

Found-in-the-Middle: 위치 주의 편향 보정을 통한 RAG 파이프라인 성능 혁신 기술 백서

1.0 서론: 긴 컨텍스트의 역설과 '중간 실종(Lost-in-the-Middle)' 현상

대규모 언어 모델(LLM)의 컨텍스트 창이 수백만 토큰 규모로 확장되고 있음에도 불구하고, 모델이 입력된 컨텍스트를 균일하고 효과적으로 활용하지 못하는 근본적인 문제가 드러나고 있습니다. 이론적으로 LLM은 정보의 위치와 무관하게 컨텍스트를 처리해야 하지만, 실제로는 그렇지 않습니다. 이러한 불균일한 컨텍스트 활용 능력은 방대한 외부 지식을 참조하여 답변의 정확성과 신뢰도를 높이는 RAG(검색 증강 생성) 시스템의 잠재력을 저해하는 핵심적인 아키텍처 과제로 부상했습니다. 이 문제의 핵심에는 Liu 등의 연구에서 명명된 '중간 실종(Lost-in-the-middle)' 현상이 있습니다. 이는 LLM이 입력된 컨텍스트의 시작이나 끝 부분에 위치한 정보는 비교적 잘 활용하지만, 그 중간에 위치한 핵심 정보는 놓치는 경향을 보이는 현상을 의미합니다. 이로 인해 정답 정보의 위치에 따른 모델의 성능은 양 끝이 높고 가운데가 움푹 파인 U자형 곡선(U-shaped performance curve)을 그리게 됩니다. 즉, 동일한 정보라도 프롬프트의 어느 위치에 제공되는지에 따라 모델의 답변 품질이 예측 불가능하게 달라지는 것입니다. 이 현상은 RAG 시스템에 중대한 신뢰성 문제를 야기합니다. 성능 저하가 심각할 경우, 모델의 성능은 관련 문서를 제공했음에도 불구하고 문서를 전혀 제공하지 않은 'closed-book' 설정보다도 낮아질 수 있습니다. 예를 들어 Liu 등의 연구에서 GPT-3.5-Turbo는 정답 문서가 중간에 위치했을 때의 정확도가 어떠한 문서도 제공받지 않았을 때의 정확도(56.1%)보다 낮아지는 역설적인 결과를 보였습니다. 이는 성능이 내용의 관련성이 아닌 문서 순서에 의해 좌우될 수 있음을 시사하며, 이는 예측 가능하고 안정적인 시스템을 구축하는 데 큰 장애물입니다. 이러한 성능 저하의 원인은 단순한 버그가 아니라 LLM의 아키텍처에 내재된 편향에서 비롯됩니다. 다음 섹션에서는 이 문제의 근본 원인이 되는 LLM의 내재적 편향을 심층적으로 분석하여, 보다 효과적인 해결책을 모색하기 위한 기반을 다룰 것입니다.

2.0 근본 원인 분석: LLM의 U자형 위치 주의 편향(Positional Attention Bias)

'중간 실종' 현상은 우연히 발생하는 오류가 아니라, LLM의 핵심 작동 원리인 주의(attention) 메커니즘에 내재된 구조적 편향에서 기인합니다. 이 근본 원인을 정확히 이해하는 것은 표면적인 문제 해결을 넘어, RAG 시스템의 성능을 근본적으로 개선하기 위한 필수적인 첫 단계입니다. Hsieh 등의 연구에 따르면, LLM은 내용의 실제 관련성과는 무관하게 컨텍스트의 시작과 끝 부분에 위치한 토큰에 본질적으로 더 높은 주의 점수를 할당하는 경향을 보입니다. 이를 'U자형 위치 주의 편향(U-shaped Positional Attention Bias)'이라고 하며, 이 편향은 '중간 실종' 현상에서 관찰되는 U자형 성능 곡선과 직접적인 상관관계를 가집니다. 즉, 모델은 정보의 중요도가 아닌 '위치' 때문에 특정 정보를 더 주목하는 내재적 한계를 가지고 있습니다. 이러한 주의 편향의 개념은 다음의 시각적 분석을 통해 더 명확하게 이해할 수 있습니다.

- (a) 정확도 저하 (U-shape RAG performance): 정답을 담고 있는 핵심 문서(Gold doc)가 프롬프트의 시작이나 끝에 있을 때는 정확도가 높지만, 중간으로 갈수록 급격히 떨어지는 U자형 성능 곡선을 보입니다. 이것이 바로 '중간 실종' 현상의 가시적인 결과입니다.
- (b) 모델 주의 편향 (U-shape model attention): 모델이 처리하는 문서의 내용과 상관없이, 모델의 주의력 자체는 프롬프트의 시작과 끝 부분에 자연스럽게 집중됩니다. 이는 모델의 주의 할당량이 선천적으로 U자형 패턴을 띤다는 것을 의미합니다.

- **(c) 중간 정보 압도 (Gold doc attention overwhelmed):** 정답 문서가 프롬프트 중간에 위치할 경우, 모델이 해당 문서에 어느 정도 주의를 기울이기는 합니다. 하지만 시작과 끝 부분에 대한 훨씬 더 강력한 위치 편향 때문에, 중간에 위치한 정보에 대한 주의력은 결국 압도당하고 최종 답변 생성 과정에서 무시되기 쉽습니다. 이 관찰은 매우 중요합니다. 모델이 관련 문서에 주의를 할당하기는 하지만, 그 신호가 위치 편향이라는 노이즈에 묻혀버리는 것입니다. 이는 만약 우리가 편향을 정확하게 측정하고 제거할 수 있다면, 진짜 관련성 신호를 복원할 수 있음을 시사하며, 이는 다음 섹션에서 수학적으로 공식화할 개념입니다. 그림 1: **'Lost-in-the-Middle'** 문제와 그 근본 원인 시각화. (a) RAG 시스템의 특징적인 U자형 성능 곡선으로, 정답 문서가 컨텍스트 중간에 있을 때 정확도가 저하됨. (b) 내용과 무관하게 프롬프트의 시작과 끝을 선호하는 모델의 내재적인 U자형 주의 편향. (c) 중간에 위치한 관련성 높은 **'Gold doc'**에 대한 주의가 더 강한 위치 편향에 의해 압도되는 현상. 이러한 위치 편향은 실험을 통해 명확히 입증됩니다. 한 실험에서 모델의 응답은 정답 문서의 실제 위치와 관계없이 프롬프트의 첫 번째 문서 내용에 강력하게 편향되는 경향을 보였습니다. 더욱 결정적인 증거는 문서들의 순서를 무작위로 섞어 첫 번째 문서의 내용이 계속 바뀌어도, 모델은 여전히 그 위치에 있는 문서에 가장 큰 영향을 받는다는 점입니다. 이는 모델의 편향이 내용의 관련성이 아닌 순수한 '위치'에 기반하고 있음을 실증적으로 보여줍니다. 결론적으로, '중간 실종' 현상은 모델이 중간 정보에 '못 보는' 것이 아니라, 시작과 끝에 대한 과도한 주의 편향 때문에 '덜 보게 되는' 구조적인 문제입니다. 다음 섹션에서는 이 근본적인 주의 편향을 수학적으로 모델링하고 교정하여 문제를 해결하는 **'Found-in-the-Middle'** 메커니즘을 상세히 다루겠습니다.

3.0 해결책: 'Found-in-the-Middle' 주의 보정 메커니즘

'중간 실종' 현상의 근본 원인이 위치 주의 편향임을 확인한 이상, 해결책은 이 편향을 직접적으로 교정하는 데 있습니다. LLM을 불변의 블랙박스로 취급하고 문서 재정렬과 같은 입력단에서의 휴리스틱으로 편향을 우회하는 대신, 이 접근법은 주의 계층에 직접 개입합니다. 이는 모델의 순방향 패스(forward pass)에 통합된 보정 모듈처럼 작동하여, 최종 생성에 영향을 미치기 전에 주의 분포를 교정합니다. 이 섹션에서는 제안된 **'Found-in-the-Middle'** 주의 보정 메커니즘의 이론적 배경과 구체적인 작동 방식을 엔지니어와 개발자가 이해할 수 있도록 상세히 설명합니다.

주의 메커니즘의 이중 요인 모델링

우선, 모델이 특정 문서에 할당하는 주의 값은 두 가지 주요 요인의 함수로 모델링할 수 있다는 가설에서 출발합니다.

- **(a) 문서의 실제 관련성 (Relevance):** 해당 문서가 사용자의 질문에 얼마나 관련이 있는가.
- **(b) 입력 내 문서의 위치 (Positional Bias):** 해당 문서가 전체 컨텍스트의 어느 위치에 있는가. 즉, 우리가 관찰하는 주의 값은 순수한 관련성과 위치에 따른 편향이 결합된 결과물이라는 것입니다. 우리의 목표는 이 두 요소를 분리하여 위치 편향의 영향을 제거하는 것입니다.

보정 메커니즘의 수학적 원리

'Found-in-the-Middle'의 핵심 아이디어는 간단한 수학적 원리를 통해 위치 편향을 측정하고 제거하는 것입니다. 이 과정은 다음 3단계로 구성됩니다.

- **1단계 (선형 모델 가정):** 복잡한 주의 메커니즘을 단순화하여, 특정 위치(k)에 있는 문서(x_{doc})에 대한 관찰된 주의 값 $Attn(x_{doc}, k)$ 을 관련성($rel(x_{doc})$)과 위치

편향(**bias(k)**)의 합으로 근사합니다. 이는 다음과 같은 선형 모델로 표현할 수 있습니다.

- **2단계 (편향 분리):** 특정 위치(**k**)가 갖는 순수한 편향 값 **bias(k)**를 측정하기 위해 내용적으로 관련성이 거의 없는 '더미 문서(**x_dum**)'를 활용합니다. 각 위치에서 보정을 위해 동일한 더미 문서(예: 빈 문자열 또는 "테스트 문서입니다"와 같은 일반적인 문구)를 사용하므로, 더미 문서의 내재적 관련성 **rel(x_dum)**은 고정된 상수 값입니다. 따라서 더미 문서를 특정 위치 **k**에 놓고 주의 값을 측정하면, **Attn(x_dum, k)**는 해당 위치의 편향 **bias(k)**를 대표하게 됩니다.
- **3단계 (보정된 주의 계산):** 실제 문서의 주의 값 **Attn(x_doc, k)**에서 더미 문서의 주의 값 **Attn(x_dum, k)**을 빼서 위치 편향 **bias(k)** 항을 상쇄시킵니다.
- **rel(x_dum)**이 상수이므로, 이 계산 결과는 위치와 무관하게 문서의 순수한 관련성 **rel(x_doc)**에 비례하는 '보정된 주의(**calibrated attention**)' 값이 됩니다. 이를 통해 모든 문서의 관련성을 위치 편향 없이 공정하게 비교할 수 있습니다.

주의 재분배(Attention Rescaling) 구현

이렇게 계산된 '보정된 주의' 값은 모델의 원래 주의 가중치를 재분배하는 데 사용됩니다. 이는 관련성이 높은 문서에 주의가 집중되도록 유도하는 과정입니다.

1. 먼저, 각 문서에 대해 계산된 보정된 관련성 점수(**rel(x_doc_k)**)를 **Softmax** 함수에 통과시켜 새로운 문서별 가중치 **ak**를 계산합니다. 온도(**temperature**) 하이퍼파라미터 **t**를 사용하여 가중치 분포의 날카로움을 조절할 수 있습니다.
2. 이 문서별 가중치(**ak**)는 해당 문서(**k**) 내의 각 토큰(**i**)에 대한 원래의 토큰별 주의 값을 재조정하는 데 사용됩니다. 전체적인 재조정 수식은 다음과 같습니다. 이 과정을 통해 모델의 최종 주의 할당량은 위치가 아닌 실제 관련성에 따라 결정되며, 중간에 있더라도 중요한 문서에 더 많은 주의를 기울이게 됩니다. 이 보정 메커니즘은 **LLM**이 컨텍스트를 인식하는 방식을 근본적으로 개선합니다. 다음 섹션에서는 이 메커니즘이 실제 **RAG** 파이프라인에서 어떻게 측정 가능한 성능 향상을 가져오는지 구체적인 실험 결과를 통해 입증하겠습니다.

4.0 성능 분석: RAG 파이프라인 개선 효과 검증

'Found-in-the-Middle' 메커니즘이 이론에 그치지 않고, 실제 **RAG** 작업에서 얼마나 실질적이고 의미 있는 성능 향상을 가져오는지 정량적 데이터를 통해 증명하는 것은 기술의 가치를 입증하는 데 필수적입니다. 이 섹션에서는 다양한 모델과 데이터셋에 걸친 실험 결과를 제시하여 기술의 효과와 범용성을 입증합니다.

보정된 주의 vs. 기본 주의 성능 비교

'Found-in-the-Middle' 메커니즘의 직접적인 효과를 검증하기 위해 **Vicuna**, **Tulu**와 같은 다양한 **LLM**을 사용하여 **NaturalQuestion**, **SynthWiki** 데이터셋에서 실험을 진행했습니다. 실험 결과, 주의 보정은 '중간 실종' 문제를 해결하는 데 탁월한 효과를 보였습니다. 특히, 정답 문서가 컨텍스트의 가장 불리한 위치인 중간에 있을 때, 'Found-in-the-Middle'을 적용한 모델은 기본 모델 대비 **6~15%p**에 달하는 일관된 정확도 향상을 달성했습니다. 이는 모델이 이전에는 놓쳤을 중간의 정보를 효과적으로 찾아내고 있음을 의미합니다. 또한, 거의 모든 실험 조건(총 24개 중 22개)에서 보정된 주의 메커니즘의 성능 곡선이 기본(**vanilla**) 주의 곡선보다 상위에 위치했습니다. 이는 특정 시나리오에 국한된 개선이 아니라, 모델의 긴 컨텍스트 활용 능력을 전반적으로, 그리고 근본적으로 개선했음을 시사하는 강력한 증거입니다. 그림 2: **NaturalQuestion** 데이터셋에서 보정된 주의(주황색)와 기본 주의(파란색)의 성능 비교. 보정된 접근법은 모든 문서 위치에서 기준선을 일관되게

능가하며, 특히 어려운 중간 컨텍스트 영역에서 가장 큰 정확도 향상을 보여 '중간 실종' 효과를 직접적으로 완화함을 입증한다.

기존 재순위화 기법 대비 우수성

'Found-in-the-Middle'은 단순히 주의를 재분배하는 것을 넘어, 문서의 실제 관련성을 추정하는 데 있어서도 기존의 다른 재순위화(re-ranking) 기법들보다 뛰어난 성능을 보였습니다. 아래 표는 NaturalQuestion 데이터셋에서 Recall@3(상위 3개 문서 내에 정답이 포함될 확률)을 기준으로 여러 기법의 성능을 비교한 결과입니다. | 방법 (Method) | 문서 10개 (K=10) | 문서 20개 (K=20) || ----- | ----- | ----- || 기본 주의 (Vanilla attention) | 0.3638 | 0.2052 || 쿼리 생성 (Query generation) | 0.6851 | 0.5815 || 관련성 생성 (Relevance generation) | 0.5521 | 0.4012 || 보정된 주의 (Calibrated attention) | **0.7427** | **0.6832** |

보정된 주의의 우수한 성능은 '쿼리 생성'과 같은 프롬프트 기반 재순위화 기법의 근본적인 한계를 보여줍니다. 이러한 기법들은 LLM에게 관련성을 추론 하도록 요청하는 반면, 우리의 접근법은 모델의 내재된 편향을 보정한 후 모델 자신의 주의 메커니즘을 통해 관련성의 대리 지표를 직접 측정 합니다. 이 직접적인 측정이 더 신뢰할 수 있는 신호임이 입증되었습니다.

기존 RAG 파이프라인과의 시너지

'Found-in-the-Middle'은 기존의 문서 재정렬(reordering) 기법을 대체하는 것이 아니라, 상호 보완적으로 작동하여 추가적인 성능 향상을 이끌어낼 수 있습니다. 재정렬은 입력 조작을 통해 위치 편향의 증상을 완화하려는 휴리스틱 접근법입니다. 반면, 주의 보정은 모델의 처리 계층 내에서 근본 원인을 교정하는 기계적 개입입니다. 이러한 근본적인 차이가 두 기법의 시너지 효과를 설명하며, 더 견고하고 예측 가능한 RAG 파이프라인을 가능하게 합니다. 실험적으로도 LongLLMLingua와 같은 재정렬 기법 위에 주의 보정 메커니즘을 추가로 적용했을 때 일관되게 가장 높은 성능을 달성했습니다. 이러한 기술적 분석과 검증 결과를 바탕으로, 다음 섹션에서는 이 기술이 AI 개발자들에게 제공하는 실질적인 가치와 실제 적용 시 고려해야 할 사항들을 종합적으로 논의하겠습니다.

5.0 결론: 실용적 가치와 향후 과제

본 백서는 LLM이 긴 컨텍스트를 처리할 때 발생하는 '중간 실종' 현상의 근본 원인을 진단하고, 이를 해결하기 위한 'Found-in-the-Middle' 주의 보정 메커니즘을 제안했습니다. 이 기술은 RAG 시스템의 신뢰성과 효율성을 한 단계 끌어올리는 근본적인 해결책으로서 중요한 실용적 가치를 지닙니다. AI 엔지니어와 개발자는 이 기술을 통해 더 길고 복잡한 문서 기반의 태스크를 안정적으로 처리하는 차세대 AI 애플리케이션을 구축할 수 있습니다.

핵심 요약

본 백서에서 다룬 핵심적인 내용은 다음과 같이 세 가지로 요약할 수 있습니다.

1. 문제 진단: LLM의 긴 컨텍스트 활용 능력은 모델의 내재적인 'U자형 위치 주의 편향'으로 인해 심각하게 제약을 받습니다. 이 편향은 내용의 관련성과 무관하게 컨텍스트의 시작과 끝에 과도한 주의를 할당하여 '중간 실종' 현상을 야기합니다.
2. 기술적 해결책: 'Found-in-the-Middle'은 더미 문서를 활용하여 위치별 순수 편향을 측정 한 뒤, 이를 실제 문서의 주의 값에서 차감하는 방식으로 위치 편향을 제거합니다. 그 결과, 내용의 실제 관련성에 따라 주의를 재분배하여 모델이 컨텍스트를 공정하게 보도록 교정합니다.
3. 검증된 효과: 실험 결과, 이 메커니즘은 RAG 작업의 정확도를 가장 어려운 시나리오에서 최대 15%p까지 향상시켰습니다. 또한, 기존의 문서 재정렬 기법과

결합하여 시너지를 창출하는 등, 현재 **RAG** 파이프라인에 즉시 적용 가능한 효과적인 솔루션임이 입증되었습니다.

한계점 및 고려사항

모든 기술과 마찬가지로 '**Found-in-the-Middle**' 역시 실제 적용 시 고려해야 할 한계점을 가지고 있습니다. 시스템 아키텍트의 관점에서 이러한 트레이드오프를 명확히 인지하는 것이 중요합니다.

- **계산 오버헤드:** 위치별 주의 편향을 측정하기 위해 더미 문서를 사용한 추가적인 모델 순방향 패스($O(K)$)가 필요합니다. 이는 지연 시간과 정확도 간의 트레이드오프를 발생시킵니다. 실시간 응답성이 중요한 애플리케이션에서는 $O(K)$ 오버헤드가 부담스러울 수 있지만, 오프라인 또는 품질 중심의 작업에서는 정확도 향상이 추가적인 계산 비용을 정당화할 것입니다.
- **편향 모델의 단순성:** 본 백서에서는 주의 편향을 설명하기 위해 단순한 선형 모델을 가정했습니다. 실제 **LLM** 내부의 주의 편향 메커니즘은 이보다 더 복잡하고 동적일 수 있으며, 이러한 복잡성을 완전히 포착하지 못할 수 있습니다.
- **편향의 잠재적 유용성:** 모든 위치 편향이 해로운 것은 아닙니다. 예를 들어, 대화 요약이나 법률 문서 분석과 같이 특정 위치(예: 결론 부분)에 핵심 정보가 집중되는 작업에서는 모델의 자연스러운 위치 편향이 오히려 유익하게 작용할 수도 있습니다. 따라서 보정 메커니즘을 적용하기 전, 대상 작업의 특성을 신중히 고려해야 합니다.

최종 결론

'**Found-in-the-Middle**'은 단순히 '중간 실종' 현상을 해결하는 기법을 넘어, **LLM**이 컨텍스트를 이해하고 활용하는 방식을 근본적으로 개선하는 중요한 진전입니다. 이 연구는 **LLM**의 성능을 저해하는 보이지 않는 편향을 정량적으로 분석하고, 이를 수학적 원리에 기반하여 교정할 수 있음을 보여주었습니다. 이는 향후 **LLM**의 주의 메커니즘과 그 영향을 이해하고, 더 공정하고 신뢰할 수 있는 **AI** 시스템을 구축하기 위한 새로운 연구 방향을 제시합니다. 이 기술을 통해 개발자들은 더욱 강력하고 안정적인 **RAG** 시스템을 구현하여 정보 검색과 생성의 패러다임을 한 단계 더 발전시킬 수 있을 것입니다.