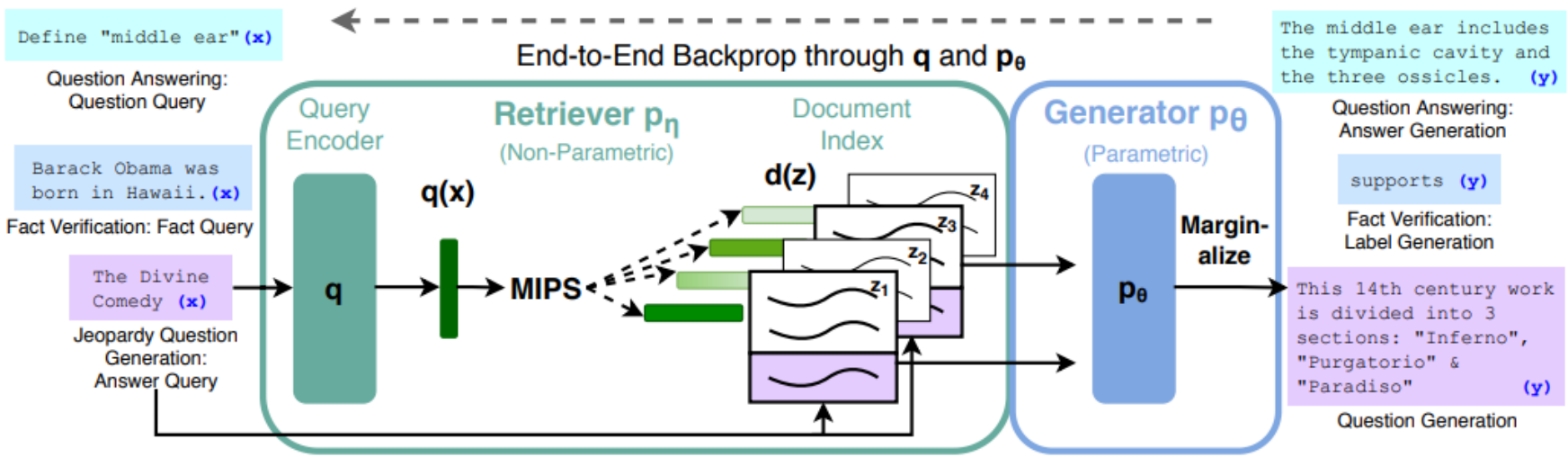
KSC 2022
Ko-kNN-LM: K-Nearest Neighbor를 이용한 문장생성 강화된 언어모델

이현민*, 나승훈, 임준호, 김태형, 류휘정, 장두성
전북대학교*, 한국전자통신연구원, KT
leehm@jbnu.ac.kr*

목차

1. 연구 개요
2. 관련 연구, kNN-LM
3. Ko-kNN-LM
   - 사전 학습된 언어모델
   - 데이터 스토어 구축
   - 데이터 검색
   - 결과
4. 실험 분석
   - kNN CASE
   - K값에 따른 결과값이 미미한 이유 분석

1. Ko-kNN-LM: 연구 개요

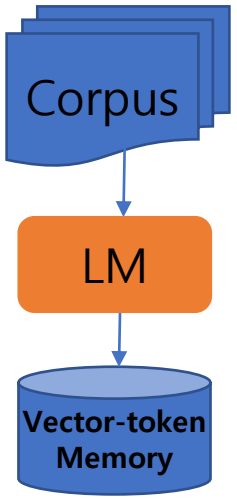


https://arxiv.org/pdf/2005.11401.pdf

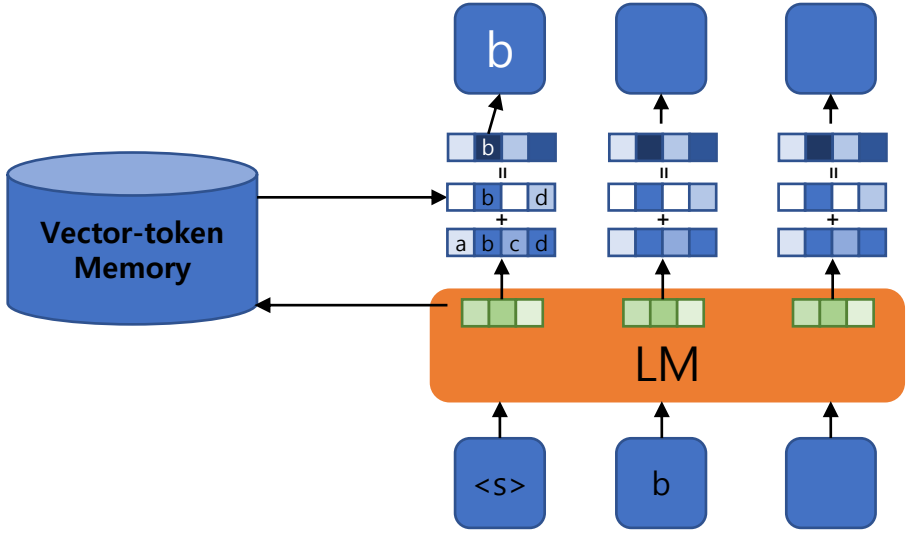
- 외부지식을 활용하여 언어모델을 강화하는 연구가 계속되고 있다.
- kNN-LM은 데이터의 representation을 token 단위로 외부지식에 저장한다.
- 외부지식를 이용하여 추가적인 학습없이 언어모델의 생성 token을 보강한다.
- Perplexity를 사용하여 모델의 성능을 비교한다.

2. Ko-kNN-LM: 관련 연구, kNN-LM

1 step.
대용량 말뭉치
외부지식 구축



2 step.
외부지식 활용
단어의 확률 보강



- Token 단위 외부지식 구축
- LM에 hidden state를 query로 외부지식에 최근접 이웃 검색: $L2(q,\ k) = \sqrt{\sum_{n=1}^{hidden}(q_n - k_n)^2}$
- 검색된 값들을 활용하여 최종 단어의 확률 보강
- $\mathrm{p(y|x)} = \lambda p_{kNN}(y|x) + (1 - \lambda)p_{LM}(y|x)$, λ는 하이퍼 파라미터 값

3. Ko-kNN-LM: 사전 학습된 언어모델

표 2. KoGPT2 구성

#Params	#Layers	#Heads	FFN dim	Hidden dim
125M	12	12	3072	768

- KoGPT2의 파라미터 구성
- Transformers의 디코더 기반 모델
- 사전학습된 언어모델 (SKT_gpt2, Train_data: 한국어 위키 백과, 뉴스, 모두의 말뭉치 v1.0, 청와대 국민청원)

3. Ko-kNN-LM: 외부지식 구축

Sliding window 기법을 사용하여 데이터스토어를 구축하였다. 한국어 Wiki 데이터의 약 80%를 사용하여 구축 하였고, 총 토큰수는 약 1억(107,274,239)개이다. 상세한 데이터 스토어의 크기는 표2와 같다.

표 2. 데이터 스토어 구성

	Key	value
size	107274239, 768	107274239, 1

데이터 스토어 구성

실험 설정

Data

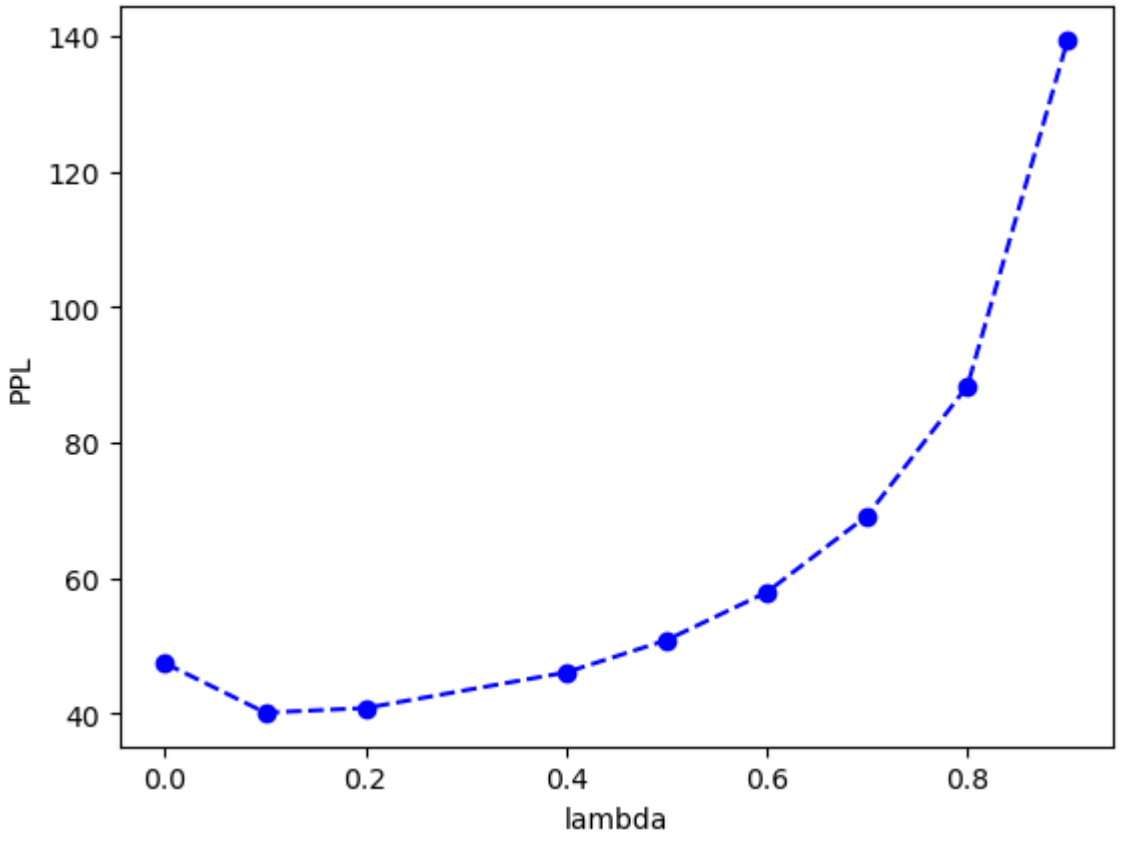
- 한국어 Wiki 데이터의 80% 만 사용
- Token 약 100M
- Key = [107,274,239 - 768], value = [107,274,239 - 1]
- Data slicing = 1024
- Sliding window 사용
- Ex) [0, 1024], [0,1024], [512, 1536], [1024, 2048], ⋯ , [total_length – 1024, total_length]
- 각 배치마다 512개의 representation 추출

3. Ko-kNN-LM: 데이터 검색

Distance, p_{kNN}, faiss, np.memmap

- Faiss-L2 distance(Euclidean distance): $L2(q,\ k) = \sqrt{\sum_{n=1}^{768}(q_n - k_n)^2}$
- $p_{kNN}(y|x) = \sum_{(k_i,\, v_i)\, \in N} 1_{y=v_i} \exp(-d(k_i, f(x)))$
- Create a memory-map to an array stored in a *binary* file on disk.
- 적은 양의 ram을 사용하고도 데이터스토어 생성 및 저장가능
- This study is a proof-of-concept for the index. It showed that vanilla Faiss with an on-disk file-based storage can be used with interactive performance.
- The code for these experiments can be found here: Distributed on-disk index. It runs at scale 1/1000th on the Deep1B dataset.
- Faiss, indexing 1T vectors https://github.com/facebookresearch/faiss/wiki/Indexing-1T-vectors
- Memmap: https://numpy.org/doc/stable/reference/generated/numpy.memmap.html

3. Ko-kNN-LM: 결과



Lambda (λ) 값에 따른 PPL 변화 (k = 8)

파라미터 Lambda (λ)

- λ 값에 따른 PPL 변화
- $p(y|x) = \lambda p_{kNN}(y|x) + (1 - \lambda) p_{LM}(y|x)$
- λ 값을 0.1씩 증가하며 측정
- λ 값이 0 때가 기존 LM 모델
- 0.1에서 성능이 가장 좋은 것으로 나온다.
- 기존 47.44에 비해 7.44 낮아진 40.00 기록
- 하지만 K값에 다른 ppl 변화가 미미하다는 한계점이 존재

3. Ko-kNN-LM: 결과

실험 결과 요약

- 사전 학습된 언어모델과 대규모 데이터를 통해 추가적인 학습 없이(모델의 파라미터 값 변경 없이) 외부 지식을 구축하고 검색하는 것 만으로도 언어모델의 성능 향상을 이끌어 낼 수 있음.
- Base모델 ppl 47.44 보다 7.44 낮은 40.00 을 기록
- 하이퍼 파라미터 λ=0.1 일 때 가장 성능이 좋았음
- K값의 변화에 다른 성능 변화는 미미하는 한계점이 존재

4. 실험 분석 – kNN Case


```
"label": "체가",
"lm": "체, 체의, 체와, 적으로, 적, 적인, 체가, 체(",
"knn": "체가, 적인, 체, 체와, 체의, <pad>, <d>, </s>",
"lm+knn": "체가, 체, 체의, 체와, 적으로, 적, 적인, 체(",
"lm_probability": "0.2346770167350769, 0.18711385130882263, 0.16013170778751373, 0.11449795216321945, 0.08899658918380737, 0.084327638149
"knn_probability": "1.0, 1.7164902439270304e-18, 2.459234779866574e-20, 3.9922008991892005e-29, 9.240438399591135e-33, 0.0, 0.0, 0.0",
"knn_lm_percent": "0.20559227466583252, 0.19942660629749298, 0.15900781750679016, 0.1360786110162735, 0.09729941934347153, 0.075628578662
```


고유성과 자생성은 유기체가 오늘날 서식하는 곳에서 …

Case 1: kNN의 도움으로 올바른 정답을 고르는 경우

- 가장 이상적인 경우

4. 실험 분석 – kNN Case


```
"label": "될",
"lm": "될, 되며,, 되는데,, 되기, 되지만,, 되기도, 되고,, 되",
"knn": "되기도, </s>, <d>, <sys>, <pad>, <unk>, <s>, <usr>",
"lm+knn": "될, 되기도, 되며,, 되는데,, 되기, 되지만,, 되고,, 되",
"lm_probability": "0.5139708518981934, 0.10324905067682266, 0.09430097788572311, 0.08083964139223099, 0.07599081099033356, 0.06488565355
"knn_probability": "1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0",
"knn_lm_percent": "0.44798123836517334, 0.18494664132595062, 0.08999272435903549, 0.08219350874423981, 0.0704604983329773, 0.06623421609
```


자생종은 염두에 두고 있는 구역 외의 지역에서도 발견 될수도 …

Case 2: kNN이 LM의 top K를 맞추는 경우

- 이상적이진 않지만, 어색하지 않은 문장 생성이 기대된다.

4. 실험 분석 – kNN Case


```
"label": "성과",
"lm": "종은, 종이, 종의, 종을, 종, 종에, 종과, 종(",
"knn": "종은, 종,, 성, <d>, <pad>, </s>, <sys>, <unk>",
"lm+knn": "종은, 종이, 종의, 종을, 종, 종에, 종과, 종(",
"lm_probability": "0.6638354659080505, 0.11213230341672897, 0.10312147438526154, 0.03964695706963539, 0.03671017289161682, 0.01763640344142913
"knn_probability": "1.0, 1.9426926972480885e-13, 7.293113732076881e-23, 0.0, 0.0, 0.0, 0.0, 0.0",
"knn_lm_percent": "0.6986895799636841, 0.10050621628761292, 0.09242964535951614, 0.03553628548979759, 0.032903991639614105, 0.0158078279346227
```


고유성과 자생성은 유기체가 오늘날 서식하는 곳에서 …

Case 3: LM과 kNN이 모두 틀린 경우

- LM의 값을 query로 사용하기 때문에, 기존 LM의 값이 다른 값을 예측하면 kNN도 좋은 결과를 뽑아 내긴 어렵다.

4. 실험 분석 – k값에 따른 성능 변화가 미미한 이유 분석


```
"labels": "성과",
"values": "좋은, 종,, 성, 성, 성, 종에, 종, 종의",
"distance": "-875.3125, -904.58203125, -926.28515625, -935.2421875, -949.33984375, -993.765625, -1018.3984375, -1034.5390625",
"softmax": "1.0, 1.9426926972480885e-13, 7.29216773007934e-23, 9.394365023606213e-27, 7.084889592766181e-33, 0.0, 0.0, 0.0"
```


고유성과 자생성은 유기체가 오늘날 서식하는 곳에서 …

- 실제 kNN이 검색되는 value 값과 distance 값을 확인
- Distance 값이 커서 softmax 값이 제대로 증류되지 않음.
- 모델의 size 문제(실험 모델은 base 모델, large 모델로 키우면 어느정도 해결될 것으로 추측)
- 사전 학습된 언어모델의 데이터와 외부지식을 구성한 데이터 간의 간극으로 인한 문제로 추정

4. 실험 분석 – 결론

실험 분석 결론

- kNN 검색 결과가 잘 되는 것을 확인
- 크게 3가지 Case로 분류하였음
- K값에 따른 값이 미미한 이유는 모델의 size 문제, 언어모델을 학습한 데이터와 외부지식을 구성한 데이터 간의 간극으로 추측

추후 실험 방향

- GPT2 Large 모델을 특정 데이터를 사용하여 사전 학습하고 외부지식 구축
- 다운 스트림 태스크에 모델을 미세조정 후에 실험 ex) Summary, QA

감사합니다.