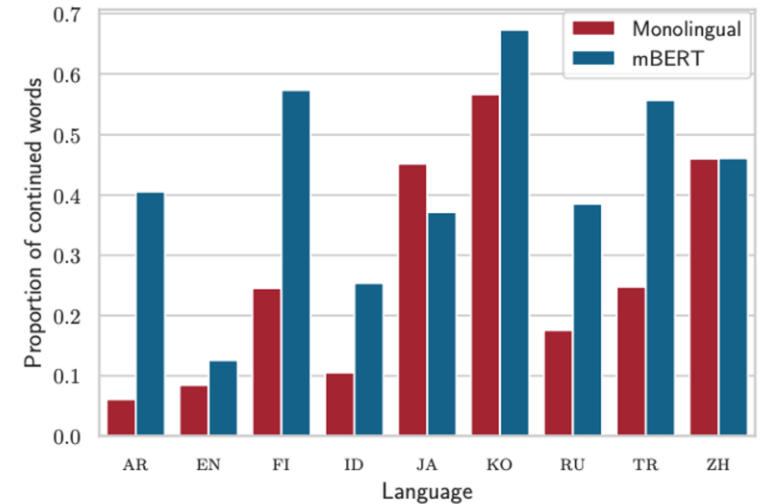


Figure 3: Proportion of continued words (i.e., words split into multiple subword tokens (Ács, 2019)) in monolingual corpora tokenized by monolingual models vs. mBERT.

## 3.3. Difficulty of Task



도메인 특화 언어모델 만드는거 너무 어려워요ㅠ

걱정 마세요 **필요하지 않을지도 몰라요**

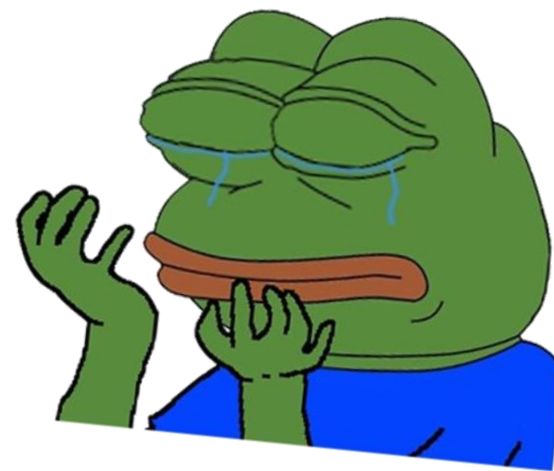


# 사실은 모두 의미 없던 게 아니었을까?

뭐?! 막상 개고생했는데



필요 없었다고...?



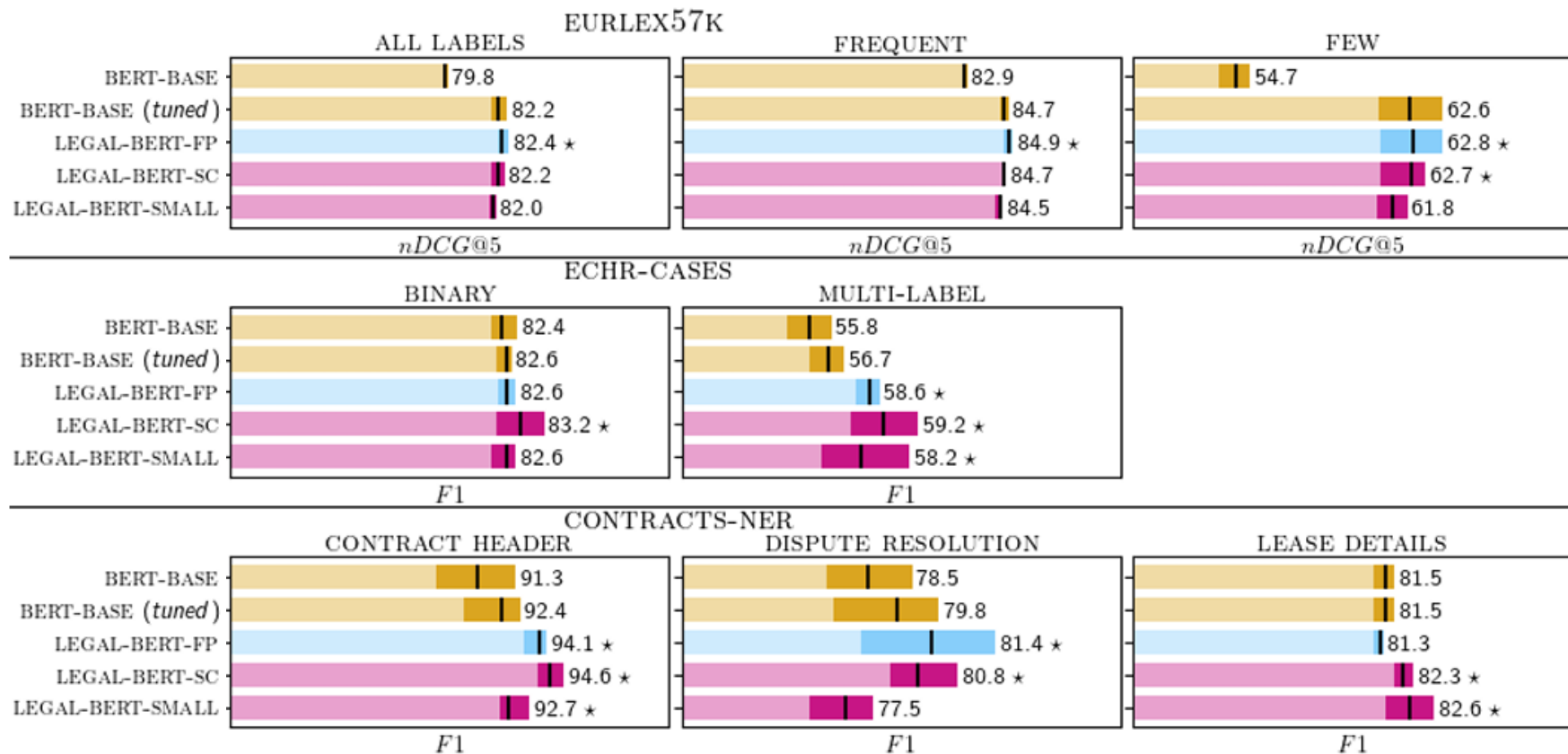
One of the emerging puzzles for law has been that while *general* pretraining (on the Google Books and Wikipedia corpus) boosts performance on a range of legal tasks, **there do not appear to be any meaningful gains from *domain-specific* pretraining (domain pretraining) using a corpus of law.** Numerous studies have attempted to apply comparable Transformer architectures to pretrain language models on law, but **have found marginal or insignificant gains on a range of legal tasks [7, 14, 49, 50].** These results would seem to challenge a fundamental tenet of the legal profession: that legal language is *distinct* in vocabulary, semantics, and reasoning [28, 29, 44]. Indeed, a common refrain for the first year of U.S. legal education is that students should learn the “language of law”: “Thinking like a lawyer turns out to depend in important ways on speaking (and reading, and writing) like a lawyer.” [29].

**Legal-BERT의 성능 향상이 생각보다 크지 않다!**

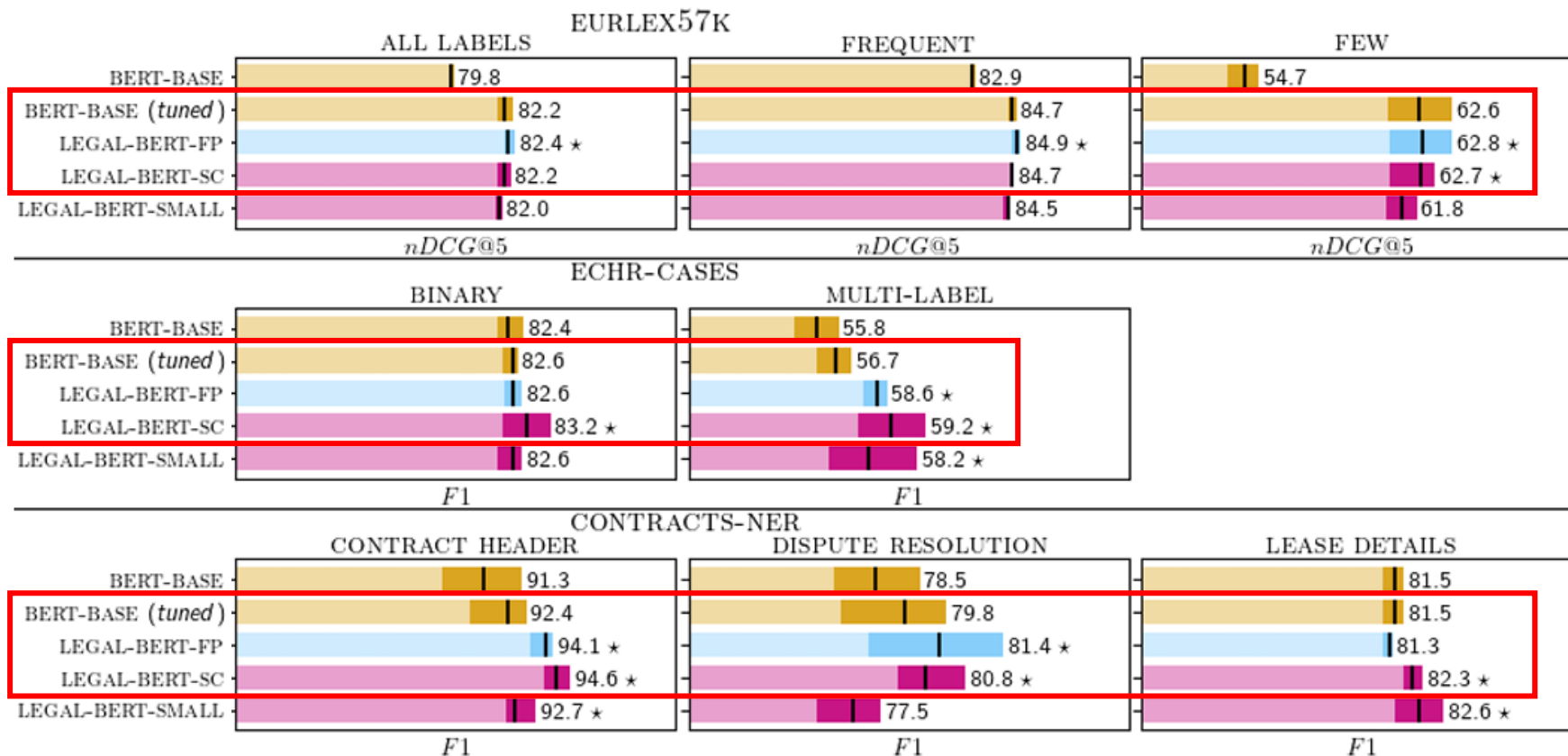
We hypothesize that the puzzling failure to find substantial gains from domain pretraining in law stem from the fact that **existing fine-tuning tasks may be too easy and/or fail to correspond to the domain of the pretraining corpus task.** We show that existing legal NLP tasks, Overruling (whether a sentence overrules a prior case, see Section 4.1) and Terms of Service (classification of contractual terms of service, see Section 4.2), are simple enough for naive baselines (BiLSTM) or BERT (without domain-specific pretraining) to achieve high performance. Observed gains from domain pretraining are hence relatively small. Because U.S. law lacks any benchmark task that is comparable to the large, rich, and challenging datasets that have fueled the general field of NLP (e.g., SQuAD [36], GLUE [46], CoQA [37]), we present a new dataset that simulates a fundamental

**그 이유는 Legal Task의 난이도가 쉬워서이다**





# Legal-BERT의 성능표를 살펴보자



0.0% ~ 0.2%

0.0% ~ 2.5%

-0.2% ~ 1.7%

**Table 4: Performance with different pre-training strategies**

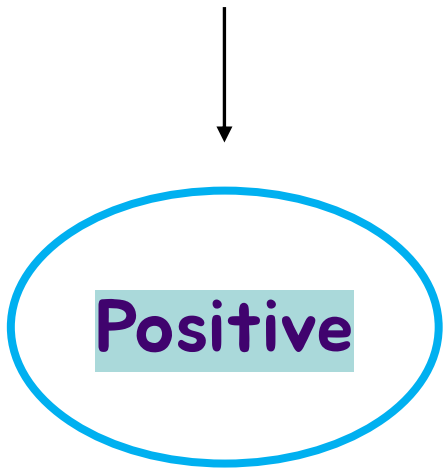
Model	Loss	Accuracy	F1 Score
Vanilla BERT	0.38	0.85	0.84
FinBERT-task	0.39	0.86	<b>0.85</b>
FinBERT-domain	<b>0.37</b>	<b>0.86</b>	0.84

**Bold face** indicates best result in the corresponding metric. Results are reported on 10-fold cross validation.

1%.....!



Sales increased due to growing market rates and increased operations.  
(시장 요율 상승과 운영 증가로 인해 매출이 증가했습니다.)



먼저 내가 해결하려는 Task의 난이도를 파악해보세요!

쉬운 난이도라면 필요하지 않을 수도!



반대로 난이도가 어렵다면  
도메인 특화 언어모델은 필요합니다!



4. 저도....만들어 보고 싶어요!

# 저도....만들어 보고 싶어요!





# 요즘은 기본 텍스트 데이터가 많이 있습니다

모두의 말뭉치 | 인공지능의 언어 능력 평가

모두의 말뭉치 | 돌아가기 | 회원 가입

말뭉치 신청 | 사용자 참여 | 말뭉치 활용 | 알립니다

모두의 말뭉치 | 미래를 준비하는 소중한 우리말 자원

말뭉치 신청 | 말뭉치 신청 내역

총 37건 | 보기 | 자세히 보기

<p>신규</p> <p>신문 말뭉치 2022</p> <p>(버전 1.0) 2022년 생산된 신문 기사 중 해제로부터 저작권 이용을 위한 받은 기사를 기계 분석 가능한 형식으로...</p> <p>신청하기</p>	<p>신규</p> <p>일상 대화 음성 말뭉치...</p> <p>(버전 1.0) 일상 대화의 음성(PCM 파일) 과 전사 자료로 구성된 말뭉치입니다.</p> <p>신청하기</p>	<p>신규</p> <p>일상 대화 말뭉치 2021</p> <p>(버전 1.0) 특정 주제 또는 제시 자료로 자유롭게 대화를 하는 일상 대화 말뭉치입니다.</p> <p>신청하기</p>	<p>수정</p> <p>일상 대화 음성 말뭉치...</p> <p>(버전 1.3) 일상 대화의 음성(PCM 파일) 과 전사 자료로 구성된 말뭉치입니다.</p> <p>신청하기</p>
<p>수정</p> <p>일상 대화 말뭉치 2020</p> <p>(버전 1.3) 특정 주제 또는 제시 자료로 자유롭게 대화를 하는 일상 대화 말뭉치입니다.</p> <p>신청하기</p>	<p>수정</p> <p>개체명 분석 말뭉치 개...</p> <p>(버전 1.2) 개체명 분석 말뭉치에 워키피디어 연결 정보를 부착한 자료입니다.</p> <p>신청하기</p>	<p>수정</p> <p>개체명 사전 2021</p> <p>(버전 1.1) 개체명 및 개체 연결 정보가 부착된 말뭉치에서 개체 표현, 개체 유형, 지식베이스 연결 정보를 추출...</p> <p>신청하기</p>	<p>수정</p> <p>2022 인공지능 언어...</p> <p>(버전 1.0) 2022년 국립국어원 인공지능 언어 능력 평가 대회 과제 말뭉치입니다.</p> <p>신청하기</p>

AI Hub | AI 데이터 찾기 | AI 개발지원 | 참여하기 | 정보공유 | 고객센터 | AI 허브소개 | 로그인 | 회원가입

데이터 분야 | AI 데이터 찾기 > 데이터 분야

검색어를 입력해주세요 | 데이터셋 검색

객체별 검색 | 데이터 유형별 검색

분야 선택 | 한국어 | 영상이미지 | 헬스케어 | 재난안전환경 | 농축수산 | 교통물류

데이터유형 선택 | 이미지 | 비디오 | 텍스트 | 오디오 | 3D | 센서 | 초기화

데이터셋 (75건) \* 데이터 다운로드 PC에서만 가능합니다. | 최신순

<p>기술과학 분야 한-영 번역 병렬 말뭉치 데이터 NEW</p> <p>6726   21   548 / 660.49 MB</p> <p>갱신년월 : 2022-10   구축년도 : 2021</p> <p>다운로드</p>
<p>일상생활 및 구어체 한-영 번역 병렬 말뭉치 데이터 NEW</p> <p>6725   25   699 / 513.94 MB</p> <p>갱신년월 : 2022-12   구축년도 : 2021</p> <p>다운로드</p>
<p>방송 콘텐츠 한-중, 한-일 번역 병렬 말뭉치 데이터 NEW</p> <p>2040   9   179 / 355.50 MB</p> <p>갱신년월 : 2022-10   구축년도 : 2021</p> <p>다운로드</p>
<p>전문분야 영-한-중-한 번역 말뭉치 (식품) NEW</p> <p>1831   4   175 / 1.05 GB</p> <p>갱신년월 : 2022-12   구축년도 : 2021</p> <p>다운로드</p>
<p>대규모 구매도서 기반 한국어 말뭉치 데이터 NEW</p> <p>4107   21   421 / 188.87 MB</p> <p>갱신년월 : 2023-01   구축년도 : 2021</p> <p>다운로드</p>
<p>온라인 구어체 말뭉치 데이터 NEW</p> <p>6254   39   637 / 1.59 GB</p> <p>갱신년월 : 2022-10   구축년도 : 2021</p> <p>다운로드</p>

# 코드도....모두 준비되어 있다고!

The screenshot displays the GitHub repository page for `huggingface/transformers`. The left sidebar shows a file tree with the following structure:

- .circleci
- .github
- docker
- docs
- examples
  - flax
  - legacy
  - pytorch
    - audio-classification
    - benchmarking
    - contrastive-image-text
    - image-classification
    - image-pretraining
    - language-modeling (highlighted in red box)
      - README.md
      - requirements.txt
      - run\_clm.py
      - run\_clm\_no\_trainer.py
      - run\_mlm.py
      - run\_mlm\_no\_trainer.py
      - run\_plm.py
    - multiple-choice
    - question-answering
    - semantic-segmentation

The main content area shows the repository details for `huggingface/transformers` (Public). It includes navigation links for Code, Issues (425), Pull requests (118), Actions, Projects (25), Security, and Insights. The commit history shows a recent commit by `younesbelkada` [BLIP] update blip path on slow tests (#21476) with 12,081 commits. Below the commit history is a file list:

File	Description	Last Commit
.circleci	A new test to check config attributes being used (#21453)	last week
.github	Remove Niels from templates (#21564)	3 days ago
docker	Use torch 1.13.1 in push/schedule CI (#21421)	2 weeks ago
docs	[bnb] Introducing BitsAndBytesConfig (#21579)	10 hours ago
examples	Bump werkzeug from 2.0.3 to 2.2.3 in /examples/research_projects/d...	yesterday
model_cards	Update URL for Hub PR docs (#17532)	9 months ago
notebooks	Add tutorial doc for TF + TPU (#21429)	2 weeks ago
scripts	transformers-cli login => huggingface-cli login (#18490)	6 months ago
src/transformers	[ImageProcessor] Refactor default mean & std to `OPENAI_CLIP_...	1 hour ago

The right sidebar contains the repository's "About" section, describing it as "Transformers: State-of-the-art Machine Learning for Pytorch, TensorFlow, and JAX." It also includes a list of tags such as `python`, `nlp`, `machine-learning`, `natural-language-processing`, `deep-learning`, `tensorflow`, `pytorch`, `transformer`, `speech-recognition`, `seq2seq`, `flax`, `pretrained-models`, `language-models`, `nlp-library`, `language-model`, `hacktoberfest`, `bert`, `jax`, `pytorch-transformers`, and `model-hub`.

5. 도메인 특화 언어모델이 도움이 되나요?

## General Domain

## Legal Domain

■ koelectra v3 base 
 ■ klue roberta base 
 ■ BHSN Language Model (Base)

NSMC

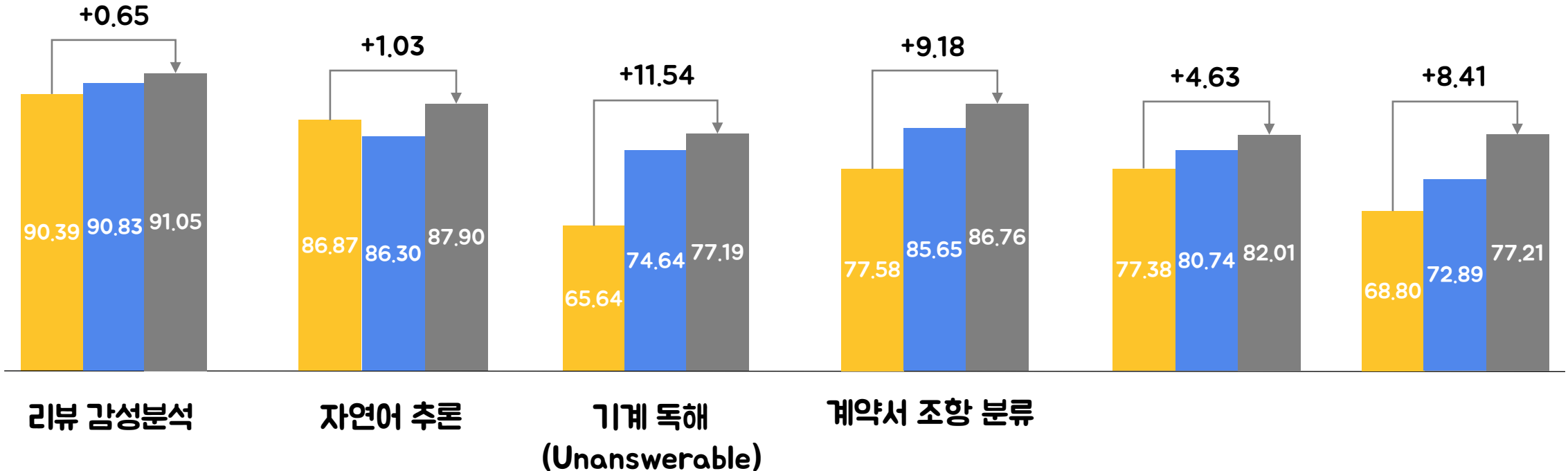
KLUE NLI<sup>1</sup>

KLUE MRC<sup>1</sup>

LEDGAR<sup>2</sup>

BHSN Task A

BHSN Task B



<sup>1</sup> KLUE의 경우 Dev Set으로 평가 <sup>2</sup> 한국어로 번역하여 평가



<https://careers.bhsn.ai>



감사합니다!

**BHSN**

박장원 (ML Engineer)

Email. [jwpark@bhsn.ai](mailto:jwpark@bhsn.ai)

Blog. <https://monologg.kr/about/>

LangCon 2023