What Changes Can Large-scale Language Models Bring? Intensive Study on HyperCLOVA: Billions-scale Korean Generative Pretrained Transformers

**What Changes Can Large-scale Language Models Bring?
Intensive Study on HyperCLOVA: Billions-scale Korean Generative
Pretrained Transformers**

Boseop Kim[*,1] **HyoungSeok Kim**[*,1] **Sang-Woo Lee**[*,1,2] **Gichang Lee**[1]
Donghyun Kwak[1] **Dong Hyeon Jeon**[3] **Sunghyun Park**[4] **Sungju Kim**[1,3]
Seonhoon Kim[3] **Dongpil Seo**[1] **Heungsub Lee**[1] **Minyoung Jeong**[1] **Sungjae Lee**[1]
Minsub Kim[1] **Suk Hyun Ko**[1] **Seokhun Kim**[1] **Taeyong Park**[1] **Jinuk Kim**[1]
Soyoung Kang[1] **Na-Hyeon Ryu**[1] **Kang Min Yoo**[1,2] **Minsuk Chang**[2] **Soobin Suh**[1,3]
Sookyo In[1,3] **Jinseong Park**[1,3] **Kyungduk Kim**[1,3] **Hiun Kim**[1] **Jisu Jeong**[1,2]
Yong Goo Yeo[1] **Donghoon Ham**[1] **Dongju Park**[1] **Min Young Lee**[1] **Jaewook Kang**[1]
Inho Kang[1,3] **Jung-Woo Ha**[1,2] **Woomyoung Park**[1] **Nako Sung**[1]

NAVER CLOVA[1] NAVER AI Lab[2] NAVER Search[3] Search Solutions, Inc.[4]

0. Abstract

- GPT-3는 LM의 강력한 in-context learning ability를 갖고 있다.
- GPT-3에 남아있는 issues
  - non-english LM, 다른 사이즈의 모델, in-context learning에 맞는 프롬프트 최적화 등..
- HyperCLOVA: 560B의 한국어 토큰을 학습한 82B GPT-3의 한국어 variant
  - korean-specific tokenization을 통해 in-context zero-shot, few-shot learning에서의 SOTA 달성
  - 프롬프트 기반 학습에 성능이 향상됨을 보이고, 프롬프트 엔지니어링 파이프라인에 어떻게 통합될 수 있는지 demonstrate
  - HyperCLOVA studio(인터랙티브 프롬프트 엔지니어링 인터페이스)를 머신러닝 비전문가들에게 쓰게 해서 No Code AI paradigm를 실현할 수 있는 가능성에 대해 논의한다.

1. Introduction

- GPT의 zero shot, few shot을 통한 in-context learning이 주목받고 있다
- 최근엔 파라미터 업데이트 없이 LM(large-scale language models)의 성능을 향상시킬 수 있는 프롬프트 기반 학습 방법이 보고 되고 있음
- GPT의 세 가지 현실적 문제점
  1. 학습한 corpus가 영어에만 치중된 것
  2. 다양한 사이즈의 모델을 아는 것이 좋지만 지금은 13B, 175B인 모델에만 접근할 수 있다.

3. backward gradient를 필요로 하는 advanced 프롬프트 기반 학습법은 아직 LM에서 실험되지 않았다.

contributions

1. 100B개의 파라미터를 가졌고, 한국어 중심 corpus를 학습한 한국어 in-context learning based LM, HyperCLOVA를 introduce
2. 비영어 언어를 학습하는 in-contex LM에 대한 language-specific tokenizationdml 효과를 알아본다
3. 39B 및 82B 파라미터를 사용하여 mid-size HyperCLOVA의 제로샷 및 퓨샷 기능을 탐색하고 프롬프트 기반 튜닝이 성능을 향상시켜 SOTA를 outperforming하는 것을 발견
4. HyperCLOVA Studio를 통한 no code AI 출시

2. Previous Work

2.1 Prompt Optimization

- 언어모델이 커질수록 “full fine-tuning paradigm”을 프롬프트 기반 접근으로 대체하는 것이 시간, 공간 복잡도를 고려했을 때 효과적이라는 가능성이 보고되고 있음
- 그러나 프롬프트 디자인에 굉장히 민감하기 때문에 프롬프트를 최적화하는 것이 중요함

1. discrete approach
    - token space를 최적화하여 transferability에 advantage
    - 그러나 interpretability(해석력)가 좋지 않아 suboptimal 하다는 연구 있음
2. continuous space에서 프롬프트를 최적화하는 새로운 방향

- p-tuning for autoregressive LM이 몇몇 task에서 MLM-based fine-tuning을 outperforming한다는 연구

2.2 Language Models

- language specific LM이 필요하지만 비용으로인해 limited in availability
- multilingual in-context learners는 아직 연구되지 않았고, 몇몇 소수의 주요 언어(ex: 중국어) 에 초점맞춰 연구되고 있음

3. Pre-training

3.1 Data Description

- 네이버와 external source에서 얻은 text data

Name	Description	Tokens
Blog	Blog corpus	273.6B
Cafe	Online community corpus	83.3B
News	News corpus	73.8B
Comments	Crawled comments	41.1B
KiN	Korean QnA website	27.3B
Modu	Collection of five datasets	6.0B
WikiEn, WikiJp	Foreign wikipedia	5.2B
Others	Other corpus	51.5B
Total		561.8B

Table 1: Descriptions of corpus for HyperCLOVA

3.2 Model and Learning

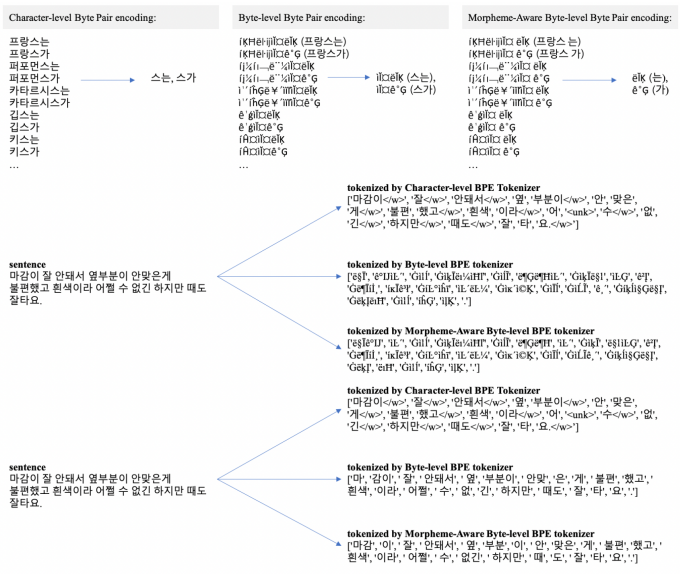
# Param	n_{layers}	d_{model}	n_{heads}	d_{head}	lr
137M	12	768	16	48	6.0e-4
350M	24	1024	16	64	3.0e-4
760M	24	1536	16	96	2.5e-4
1.3B	24	2048	16	128	2.0e-4
6.9B	32	4096	32	128	1.2e-4
13B	40	5120	40	128	1.0e-4
39B	48	8192	64	128	0.8e-4
82B	64	10240	80	128	0.6e-4

Table 2: Detailed configuration per size of HyperCLOVA

- GPT-3와 같은 transformer decoder architecture 사용
- mid-size 파라미터 모델의 capability와 representation power를 탐색해보고자 함
  - 현실에서 많이 쓸 수 있는 파라미터이기 때문에 유용할 것으로 생각
- megatron-LM에 기반, NVIDIA GPU, optimizer = AdamW with cosine learning rate scheduling and weight decay, mini batchsize = 1024, minimum learning rate = 1/10 of the original lr
- 82B 파라미터 with 150B token 학습하는 데 13.4일 걸림

3.3 Korean Tokenization

- 한국어는 명사 → particle → 동사/형용사의 stem → 종결어미 순으로 나오는 교착성(agglutinative) 언어
- **morpheme-aware byte-level BPE**를 토큰화 기법으로 사용



- morpheme analyzer로 문장을 pre-split →morpheme aware byte level BPE가 단일 바이트 character로 표현된 non-Korean character를 학습
- HuggingFace's tokenizer

4. Experimental Results

4.1 Experimental setting

- NSMC(네이버 영화 리뷰 데이터셋)
  - few-shot experiment
    - 12개의 in-context 70-shot leraning model의 test 정확도 평균
- KorQuAD 1.0(한국어 machine reading comprehension 데이터셋)
  - passage + questions
  - evaluation: paragraph, 4 QA pairs, question 넣어준다.

  → passage에 대해서는 zero-shot, question에 대해서는 4-shot learner
- AI hub Korean-English: 한 → 영, 영 → 한 번역 task
- YNAT(연합뉴스 헤드라인): 주제 분류
- KLUE-STS: 문장 간 유사성 예측
- Query modification task: AI 스피커 사용자들의 multi-turn query를 single-turn query로 바꾸는 task

- baseline: BERT, Transformer from Park et al.(2020), mBERT from Park et al.(2021)

4.2 In-context Few-shot Learning

- 모델 사이즈가 증가함에 따라 모델 성능이 단조증가한다.
- 그러나 한→영 번역, KLUE-STS는 베이스라인보다 낮았다

- 한→영의 성능이 낮은 것은 corpus에 영어 비율이 낮았기 때문이라고 생각되고, 더 정교화된 프롬프트 엔지니어링이 개선시킬 수 있을 것이라고 생각

	NSMC (Acc)	KorQuAD (EA / F1)		AI Hub (BLEU) Ko→En	AI Hub (BLEU) En→Ko	YNAT (F1)	KLUE-STS (F1)
Baseline	89.66	74.04	86.66	40.34	40.41	82.64	75.93
137M	73.11	8.87	23.92	0.80	2.78	29.01	59.54
350M	77.55	27.66	46.86	1.44	8.89	33.18	59.45
760M	77.64	45.80	63.99	2.63	16.89	47.45	52.16
1.3B	83.90	55.28	72.98	3.83	20.03	58.67	60.89
6.9B	83.78	61.21	78.78	7.09	27.93	67.48	59.27
13B	87.86	66.04	82.12	7.91	27.82	67.85	60.00
39B	87.95	67.29	83.80	9.19	31.04	71.41	61.59
82B	88.16	69.27	84.85	10.37	31.83	72.66	65.14

4.3 Prompt-based Tuning

Methods	Acc
Fine-tuning	
mBERT (Devlin et al., 2019)	87.1
w/ 70 data only	57.2
w/ 2K data only	69.9
w/ 4K data only	78.0
BERT (Park et al., 2020)	89.7
RoBERTa (Kang et al., 2020)	91.1
Few-shot	
13B 70-shot	87.9
39B 70-shot	88.0
82B 70-shot	88.2
p-tuning	
137M w/ p-tuning	87.2
w/ 70 data only	60.9
w/ 2K data only	77.9
w/ 4K data only	81.2
13B w/ p-tuning	91.7
w/ 2K data only	89.5
w/ 4K data only	90.7
w/ MLP-encoder	90.3
39B w/ p-tuning	**93.0**

- p-tuning을 통해 메인 모델에는 파라미터 업데이트 없이 성능 향상
  - 4K개의 example로 p-tuning한 게 RoBERTa를 150K개의 데이터로 fine-tuning한 것만큼의 결과
  - 생성 task를 위한 input side에 대한 p-tuning의 효과

Model sizes	Few-shots	p-tuning	BLEU
13B	zero-shot	×	36.15
		O	**58.04**
	3-shot	×	45.64
		O	**68.65**
39B	zero-shot	×	47.72
		O	**73.80**
	3-shot	×	65.76
		O	**71.19**

Table 5: Results of p-tuning on in-house query modification task.

  - p-tuning이 zero-shot, 3-shot에서 input query 퀄리티를 개선시켰다.
  - 큰 모델에서는 discrete prompt의 효과가 작은 것처럼 보인다.

- 선행 연구와도 비슷한 결과: LM의 크기가 커지면 discrete prompt가 사용되지 않아도 비슷한 성능을 보일 수 있다.
- 생성 task에 대해 input-side p-tuning을 적용시킨 첫 연구
- 이 결과는 GPT-3 스케일의 모델에서 input data에 대한 backwrd gradient에 접근할 수 있으면 프롬프트 최적화 방식이 LM 모델의 표현력을 향상시킬 수 있는 방법일 수 있다는 것을 암시한다.

4.4 Effect of Tokenization

	KorQuAD (EA / F1)		AI Hub (BLEU) Ko→En	AI Hub (BLEU) En→Ko	YNAT (F1)	KLUE-STS (F1)
Ours	**55.28**	**72.98**	3.83	**20.03**	**58.67**	**60.89**
byte-level BPE	51.26	70.34	**4.61**	19.95	48.32	60.45
char-level BPE	45.41	66.10	3.62	16.73	23.94	59.83

- byte-level BPE, char-level BPE 비교
  - out of vacabulary(OOV): char-level BPE token에 포함되어 있지 않은 경우
- 1.3B 가벼운 모델로 pre-training 실험
- 대부분 byte-level이 잘 했지만, 한→영 task에서는 char-level이 더 잘함
- 결론: language-specific tokenization is essential for training large-scale LMs

5. Discussion on Industrial Impacts

“accelerating the life-cycle of NLP ML operation”

- 기존에는 ML 전문가들이 데이터셋과 well-defined object function이 필요했지만, 이제는 한 명의 비개발자가 프로토타입 시스템을 만들 수 있을 것이다.

5.1 HyperCLOVA Studio

- OpenAI Playground와 같은 GUI 제공
- API 지원

5.2 Case Studies on HyperCLOVA Studio

1. 성격이 있는 챗봇 프로토타입 만들기
   - 1-2줄의 성격 설명으로 만들 수 있었다.
2. zero-shot transfer data augmentation
   - 사용자 의도에 맞는 발화 생성하기
   - “source-domain classes” 제공
3. 이벤트 이름 생성
   - 제품 특성에 대한 키워드를 인상적인 이벤트 이름으로 만드는 sequence to sequence task
   - BLEU 점수가 낮은 것이 반드시 높은 퀄리티를 의미하는 것은 아니었으며, 오히려 더 창의적인 제목일 수 있다.

5.3 Opportunity of HyperCLOVA Studio

- HyperCLOVA Studio가 HyperCLOVA의 추가적인 AI 기능에 다양한 boost를 줄 수 있다.

1. input gradient API
    - 프롬프트 기반 최적화로 local downstream tasks의 성능을 향상 시킬 수 있다.
2. prompt injection module 적용
    - 적절한 문서를 넣어줌으로써 오픈 도메인 QA reader로 사용될 수 있다.
3. filters
    - 필터를 통해 남용을 막을 수 있다.

5.4 No/Low Code AI Paradigm

- 기존 머신러닝 개발 파이프라인

  1) 문제 정의와 사용자 리서치

  2) 데이터 수집, 어노테이션

  3) 모델 훈련/검증

  4) MLOps

  5) 오류 분석 및 사용자 모니터링
- 하이퍼 클로바 스튜디오와 같은 GUI를 도입함으로써 2-4단계를 한 단계로 줄일 수 있다.
    - examples 큐레이팅, 프롬프트 디자인, API 파라미터 튜닝, API integration
- ML 개발의 비용을 크게 줄일 수 있기 때문에 AI 기술을 도입하고자 하는 회사들에게 큰 도움이 될 것이다.

6. Conclusion

- HyperCLOVA with 82B 파라미터는 in context zero-shot and few-shot에서 SOTA 성능을 보이고, 프롬프트 기반 학습 방법으로 더 향상될 수 있다.
- HyperCLOVA Studio와 같은 프레임워크는 No Code AI paradigm을 달성할 수 있을 것이다.
- HyperCLOVA Studio 서비스로 AI에 익숙하지 않은 사용자들도 자신의 모델을 만들 수 있도록 하고자 한다.

p-tuning 참고

https://velog.io/@seopbo/GPT-Understands-Too

데이터 어노테이션

https://cordingdiary.tistory.com/86