2024 년 봄 POSTECH 컴퓨터공학과 과제연구 진행보고서

Conformalized Retrieval for RAG

학 번: 20210792
이 름: 최준하
연구 지도교수: 박상돈
학 과: 컴퓨터공학과

연구 목적 (Problem statement)

최근 대규모 언어 모델(LLM)은 다양한 자연어 처리(NLP) 작업에서 획기적인 성과를 거두었습니다. LLM 은 대량의 텍스트 데이터를 학습하여 텍스트 생성, 번역, 질의응답 등 다양한 작업을 수행할 수 있습니다. 하지만 LLM 은 학습되지 않은 정보에 대한 query 에 대해 거짓된 정보를 출력하는 hallucination 문제가 있습니다. 이를 해결하기 위한 방법 중 하나로 RAG 모델이 쓰입니다. RAG(Retrieval Augmented Generation) 모델은 검색 결과를 기반으로 텍스트를 생성하는 LLM 모델입니다. RAG 모델은 검색 엔진(Retriever)과 생성 엔진(Generator)으로 구성됩니다. Retriever 는 질의와 관련된 텍스트를 검색하고, Generator 는 검색 결과를 기반으로 새로운 텍스트를 생성합니다.

본 연구는 RAG 모델의 Retriever 성능을 개선하기 위해 conformal prediction 을 사용하여 Top-K 를 설정하는 연구를 수행합니다. Conformal prediction 은 모델이 내놓는 대답의 집합이 높은 확률로 정답을 포함하도록 하는 것이고, Top-K 설정은 Retriever 가 질의와 관련된 상위 K 개의 텍스트만 검색하도록 설정하는 방법입니다. 적절한 Top-K 설정을 통해 Retriever 의 성능을 개선하여 RAG 모델의 전체적인 성능을 개선하려 합니다.

연구 배경 (Motivation and background)

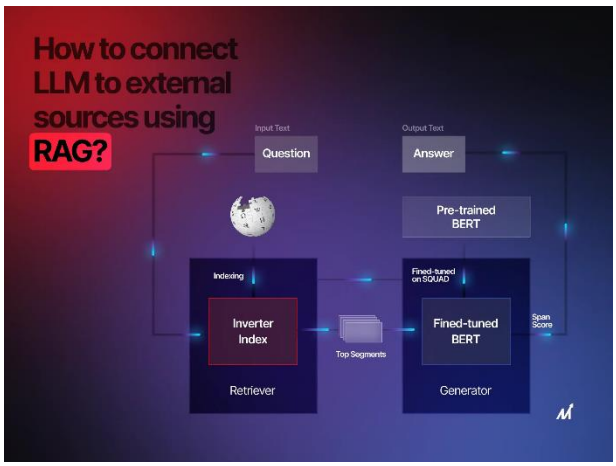

Figure 1. RAG 의 구조를 간략히 표현한 그림

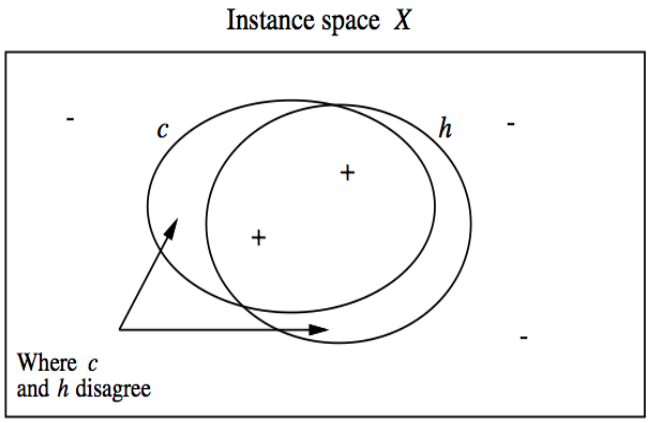

Figure 2. 고민중..

기존 LLM 은 generator 가 최신 데이터를 학습하지 못해 최신 데이터에 대한 query 에 대해 거짓된 정보를 출력하는 문제가 있었습니다. RAG 는 retriever 를 통해 검색 결과를 generator 에 넣어 해당 문제를 해결하는 데 도움을 주었습니다. Retriever 성능의 향상은 곧 LLM 성능의 향상이기 때문에 retriever 의 성능 향상을 위한 연구가 진행되고 있습니다. 지금까지 query expansion 이나 document ranking 등의 방법을 사용한 연구가 retriever 의 성능을 향상시켰습니다.

하지만 여전히 hallucination 문제는 해결되지 않았습니다. 본 연구에서는 모델의 신뢰성을 중심으로 이 문제를 바라보려 합니다. 지금까지의 연구는 모델을 학습할 때 학습 데이터를 가장 잘 설명하는 모델을 찾는 것에 주력했습니다. 그리고 이 과정에서 모델이 overfitting 되는 것을 방지하기 위해 regularizer 를 도입하기도 했습니다. 하지만 이렇게 학습된 모델을 신뢰할 수 있는 것은 아닙니다. 이는 신뢰할 수 있는 모델이 흔히 학습에 사용되는 목적 함수의 기준에 부합하지 않기 때문입니다.

Figure 2 는 일반적으로 완전히 신뢰 가능한 모델이 존재할 수 없음을 나타냅니다. c 는 target concept 이고, h 는 hypothesis 입니다. 학습 목적은 c 와 가장 비슷한 h 를 찾는 것이 될 것입니다. 하지만, c 와 h 가 정확히 일치하는 것은 불가능에 가깝습니다. Conformal prediction 은 correctness 을 다르게 해석하여 '모델이 출력한 대답이 정답을 포함하는 것'을 correct 하다고 보고, 이 확률과 데이터셋의 크기에 대해 논합니다. 본 연구는 RAG 의 retriever 가 90%의 확률로 정답 document 를 포함하는 것을 보장하는 적절한 top-K 를 설정하여 retriever 의 결과를 출력하고, 이 retriever set 과 generator 가 결합한 LLM 모델의 성능과 hallucination 발생 빈도를 기존 모델과 비교하는 것이 목적입니다.

연구 방법 (Design and methodology)

본 연구는 huggingface 에 공개된 facebook 의 RAG 코드를 바탕으로 합니다. transformers/examples/research_projects/rag/distributed_pytorch_retriever.py 의 retrieve 부분을 수정할 예정입니다. 자세한 내용은 다음과 같습니다.

1. huggingface 의 rag-sequence-base 모델과 rag-token-base 모델의 학습 데이터를 얻습니다.

2. 학습 데이터를 바탕으로 conformal prediction 개념을 활용해 top-k 가 정답 데이터를 포함할 확률이 90% 이상인 k 를 구합니다.

3. 해당 k 로 학습시킨 retriever 가 실제로 정답 데이터를 90% 이상으로 포함하는지 확인합니다.

4. generator 와 이 retriever 를 결합한 RAG 와 기존 RAG 의 성능과 hallucination 의 발생 빈도를 비교합니다.

5. (시간이 남는다면) query 와 answer 에 따라 k 가 유동적으로 변하는 모델을 생성해보고, 위에서 만든 k 가 불변하는 모델과 비교합니다.

연구 진행 상황 (Progress report)

- 코드 분석

기존 제안서에서 계획한 대로 먼저 RAG 코드를 공부하고 코드를 분석하였습니다. 가장 먼저 코드에서 retriever 를 찾았고, distributed_pytorch_retriever.py 에 있는 RagPyTorchDistributedRetriever class 가 retriever 임을 확인했고, 이를 수정하여 conformal prediction 을 적용하기로 했습니다.

- top-K 에 따른 score 경향 파악

적절한 top-K 설정을 하기 전에, K 값에 따른 score 의 변화 경향을 파악하기 위해 K 값을 수정하며 정답 데이터를 포함할 확률을 구해보았습니다. 이때, score 측정 방식은

transformers/examples/research_projects/rag/eval_rag.py 의 get_precision_at_k 함수를 사용했습니다. Figure 3 은 K 에 따른 score 를 나타냅니다. Figure 4 에서 확인할 수 있듯이 K 가 증가할수록 score 가 감소하는 경향을 보였습니다. 학습 데이터인 question set 과 gold_data 를 살펴보면, 한 질문마다 정답이 한 개가 아니라 여러 개인데, 기존 score 측정 방식이 정답 데이터와 모델의 출력 데이터가 평균적으로 몇 개 일치하는지 구하는 방식입니다. 지금은 모델이 출력한 데이터가 정답을 포함할 확률을, 'top-K document 에 정답이 하나라도 있을 확률'로 해석하고 score 측정 방식을 수정했습니다. 이후 다시 score 를 측정한 결과를 Figure 4 에서 확인할 수 있고, 현재 목표로 잡은 threshold 인 90%의 신뢰도를 최초로 달성하는 K 값은 50 이었습니다.

k	score
1	68.80
2	40.58
3	29.85
5	20.11
10	10.06
15	6.70
20	5.03
50	2.01
100	1.01

Figure 3 기존 함수로 측정한 score

k	score
1	68.80
2	75.84
3	79.26
4	81.40
5	82.62
7	84.30
10	85.74
49	89.95
50	90.01
100	91.31
500	93.05
1000	93.51

Figure 4 바꾼 함수로 측정한 score

연구 추진 일정 (Future plan)

4/29 ~ 5/5: 데이터 추출 및 데이터 간 관계 파악

5/6 ~ 5/10: conformal prediction 을 RAG-sequence-base 모델의 retriever 에 적용 및 evaluation 진행

5/11 ~ 5/17: conformal prediction 을 RAG-token-base 모델의 retriever 에 적용 및 evaluation 진행

5/18 ~ 5/25: 기존의 RAG 모델과 비교 및 분석

5/26 ~ 5/27: 포스터 제작

5/28 ~ 5/31: 최종 보고서 작성