**How Language Models Use Long Contexts**

* LLM(gpt-3.5-turbo-0613)에게 context를 주고 문답을 진행하면서 질문에 대한 답의 위치에 따라서 성능이 확연하게 차이가 났다. 답의 위치가 앞과 뒤에 있을 수록 성능이 괜찮았고(앞에 있을 때가 가장 좋음), 중간으로 향할 수록 성능이 안좋아졌다.(국자 형태의 성능 그래프) 답이 중간에 위치해 있다면 context를 주지 않았을 때의 성능이 오히려 더 좋았다.
  + [retrieval-augmented generation](https://arxiv.org/abs/2005.11401" \o "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks) 논문의 setup과 비슷하게 설정하여 진행했다.
* 첫번째 실험: Multi-Document Question Answering
  + 주어진 여러 문서들에서 질문과 연관된 내용을 추리하여 답변해야 하는 task이다.
  + 실험 Setup:-
    - 1. Context의 크기: 문서를 추가하거나 빼서 크기를 조절한다.
    - 2. 질문과 연관된 내용의 위치: 주어지는 문서들의 순서를 바꿔서 위치를 조절한다.
    - 3. Greedy encoding 방식을 사용: Randomness가 없는 환경에서 실험을 진행했다.
  + LLM(gpt-3.5-turbo-0613)에게 context를 주고 문답을 진행하면서 질문에 대한 답의 위치에 따라서 성능이 확연하게 차이가 났다. 답의 위치가 앞과 뒤에 있을 수록 성능이 괜찮았고(앞에 있을 때가 가장 좋음), 중간으로 향할 수록 성능이 안좋아졌다.(국자 형태의 성능 그래프) 답이 중간에 위치해 있다면 context를 주지 않았을 때의 성능이 오히려 더 좋았다.
  + Context가 길어질 수록 성능이 감소했다.
* 두번째 실험: Key-Value Retreival Task
  + Json 포맷의 입력이 여러개 주어졌을 때 특정 key에 대한 value를 반환하는 task이다.
  + 첫번째 실험과 비슷한 결과가 나옴
  + Query-aware contextualization(질문을 앞에 한번, 뒤에 한번 놓는 방법)을 사용하면 key-value task를 완벽하게 해내지만, Multi-document QA에서는 연관 내용이 초반에 있을 때만 적용됐다.
  + 모델 구조(decoder-only vs encoder-decoder) - encoder-decoder 모델이 연관 내용의 위치 변화에 비교적 Robust한 모습을 보여줬다. 하지만 이 성능은 학습에 사용된 context length 이내로 한정된다. 테스트에서 긴 context length가 사용되면 decoder-only와 비슷하게 국자 형태의 성능 저하를 보여줬다.
  + Instruction fine-tuning되지 않은 모델 또한 첫번째 실험에서 국자 형태의 성능을 보여줬다.
* [이전 연구](https://arxiv.org/abs/1805.04623)에도 볼 수 있듯, 언어 모델은 최근 토큰(end of the input context)에 편향되어 있다는 것을 확인할 수 있다.
* 하지만 적절한 Instruction format의 프롬트가 주어진다면 long-range information에서 제한적으로 초반에 있는 관련 내용을 활용하는 것이 가능하다.
* 위 테스트 결과를 토대로 답일 확률이 높은 내용을 context 초반과 끝에 주면 성능에 긍정적인 영향을 미칠 것으로 보인다.
* 긴 섹션에 대한 해결책
  + 섹션은 적당한 크기로 나누고 vector similarity와 기존의 Information Retreival Task에 사용되었던 모델로 관련 있는 문서 찾기
    - BM25
    - [Contriever](https://arxiv.org/pdf/2112.09118.pdf" \o "Unsupervised Dense Information Retrieval with)