JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 27, No. 3, pp. 849-856, Mar. 2026

Check for updates

# 문맥 단절 완화를 위한 순차적 문맥 주입 기반 청킹 기법

이 은 주[1] · 한 동 숭[2] · 이 성 민[3] · 박 경 수[4*]
[1]전주대학교 인공지능연구소 연구원
[2]전주대학교 문화콘텐츠학과 교수
[3]세희에스앤디 대표
[4]전주대학교 게임콘텐츠학과 교수

# Sequential Context Injection-Based Chunking for Mitigating Context Fragmentation

**Eun-Ju Lee[1] · Dong-Soong Han[2] · Sung-Min Lee[3] · Kyeong-Su Park[4*]**

[1]Researcher, Artificial Intelligence Research Institute, Jeonju University, Jeonju 55069, Korea
[2]Professor, Department of Cultural Contents, Jeonju University, Jeonju 55069, Korea
[3]CEO, Sehee Software & Design, Jeonju 54964, Korea
[4]Professor, Department of Game Contents, Jeonju University, Jeonju 55069, Korea

**[요 약]**

기존 청킹 방법은 각 청크를 독립적으로 처리하므로, 소설이나 대본 등 장기 의존성을 지닌 서사 데이터 분할 시 필수 문맥이 누락되는 한계가 있다. 이를 완화하기 위해 본 연구는 현재 청크의 이해에 필요한 과거 문맥을 선택적으로 요약 및 결합하는 순차적 문맥 주입 기반 청킹 기법을 제안한다. NarrativeQA 데이터셋 실험 결과, 제안 방법은 모든 설정에서 기존 방식보다 높은 Context Recall을 기록하며 적은 수의 검색 결과만으로도 정답 생성에 필요한 정보를 효과적으로 확보함을 확인하였다. 다만 본 연구는 제한된 데이터 규모에서 수행되었으므로, 향후 다양한 데이터셋을 통해 일반화 성능을 검증함으로써 제안 방법의 신뢰성과 실용성을 더욱 높일 수 있을 것으로 기대된다.

**[Abstract]**

Conventional chunking methods process each chunk independently, causing essential context to be omitted when segmenting narrative data with long-range dependencies, such as novels or screenplays. Therefore, this study proposes a sequential context injection–based chunking method that selectively summarizes and integrates past context necessary for understanding the current chunk. Experimental results on the NarrativeQA dataset show that the proposed method achieves higher context recall than existing approaches across all settings, effectively securing the information required for answer generation with fewer retrieved chunks. Although this study was conducted on a limited data scale, future studies should validate generalizability across diverse datasets to further enhance the reliability and practical applicability of the proposed method.

**색인어 :** 대규모 언어 모델, 검색 증강 생성, 청킹, 서사 데이터, 문맥 단절
**Keyword :** Large Language Models, Retrieval Augmented Generation, Chunking, Narrative Data, Context Fragmentation

**http://dx.doi.org/10.9728/dcs.2026.27.3.849**



**Received** 22 January 2026; **Revised** 19 February 2026
**Accepted** 05 March 2026

***Corresponding Author; Kyeong-Su Park**

**Tel:** +82-63-220-2758
**E-mail:** pine@jj.ac.kr

## Ⅰ. 서 론

최근 대규모 언어 모델(Large Language Models, LLM)은 방대한 사전 학습을 기반으로 다양한 분야에서 뛰어난 성능을 보이고 있다[1]. 그러나 LLM은 학습 데이터에 포함되지 않은 최신 정보를 반영하기 어렵고, 혹은 학습 데이터에 포함된 오정보나 편향 등으로 인해 사실에 근거하지 않은 출력을 생성하는 환각 문제가 존재한다[2]. 이러한 한계를 보완하기 위해 외부 지식베이스에서 관련 정보를 검색해 모델 입력으로 제공하는 검색 증강 생성(Retrieval-Augmented Generation, RAG)이 오류를 줄이고 정확도와 신뢰성을 향상시키기 위한 핵심 기술로 자리 잡고 있다[3],[4].

RAG는 LLM의 문맥 길이 제한을 고려하고 정보 과잉으로 인한 성능 저하를 방지하기 위해, 문서를 처리 가능한 단위로 분할하는 과정이 필수적이다. 이 과정에서 청킹(Chunking)은 원본 문서를 적절한 크기와 의미 단위로 재구성하는 역할을 수행하며, 이는 검색 단계의 정확도뿐만 아니라 최종 생성 결과의 품질에도 중요한 영향을 미치는 핵심적인 전처리 기술로 작용한다[5],[6].

그러나 현재 널리 사용되는 기존의 청킹 방식은 문서 전반에 걸쳐 의미가 축적되는 서사적 데이터의 특성을 충분히 반영하지 못한다는 점에 한계가 있다. 소설이나 시나리오 등과 같은 서사 데이터는 사건과 인물 관계가 긴밀히 연결된 장기 의존성을 가지는데, 기존 방식은 주체나 시점 등의 필수 문맥을 단절시켜 식별 정보를 누락시킨다[7]. 이를 보완하기 위해 검색 수를 증가시키는 방법도 고려될 수 있으나, 문맥 자체가 결여된 청크는 검색 양을 늘려도 올바른 정보를 제공하지 못할 가능성이 높으며, 오히려 불필요한 정보의 과부하로 인해 연산 비용 증가와 LLM의 추론 품질을 저하시킬 수 있다.

최근에는 이러한 한계를 극복하기 위해 문서 전체를 참조하여 각 청크에 문맥을 부여하는 전역적 문맥 보강 기법들이 제안되고 있다[8]. 다만 이 방식은 문맥 생성 과정에서 문서 전체를 반복적으로 참조해야 하므로, 긴 문서에서는 연산 비용과 지연이 증가할 수 있다. 따라서 본 연구는 서사 데이터의 순차적 전개 특성을 반영하여, 현재 청크 이전의 문맥 중 해당 청크의 이해에 실제로 기여하는 정보만을 선택적으로 활용하는 순차적 문맥 주입 기반 접근을 제안한다. 이는 문서 전체를 반복적으로 참조하지 않으면서도 시간적 흐름과 인과적 연결성을 보존하여, 각 청크가 검색 단계에서 충분히 식별 가능하고 독립적으로 해석 가능한 형태를 갖도록 하는 것을 목표로 한다.

## Ⅱ. 관련 연구

RAG는 외부 지식 소스에 기반하여 사실적 근거를 바탕으로 대규모 언어 모델의 응답 정확성을 향상시키는 중요한 접근 방식이다[9]. 이 과정에서 청킹은 문서를 임베딩 모델을 통해 벡터 표현으로 변환하기 위한 전처리 단계로서, 원본 문서를 적절한 단위로 분할하는 역할을 수행한다. 적절한 청킹은 불필요한 노이즈를 줄이고 문맥 정보를 보존함으로써, 검색 단계의 성능과 최종 생성 결과의 품질에 중요한 영향을 미친다[9],[10].

대표적으로 사용되는 청킹 방법으로는 고정 길이 분할 청킹(Fixed-size Chunking), 재귀적 청킹(Recursive Chunking), 의미론적 청킹(Semantic Chunking) 등이 있다. 고정 길이 분할 청킹은 텍스트를 일정한 크기의 청크로 분할하는 가장 단순한 방식이지만, 문장이 임의로 분리되어 의미 손실이 발생할 수 있다는 한계를 가진다[11].

재귀적 문자 기반 청킹은 문단, 문장, 단어와 같이 사전에 정의된 구분자를 단계적으로 적용하여 텍스트의 구조를 최대한 유지하며 분할하는 데 중점을 둔다[12]. 그러나 분할 기준이 의미적 단위가 아닌 기호나 분량에 기반하기 때문에, 서사적 문맥이 중간에 단절될 가능성이 존재한다.

의미론적 청킹은 문장 간 의미 유사도를 측정하여 의미나 문맥이 유사한 문장들을 하나의 청크로 묶는 방식으로[12], 텍스트 내의 의미 변화 지점을 중심으로 분할할 수 있다는 장점이 있다. 다만 문서 전체에 걸쳐 축적되는 장기적인 문맥이나 서사적 인과 구조를 충분히 반영하기에는 한계가 있다[13].

이러한 기존 청킹 방식의 한계는 인물 관계와 사건의 인과 구조가 문서 전반에 걸쳐 장기적으로 연결되는 서사 데이터에서 더욱 두드러진다. 대표적인 서사 데이터 벤치마크인 NarrativeQA[14]가 보여주듯, 소설이나 영화 대본과 같은 문서는 단순한 사실 검색이 아니라 문서 전반에 분산된 단서를 종합적으로 연결해야만 올바른 추론이 가능하다.

최근에는 이러한 문맥 정보 부족 문제를 해결하기 위해 문서 전체를 참조하여 각 청크에 문맥을 부여하는 전역적 문맥 보강 기법[8]이나, 검색 성능 향상을 위해 LLM을 활용하여 내용이 변화하는 지점을 탐지하고 청크를 동적으로 분할하는 기법이 제안되었다[15]. 또한, 검색된 개별 청크의 연관성을 LLM으로 직접 평가하여 불필요한 노이즈를 필터링하는 기법[16] 등 다양한 접근법이 활발히 연구되고 있다. 그러나 이러한 방식들은 문맥 생성 및 분할 과정에서 문서 전체의 반복적인 참조나 방대한 LLM 추론 연산을 필요로 하므로 비용과 처리 시간이 증가한다. 따라서 서사 데이터를 효율적으로 처리하기 위해서는 고비용의 전역 참조나 복잡한 동적 분할보다는 각 청크가 검색 단계에서 충분히 식별 가능하고 이해에 필요한 핵심 정보를 포함하도록 설계하는 효율적인 접근이 필요하다.

이에 본 연구는 검색 성능 향상을 위해 높은 연산 비용과 처리 시간을 요구하는 기존의 방식과 달리, 서사의 순차적 흐름에 기반하여 현재 청크 이해에 직결되는 과거 문맥만을 선택적으로 반영하는 접근법을 통해 연산 비용을 최소화하면서도 서사적 인과관계를 보존할 수 있다는 점에서 기존 방법들과 차별성을 가진다.

## Ⅲ. 제안하는 방법

본 연구는 서사 구조를 가진 문서의 청킹 과정에서 발생할 수 있는 문제를 개선하기 위해 순차적 문맥 주입 기반의 청킹을 제안한다. 기존 RAG 방법은 긴 문서를 고정된 단위로 분할하여 독립적인 청크로 처리하는 경우가 많으며, 이로 인해 청크 간의 연결성이 약화되어 서사 데이터에서 중요한 정보가 충분히 반영되지 않을 수 있다. 그 결과, 정답 생성에 필요한 청크가 누락되거나 관련도가 낮은 청크가 선택될 가능성이 존재한다. 이에 본 연구는 각 청크가 현재 청크 이전의 문맥을 포함하도록 설계함으로써, 서사의 흐름을 보다 안정적으로 반영하고자 한다.

### 3-1 순차적 문맥 주입 기반 청킹

청킹 과정에서 현재 청크에 필요한 선행 문맥을 선택적으로 수집하고, 요약하여 결합하는 절차로 구성된다. 먼저 전체 문서를 Recursive Chunking을 사용해 $n$개의 기본 청크 집합 $C=\{c_1, c_2, ..., c_n\}$로 분할한다. 이후 시점 $t$의 청크 $c_t$를 처리할 때, 이전 단계에서 분할된 과거 청크들 $\{c_1, c_2, ..., c_{t-1}\}$ 중에서 현재 청크 $c_t$와 가장 유사한 상위 $k_{past}$개의 청크를 선별하여 집합 $R_t$를 구성한다.

이렇게 선택된 $R_t$는 현재 청크의 이해에 필요한 핵심만 남기기 위해 LLM을 통해 $s_t$로 요약되며, 이를 최종적으로 원본 청크 $c_t$와 결합하여 문맥이 보강된 청크 $sc_t$를 생성한다. 이렇게 생성된 문맥이 보강된 청크들의 전체 집합은 $SC=\{sc_1, sc_2, ..., sc_n\}$와 같이 정의된다.

결과적으로 각 청크는 해당 시점을 이해하는 데 필요한 문맥 정보를 함께 포함하게 되며, 그림 1은 과거 청크 집합으로부터 관련 문맥을 선택하고 요약하여 현재 청크와 결합하는 전체 과정을 보여준다.

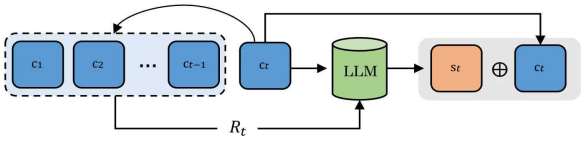



**그림 1.** 제안하는 방법
**Fig. 1.** The proposed method

### 3-2 프롬프트 설계 및 데이터 처리

제안하는 방법에서의 핵심은 문맥 요약 단계는 LLM이 현재 청크와 과거 문맥 사이의 핵심 연결 정보를 간결하게 추출하는 데 목적이 있다. 이를 위해 본 연구에서는 표 1에 제시한 프롬프트를 사용하여, 단순한 정보 나열보다 현재 청크를 이해하는 데 기여하는 서사적 단서인 인물, 사건, 배경 등을 중심으로 요약이 생성되도록 한다. 또한 검색된 문맥이 현재 청크와 충분히 관련되지 않은 경우에는 "No relevant information"을 반환하도록 설계하여, 불필요한 문맥 주입으로 인한 오류 및 노이즈를 최소화하였다.

**표 1.** 문맥 요약 생성을 위한 프롬프트
**Table 1.** Prompt for context-aware summary generation

| Content |
|---|
| You are an expert in narrative flow analysis.<br><br>[Retrieved Past Context]<br><br>[Current Chunk]<br><br>Based on the retrieved past context, summarize key information helpful for understanding the current chunk in 1–2 sentences. If not relevant, output "No relevant information". |

위 프롬프트를 적용한 예시는 표 2에 제시한다. 문맥 정보가 결여된 원본 청크에 선행 사건 정보가 요약되어 주입됨으로써, 개별 청크가 보다 독립적으로 해석 가능한 형태로 변환되는 과정을 확인할 수 있다.

**표 2.** 문맥 주입 결과 예시
**Table 2.** Example of context injection results.

| Stage | Content |
|---|---|
| Original | NICK: I'm talking personal, private stuff. The stuff that nobody on earth is supposed to hear, I hear that stuff! You know what I'm saying, man? I can hear what women think.<br>MORGAN: Can you? Good, 'cause that's not a talent a lot of guys have these days.<br>NICK: Oh, you don't believe me, huh? You want me to prove it?<br>NICK (O.S.): See this attorney coming toward us?<br>MORGAN (O.S.): Yeah. |
| Summary | In the previous scenes, Nick recounts a humorous incident where he accidentally electrocuted himself while trying on women's products, which resulted in him gaining the ability to hear women's thoughts. This sets the stage for the current chunk, where Nick is trying to convince Morgan of his newfound ability, leading to a challenge to prove it as a woman approaches. |
| Result | [Context Summary] In the previous scenes, Nick recounts a humorous incident where he accidentally electrocuted himself while trying on women's products, which resulted in him gaining the ability to hear women's thoughts. This sets the stage for the current chunk, where Nick is trying to convince Morgan of his newfound ability, leading to a challenge to prove it as a woman approaches.<br>NICK: I'm talking personal, private stuff. The stuff that nobody on earth is supposed to hear, I hear that stuff! You know what I'm saying, man? I can hear what women think.<br>MORGAN: Can you? Good, 'cause that's not a talent a lot of guys have these days.<br>NICK: Oh, you don't believe me, huh? You want me to prove it?<br>NICK (O.S.): See this attorney coming toward us?<br>MORGAN (O.S.): Yeah. |

표 2의 예시는 실험에 사용된 데이터셋에서 추출된 청크 중 하나이며, 제안하는 방법에 대해 가장 유사도 높은 과거의 청크 $k_{past}=1$를 적용하여 문맥을 생성한 결과이다. Original은 분할된 원본 청크를 의미하며, Summary는 이전 청크들로부터 현재 청크 이해에 필요한 정보만을 추출하여 요약된 문맥이다. 그리고 Result은 요약 문맥이 원본 청크 앞에 주입되어, 개별 청크가 보다 독립적으로 해석 가능한 형태로 변환된 결과를 의미한다.

표 2에서는 예시의 가독성을 높이기 위해, 의미 해석에 영향을 주지 않는 타임코드나 장면 표기와 같은 포맷 요소만을 표현상 정리하여 제시하였다. 이를 통해 문맥 주입이 서사적 인과관계와 사건의 배경 이해를 어떻게 보완하는지 확인할 수 있다.

## Ⅳ. 실험 및 분석

### 4-1 데이터셋

본 연구는 제안 방법의 효과를 검증하기 위해 NarrativeQA 데이터셋을 사용하였다. NarrativeQA는 소설 및 영화 대본과 같은 장문 서사 문서로 구성되며, 단순한 사실 검색을 넘어 인물 관계의 변화, 사건의 전개, 인과관계 등 문서 전반에 분산된 정보를 종합적으로 활용해야 답변이 가능한 질의응답 과제를 제공한다. 이러한 특성은 서사 데이터에서의 문맥 단절 문제를 다루는 본 연구의 목적을 검증하기에 적합하다.

다만 NarrativeQA 전체 데이터셋을 대상으로 LLM 기반 문맥 요약 및 생성 평가를 수행하는 것은 큰 비용과 처리 시간을 수반한다. 이에 본 연구에서는 LLM 활용에 따른 연산 제약을 고려하되, 결과의 객관성과 재현성을 확보하기 위해 Random Seed=42로 고정하여 NarrativeQA의 Train 데이터셋 중 5개를 무작위 추출하여 평가를 수행하였다. 추출된 5개 스토리는 문서 길이 및 질문 수 분포에서 이상치에 해당하지 않는 범위에 위치하여, 특정 길이 또는 질문 수 특성에 과도하게 편중되지 않은 표본임을 확인하였다. 본 연구에서 사용한 스토리들의 문맥 길이 및 질문 수 정보는 표 3에 제시한다.

**표 3.** NarrativeQA 데이터셋에서 선택된 5개 스토리 데이터
**Table 3.** Data of the 5 selected stories from the NarrativeQA dataset

| No. | Length (Chars) | Length (Tokens) | Questions |
|---|---|---|---|
| 1 | 307,482 | 59,522 | 30 |
| 2 | 283,946 | 51,337 | 30 |
| 3 | 345,973 | 65,825 | 28 |
| 4 | 147,698 | 49,756 | 29 |
| 5 | 209,986 | 74,488 | 29 |
| Total | 1,295,085 | 300,928 | 146 |

### 4-2 실험 설정

제안 방법의 효과를 검증하기 위해, 본 연구에서는 문맥 보강을 적용하지 않은 대표적인 세 가지 청킹 방법과 제안 방법을 비교 실험하였다. 또한, 제안 방법에서는 과거 문맥 선택 개수를 $k_{past}=1$로 설정하여 비교를 진행하였다.

텍스트 임베딩에는 OpenAI의 text-embedding-3-small 모델을 사용하였으며, 생성된 임베딩 벡터의 차원은 1,536차원이다. 청크 벡터들의 인덱싱 및 검색을 위해 FAISS (Facebook AI Similarity Search)를 활용하였다. 특히, 본 연구에서 사용한 임베딩 방법은 벡터의 길이가 1로 정규화되어 산출된다는 특징이 있다. 이러한 환경에서는 코사인 유사도나 유클리드 거리 등 거리 함수의 선택이 검색 순위에 영향을 미치지 않는다. 따라서 특정 거리 지표 사용으로 인해 발생할 수 있는 재현성 편차를 최소화할 수 있으며, 본 연구에서는 이러한 안정성을 바탕으로 FAISS의 근접 이웃 검색을 수행하여 현재 청크와 연관성이 높은 상위 k개의 문맥을 추출하였다. 최종 답변 생성에는 gpt-3.5-turbo를 사용하고, 추론의 일관성을 유지하기 위해 Temperature는 0으로 고정하였다. 또한 제안 방법에서 수행되는 문맥 요약 단계에는 비용 및 처리 속도를 고려하여 gpt-4o-mini를 사용하였다. 각 청킹 방법에 대한 구체적인 설정은 다음과 같다.

(1) Fixed Chunking
- LangChain의 CharacterTextSplitter를 사용하였다. 청크 길이는 1,000자, 청크 간 중첩은 200자로 설정하였으며, 구분자를 빈 문자열로 설정하여 공백 및 문장부호와 무관하게 텍스트를 분할한다.

(2) Recursive Chunking
- LangChain의 RecursiveCharacterTextSplitter를 사용하였다. 청크의 최대 길이는 1,000자, 청크 간 중첩은 200자로 설정하였으며, 분할 시 문단, 줄바꿈, 문장, 공백 등을 고려하기 위해 [“₩n₩n”, “₩n”, “. ”, “ ”, “”] 순으로 분할하였다.

(3) Semantic Chunking
- LangChain Experimental의 SemanticChunker를 사용하였다. 임베딩 기반 의미 유사도를 이용하여 문장 간 의미 변화 지점을 기준으로 분할하였으며, 임계값 설정에는 percentile 방식을 사용하였다.

(4) Proposed
- 기본적으로 Recursive Chunking과 동일하게 분할을 진행하며, 문맥 요약 정보가 추가적으로 결합되어 저장한다.

### 4-3 평가지표

각 청킹 방법에 따른 검색 성능을 평가하고 정량적으로 비교하기 위해 Context Recall을 주요 지표로 사용하였다. 이 지표는 정답을 도출하는 데 필요한 핵심 정보가 검색된 문맥 내에 얼마나 포함되어 있는지를 측정하는 지표로[17], 검색 단계의 품질을 가장 직접적으로 평가하는 방법이다.

값은 0에서 1 사이의 범위를 가지며, 1에 가까울수록 검색된 문서가 정답 생성에 필요한 핵심 정보를 갖고 있음을 나타낸다.

## V. 실험 결과 및 분석

### 5-1 실험 결과

본 연구에서 제안하는 방법과 기존 청킹 방법들과의 실험 결과는 표 4와 같다. 표의 수치는 Context Recall을 나타내며, $k$는 답변 생성을 위해 검색된 청크의 수를 의미한다. 제안 기법의 경우, 문맥을 반영하기 위한 과거 청크 수를 $k_{past}=1$로 설정하여 비교하였다.

**표 4.** 기존 청킹 방법과 제안하는 방법과의 성능 비교
**Table 4.** Performance comparison between baseline chunking methods and the proposed method

| Method | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
|---|---|---|---|---|---|
| Fixed Chunking | 0.327 | 0.466 | 0.523 | 0.554 | 0.596 |
| Recursive Chunking | 0.344 | 0.433 | 0.484 | 0.514 | 0.553 |
| Semantic Chunking | 0.339 | 0.470 | 0.543 | 0.592 | 0.616 |
| Proposed | **0.421** | **0.552** | **0.606** | **0.632** | **0.657** |

표 4에서 확인할 수 있듯이, 제안 기법은 모든 $k$값에 대한 설정에서 기존 방법론들보다 높은 성능을 보였다. 특히, 기본적으로 Recursive Chunking과 동일한 분할을 사용함에도 불구하고 성능 차이가 뚜렷하게 나타났는데, 이는 단순한 분할 전략의 차이가 아니라 분할 이후 단계에서 수행되는 순차적 문맥 선택 및 요약 주입 과정이 성능 향상에 핵심적으로 기여했음을 의미한다.

또한, 주목할 점은 낮은 $k$값에서의 성능 차이다. 표 4를 시각화한 그림 2를 보면, 제안 방법은 $k=2$일 때 이미 0.552를 기록하여 다른 청킹 방법들이 더 많은 청크 수를 사용할 때 달성한 성능과 유사하거나 더 높은 성능을 보였다. 이는 제안 방법이 단순히 검색 범위를 확장하지 않고도 각 청크에 과거의 핵심 문맥을 요약 및 주입함으로써, 적은 수의 청크만으로도 정답 생성에 필요한 정보를 충분히 확보할 수 있음을 보여준다.

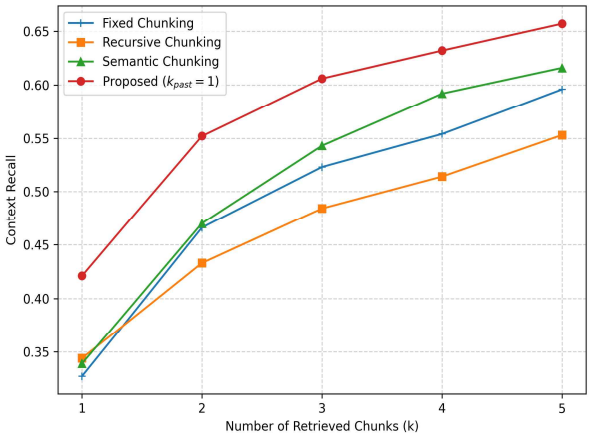



**그림 2.** 기존 청킹 방법과 제안하는 방법과의 성능 비교 시각화
**Fig. 2.** Visualization of performance comparison between baseline chunking methods and the proposed method

### 5-2 절제 연구

제안 모델 내부에서 문맥을 요약할 때 참조하는 과거 청크의 수가 성능에 미치는 영향을 분석하기 위해, $k_{past}$를 1, 3, 5로 증가시키며 성능을 비교하였으며, 결과는 표 5와 그림 3에 제시한다.

**표 5.** 과거 청크 수($k_{past}$)에 따른 실험 결과 비교
**Table 5.** Comparison of experimental results by the number of past chunks($k_{past}$)

| Method | $k=1$ | $k=2$ | $k=3$ | $k=4$ | $k=5$ |
|---|---|---|---|---|---|
| $k_{past}=1$ | 0.421 | **0.552** | **0.606** | **0.632** | **0.657** |
| $k_{past}=3$ | 0.416 | 0.544 | 0.600 | 0.620 | 0.643 |
| $k_{past}=5$ | **0.432** | 0.537 | 0.583 | 0.617 | 0.640 |

표 5에서 제시된 실험 결과, 참조하는 과거 청크 수가 $k_{past}=1$일 때, 모든 $k$값 실험에 대해 전반적으로 가장 높은 성능을 보였다. $k_{past}=3$의 경우 $k_{past}=1$와 유사하지만 성능이 소폭 낮게 나타났다. 반면, 가장 많은 과거 문맥을 참조하는 설정인 $k_{past}=5$의 경우 오히려 성능 저하가 관찰되었으며, 특히 $k=3$에서 결과가 0.583으로 가장 큰 성능 격차가 관찰되었다.

이러한 결과는 참조하는 과거 문맥의 범위가 지나치게 넓어질 경우, 현재 청크와의 관련성이 낮은 정보가 요약문에 과도하게 포함되어 검색 단계에서 노이즈로 작용하기 때문으로 해석된다. 따라서 NarrativeQA와 같은 서사 데이터에서는 가장 중요한 문맥 $k_{past}=1$에 집중하여 연결성을 강화하는 것이 가장 효율적인 전략임을 확인할 수 있다.

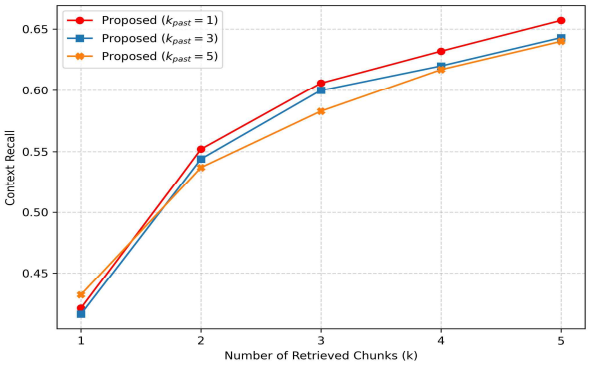



**그림 3.** 과거 청크 수($k_{past}$)에 따른 실험 결과 비교 시각화
**Fig. 3.** Visualization of experimental results comparison by the number of past chunks ($k_{past}$)

### 5-3 비용 및 효율성 분석

본 절에서는 실험에 사용한 데이터에 대해 각 청킹 방법에서 발생하는 토큰 비용을 정량적으로 분석하였으며, 그 결과는 표 6에 제시하였다. 여기서 'Total Chunks'는 분할된 청크의 총 개수, 'Final Chunk Tokens'는 최종 생성된 청크의 평균 토큰 수와 표준편차를 나타낸다. 또한, 'Additional Input Tokens'는 요약 문맥 생성을 위해 LLM에 입력되는 토큰 수의 평균과 표준편차를 의미하며, 여기에는 프롬프트, 참조된 과거 청크, 그리고 현재 처리 중인 청크의 토큰 수가 모두 합산되었다.

**표 6.** 청킹 방법별 토큰 비용 분석
**Table 6.** Token cost analysis across chunking methods

| Method | Total Chunks | Final Chunk Tokens | Additional Input Tokens |
|---|---|---|---|
| Fixed Chunking | 1621 | 231.7 ± 77.2 | - |
| Recursive Chunking | 1683 | 205.3 ± 70.7 | - |
| Semantic Chunking | 791 | 380.1 ± 482.4 | - |
| Proposed ($k_{past}=1$) | 1683 | 275.9 ± 73.1 | 471.1 ± 131.6 |
| Proposed ($k_{past}=3$) | 1683 | 278.0 ± 73.0 | 895.1 ± 249.8 |
| Proposed ($k_{past}=5$) | 1683 | 278.1 ± 72.6 | 1319.5 ± 374.1 |

제안 방법은 기반 모델인 Recursive Chunking과 동일한 분할 기준을 적용하여 전체 청크 수는 동일하게 유지되지만, 최종적으로 각 청크에 평균 약 70 토큰의 요약 문맥이 추가된다. 이 과정에서 참조하는 과거 청크 수가 증가할수록 문맥 요약 생성에 필요한 입력 토큰 수와 처리 시간이 함께 증가하므로, 초기 단계의 비용 부담이 커지는 한계가 있다. 그러나 청크 내부에 필수 서사 문맥을 사전에 보강함으로써 검색 단계에서의 문맥 단절을 완화할 수 있으며, 그 결과 적은 수의 검색 결과만으로도 기존 방법 대비 높은 성능을 나타낼 가능성이 있음을 표 4의 결과를 통해 알 수 있다.

따라서 제안 방법은 초기 비용 증가를 수반하지만, 검색 단계의 효율성과 성능 측면에서 보완 효과를 기대할 수 있는 방법으로 볼 수 있다.

## Ⅵ. 결 과

본 연구는 서사 데이터의 특성을 고려하여, 기존 청킹 방식으로 인해 발생하는 문맥 단절 문제를 완화하기 위해 순차적 문맥 주입 기반 청킹 기법을 제안하였다. 제안 방법은 현재 청크를 처리할 때 현재 청크 이전의 문맥을 요약한 정보를 선택적으로 결합함으로써, 기존 청킹 방식이 충분히 반영하지 못했던 서사적 연결성 저하와 핵심 정보 누락 문제를 보완한다.

NarrativeQA 데이터셋을 활용한 실험 결과, 제안 방법은 모든 검색 설정에서 기존 방법 대비 높은 Context Recall을 기록하였으며, 특히 적은 수의 검색 결과만으로도 정답 생성에 필요한 정보를 효과적으로 확보할 수 있음을 확인하였다. 이는 단순히 검색 범위를 확장하는 방식과 달리, 문맥 정보를 청크 수준에서 보강함으로써 검색 효율성을 향상시킬 수 있음을 보여준다. 또한 절제 연구를 통해, 과도한 과거 문맥을 누적하는 방식보다 현재 청크의 이해에 중요한 문맥 $k_{past}=1$를 중심으로 요약을 수행하는 것이 성능 향상에 보다 효과적임을 확인하였다. 이러한 결과는 서사 데이터에서 문맥 정보를 무분별하게 확장하기보다, 시간적 흐름과 인과적 연결성을 유지하는 것이 검색 정확도에 중요함을 시사한다.

다만, 본 연구의 실험 결과는 연산 비용 등의 제약으로 인해 제한된 수의 데이터를 기반으로 진행되었으므로, 일반화하기에는 한계가 존재한다. 향후 더 많은 데이터와 다양한 특성을 가진 데이터셋을 활용한 추가 실험을 통해 제안 방법의 일반화 성능을 검증하고 체계적으로 분석할 계획이다.

## 감사의 글

본 과제(결과물)은 2025년도 교육부 및 전북특별자치도의 재원으로 전북RISE센터의 지원을 받아 수행된 지역혁신중심 대학지원체계(RISE) 의 결과입니다(2025-RISE-13-JJU).

## 참고문헌

[1] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, ... and A. Mian, "A Comprehensive Overview of Large Language Models," *ACM Transactions on Intelligent Systems and Technology*, Vol. 16, No. 5, pp. 1-72, August 2025. https://doi.org/10.1145/3744746

[2] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, ...

and T. Liu, “A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions,” *ACM Transactions on Information Systems*, Vol. 43, No. 2, pp. 1-55, January 2025. https://doi.org/10.1145/3703155

[3] T. Yu, A. Xu, and R. Akkiraju, “In Defense of RAG in the Era of Long-Context Language Models,” arXiv:2409.01666, September 2024. https://doi.org/10.48550/arXiv.2409.01666

[4] B. Ni, Z. Liu, L. Wang, Y. Lei, Y. Zhao, X. Cheng, ... and T. Derr, “Towards Trustworthy Retrieval Augmented Generation for Large Language Models: A Survey,” arXiv:2502.06872, February 2025. https://doi.org/10.48550/arXiv.2502.06872

[5] C. Jeong, “Generative AI Service Implementation Using LLM Application Architecture: Based on RAG Model and LangChain Framework,” *Journal of Intelligence and Information Systems*, Vol. 29, No. 4, pp. 129-164, December 2023. https://doi.org/10.13088/jiis.2023.29.4.129

[6] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, ... and H. Wang, “Retrieval-Augmented Generation for Large Language Models: A Survey,” arXiv:2312.10997, December 2023. https://doi.org/10.48550/arXiv.2312.10997

[7] T. Chen, H. Wang, S. Chen, W. Yu, K. Ma, X. Zhao, ... and D. Yu, “Dense X Retrieval: What Retrieval Granularity Should We Use?,” in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing,* Miami, FL, pp. 15159-15177, November 2024. https://doi.org/10.18653/v1/2024.emnlp-main.845

[8] Anthropic. Contextual Retrieval [Internet]. Available: https://www.anthropic.com/engineering/contextual-retrieval.

[9] M. Abo El-Enen, S. Saad, and T. Nazmy, “A Survey on Retrieval-Augmentation Generation (RAG) Models For Healthcare Applications,” *Neural Computing and Applications*, Vol. 37, No. 33, pp. 28191-28267, October 2025. https://doi.org/10.1007/s00521-025-11666-9

[10] K. Zhong, B. Suleiman, A. Erradi, and S. Chen, “SemRAG: Semantic Knowledge-Augmented RAG for Improved Question-Answering,” arXiv:2507.21110, July 2025. https://doi.org/10.48550/arXiv.2507.21110

[11] C. Kiss, M. Nagy, and P. Szilágyi, “Max-Min Semantic Chunking of Documents for RAG Application,” *Discover Computing*, Vol. 28, No. 1, 117, June 2025. https://doi.org/10.1007/s10791-025-09638-7

[12] F. C. Mazzitelli and E. Zimeo, “Empowering RAG for Complex Documents with Paragraph-Based Chunking,” in *Proceedings of the 31st International DMS Conference on Visualization and Visual Languages*, San Francisco, CA, pp. 1-10, 2025.

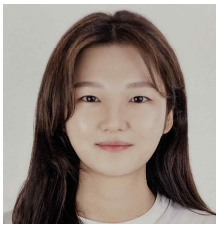
[13] R. Qu, R. Tu, and F. S. Bao, “Is Semantic Chunking Worth the Computational Cost?,” in *Findings of the Association for Computational Linguistics: NAACL 2025,* Albuquerque, New Mexico, pp. 2155-2177, April 2025. https://doi.org/10.18653/v1/2025.findings-naacl.114

[14] T. Kočiský, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, and E. Grefenstette, “The NarrativeQA Reading Comprehension Challenge,” *Transactions of the Association for Computational Linguistics*, Vol. 6, pp. 317-328, 2018. https://doi.org/10.1162/tacl_a_00023

[15] A. V. Duarte, J. D. S. Marques, M. Graça, M. Freire, L. Li, and A. L. Oliveira, “LumberChunker: Long-Form Narrative Document Segmentation,” in *Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024,* Miami, FL, pp. 6473-6486, November 2024. https://doi.org/10.18653/v1/2024.findings-emnlp.377

[16] I. S. Singh, R. Aggarwal, I. Allahverdiyev, M. Taha, A. Akalin, K. Zhu, and S. O’Brien, “ChunkRAG: Novel LLM-Chunk Filtering Method for RAG Systems,” arXiv:2410.19572, October 2024. https://doi.org/10.48550/arXiv.2410.19572

[17] A. Ammar, A. Koubaa, O. Nacar, and W. Boulila, “Optimizing Retrieval-Augmented Generation: Analysis of Hyperparameter Impact on Performance and Efficiency,” arXiv:2505.08445, 2025. https://doi.org/10.48550/arXiv.2505.08445



**이은주(Eun-Ju Lee)**

2022년 : 전주대학교 대학원 (공학석사)
2025년 : 전주대학교 대학원 (공학박사-인공지능학)
2025년～현 재: 전주대학교 인공지능연구소 연구원
※관심분야 : 인공지능(Artificial Intelligence), 데이터분석(Data Analysis) 등

**한동숭(Dong-Soong Han)**

1986년 : 서울대학교 대학원 (이학석사)
1992년 : 서울대학교 대학원 (이학박사-미분기하학)
1993년～현 재: 전주대학교 문화콘텐츠학과 교수
※관심분야 : 디지털 스토리텔링(Digital Storytelling), 콘텐츠 기획(Contents Planing), 생성형 인공지능(Generative AI) 등

**이성민(Sung-Min Lee)**

2006년～현 재: 세희에스앤디 대표

※관심분야 : 인공지능(Artificial Intelligence), 디지털 콘텐츠(Digital Contents), 지식그래프(Knowledge Graph), RAG 등

**박경수(Kyeong-Su Park)**

1989년 : 서울대학교 대학원 (이학석사-위상기하)

1997년 : 서울대학교 대학원 (이학박사-위상기하)

1998년～현 재: 전주대학교 게임콘텐츠학과 교수

※관심분야 : 기계학습(Machine Learning)