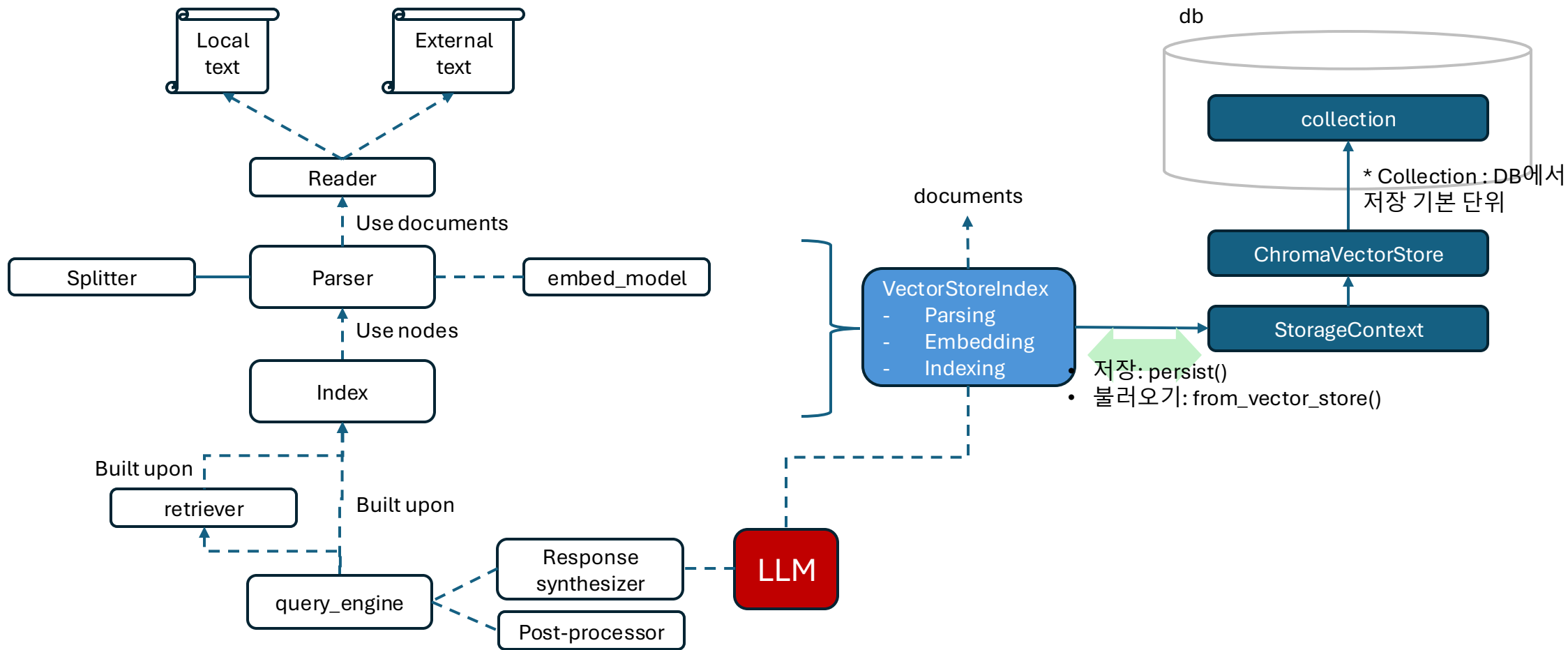


Deep Learning Application

Chapter 4. 텍스트 문서를 이용한 RAG 실습

LlamaIndex structure



4.1 개발환경구축

4.2 실습용 데이터 준비

- 프로젝트 폴더 생성
- 가상 환경 생성
- 실습용 데이터 준비
 - Txt, pdf, csv, hwp
 - 데이터 다운로드

```
urls=[
    "https://raw.githubusercontent.com/llama-index-tutorial/llama-index-tutorial/refs/head",
    "https://raw.githubusercontent.com/llama-index-tutorial/llama-index-tutorial/main/ch04",
    "https://raw.githubusercontent.com/llama-index-tutorial/llama-index-tutorial/refs/head",
    "https://raw.githubusercontent.com/llama-index-tutorial/llama-index-tutorial/main/ch04"
]

# 각 파일 다운로드
for url in urls:
    encoded_filename = url.split("/")[-1] # URL에서 파일명 추출
    decoded_filename = urllib.parse.unquote(encoded_filename) # 한글 파일명 복원
    response = requests.get(url)
    if response.status_code == 200:
        # 임시 파일명으로 저장
        temp_filename = "temp_download_file" + os.path.splitext(decoded_filename)[1]
        with open(temp_filename, 'wb') as f:
            f.write(response.content)
        os.rename(temp_filename, decoded_filename) # 파일명 변경
        print(f"완료: {decoded_filename} 다운로드 완료")
    else:
        print(f"오류: {url} 다운로드 실패 (상태 코드: {response.status_code})\n")
```

4.3 PDF 파일 다루기

4.3.1 데이터 준비

- 데이터 : 「디지털-빅데이터의 시대: 인문학의 새로운 역할에 대한 고찰」 PDF 연구 보고서
- 해당 보고서는 표지를 포함해 총 137페이지로 구성
- SimpleDirectoryReader 라이브러리를 활용
 - 라마인덱스에 내장된 데이터 커넥터
 - 예) SimpleDirectoryReader, GoogleDocsReader, NotionPageReader, SlackReader
 - 이러한 커넥터 들은 라마허브(Llamallub)를 통해 쉽게 설치하고 사용.

```
from llama_index.core import SimpleDirectoryReader

reader = SimpleDirectoryReader(input_files=["data/인문학의 새로운 역할에 대한 고찰.pdf"])
documents = reader.load_data()
```

- input_files는 해당 디렉터리 안에 있는 파일 중 원하는 파일만 로드하도록 지정하는 옵션
 - Vs. input_dir은 지정한 디렉터리에서 라마인덱스가 로딩할 수 있는 모든 파일을 가져오는 옵션.

```
print(nodes[3].get_content())
```

```
class SimpleDirectoryReader(
    input_dir: Path | str | None = None,
    input_files: list | None = None,
```

```
Document(
    id_='81167921-827a-4bda-9972-739cf91400b3',
    embedding=None,
    metadata={
        'page_label': '4',
        'file_name': '인문학의 새로운 역할에 대한 고찰.pdf',
        ...
    },
    ...
    text='연구요약\n인간 사회의 전 분야를 쓰나미처럼 ... 새',
    ...
)
```

4.3 PDF 파일 다루기

4.3.2 텍스트 분할

- 페이지별로 Document로 로드될 경우, 문서의 마지막 문장이 완결된 형태로 저장되지 않는 경우가 생김.
- 또, 하나의 document에 여러 개의 의미가 포함되면 사용자의 질문에 대해 최적의 답변을 제공 하는 데 한계가 있음.
- 텍스트 분할 (chunking):
 - 긴 문장을 짧게 나누어 노드에 담는 작업.
 - 앞서 보고서의 경우, 각 도큐먼트가 페이지별로 분할된 내용을 담고 있음. 그러나 하나의 도큐먼트 안에 여러 의미가 뒤섞여 있을 수 있으므로, 의미의 일관성을 유지하며 인덱싱을 하려면 텍스트를 별도로 분할하는 절차가 필요
 - 잘 분할된 데이터 는 RAG의 답변 성능에 큰 영향을 미치며, 답변 정확도 뿐 아니라 답변 속도에도 영향을 줌
- Chunking 방법
 - 토큰 단위 분할
 - 의미 단위 분할
 - 문장 단위 분할

4.3 PDF 파일 다루기

4.3.2 텍스트 분할

- 토큰 단위 분할
- TokenTextSplitter 클래스
 - 텍스트를 토큰 단위로 나눔
 - 청크 크기(chunk size)는 각 청크의 최대 토큰 수 지정, chunk_size=1024
 - 청크 오버랩(chunk_overlap)은 연속된 청크 간의 중첩 토큰 수를 지정, chunk_overlap=20
- [실습] “ch04_practice.ipynb”

```
# TokenTextSplitter 설정하기
splitter = TokenTextSplitter(chunk_size=1024, chunk_overlap=20)

# TokenTextSplitter 적용 후 노드에 담기
nodes = splitter.get_nodes_from_documents(documents)
```

```
print(len(nodes))
```

```
print(nodes[3].get_content())
```

Vs. `split_text(text: str) -> List[str]`

TokenTextSplitter *inherits* `get_nodes_from_documents(...)` from the `NodeParser` interface

연구요약

인간 사회의 전 분야를 쓰나미처럼 휩쓸고 들어오는 디지털화는 개인의 생활 방식, 개인들끼리의 교류 방식, 더 나아가 사회가 작동하는 방식을 총체적이고 근본적으로 변화시키고 있는 상황이다. 우리는 현재 디지털을 사용하는 선을 넘어서서 디지털 가운데서 살아가고 있다. 디지털이라는 매체가 인간 삶의 전반적 틀을 크게 좌우하는 이러한 상황은 여러 가지 전례 없는 현상들을 낳고 있으며, 이러한 변화가 긍정적일지 부정적일지는 현재의 우리가 어떻게 대처하느냐에 달려있을 수밖에 없다. 평자에 따라서는 농업혁명, 산업혁명 이상의 역사적인 전환점이라고도 평가되고 있는 현재 시점은 곧 새로운 사회구조의 원리와 더불어 새로운 방식의 학문이 출현하는 시점인 듯하다.

이 글에서는 먼저 개인정보의 문제와 젠더의 문제에 집중하여 디지털-빅데이터 시대가 낳은 사회적 변화의 양상을 조망하고자 했다. 인간의 생활 전반이 디지털에서 흔적으로 남고 있는 현실은 그 흔적, 즉 “trace data”를 그 자체로 “상품”으로 간주하려는 경향과 결부되고 있다. 디지털 사회로의 진입이 이미 이루어졌고 후퇴가 불가능할 뿐만 아니라 디지털-빅데이터 사회가 갖는 장점이 매우 큰 것을 고려할 때 “데이터 공동체”, 혹은 “연대적인 데이터 저장” 등 현재 사회학에서 제시되고 있는 방법들에 대해 성찰함으로써 개인데이터가 “상품”으로 직진하지 않도록 하기 위한 구체적인 방법을 찾아내기 위해 노력할 필요가 절실히 보인다.

젠더의 경우 디지털-빅데이터 시대가 사회 진보의 측면에서 가능성과 한계를 동시에 안고 있음을 선명하게 보여주고 있다. 한편으로 디지털 전환 자체가 젠더에 따라 다르게 경험되고 있고, 디지털 세계에서도 기존의 남성중심적인 구조가 재생산되는 경향이 분명한 것이 현실이다. 그러나 다른 한편으로 빅데이터는 2000년대 이후 정체상태에 머물러 있는 양성평등에 한걸음 더 가까이 갈 수 있는 가능성을 내포하고 있기도 하다. 젠더 격차의 문제를 빅데이터를 통해서 입체적으로 보여줄 수가 있고, 그 결과로 구체적이고 적극적인 개입이 가능하다는 것이 여러 가지 경로를 통해서 드러나고 있는 것이다. 가령 유엔이 클린턴 재단 등 거대 민간 재단들과 공동설립한 Data2X의 경우 기존의 데이터들을 모으는 데에서 더 나아가 이 데이터들을 해체함으로써 결핍된 데이터, 즉 여성에 대한 데이터를 채우는 것을 목표로 삼고 있다. 이와 같이 빅데이터의 입체성은 인간의 사회가 젠더의 측면에서 얼마나 기울어진 운동장인지를 보다 더 명징하게 보여주고 있고, 디지털을 통해 가능해진 과거와 비교할 수 없는 정도로 진전된 국제협력을 통해서 과거보다 더 적극적이고 구체적인 방식의 대응을 가능케 할 가능성을 내포하고 있는 것이다.

디지털화가 내포하고 있는 사회 전반의 변화는 분과학문 단위에서 접근할 수 있는 범위를 넘어서 있다. 이러한 문제적 상황에 대해 인문학과 사회과학 분야가 각각 어떻게 대응하고 있는지를 II부와 III부에서 살펴보았다. II부에서는 먼저 디지털인문학이 출현함으로써 새

연구요약 인간 사회의 전 분야를 쓰나미처럼 휩쓸고 들어오는 디지털화는 개인의 생활 방식, 개인들끼리의 교류 방식, 더 나아가 사회가 작동하는 방식을 총체적이고 근본적으로 변화시키고 있는 상황이다. 우리는 현재 디지털을 사용하는 선을 넘어서서 디지털 가운데서 살아가고 있다. 디지털이라는 매체가 인간 삶의 전반적 틀을 크게 좌우하는 이러한 상황은 여러 가지 전례 없는 현상들을 낳고 있으며, 이러한 변화가 긍정적일지 부정적일지는 현재의 우리가 어떻게 대처하느냐에 달려있을 수밖에 없다. 평자에 따라서는 농업혁명, 산업혁명 이상의 역사적인 전환점이라고도 평가되고 있는 현재 시점은 곧 새로운 사회구조의 원리와 더불어 새로운 방식의 학문이 출현하는 시점인 듯하다. 이 글에서는 먼저 개인정보의 문제와 젠더의 문제에 집중하여 디지털-빅데이터 시대가 낳은 사회적 변화의 양상을 조망하고자 했다. 인간의 생활 전반이 디지털에서 흔적으로 남고 있는 현실은 그 흔적, 즉 “trace data”를 그 자체로 “상품”으로 간주하려는 경향과 결부되고 있다. 디지털 사회로의 진입이 이미 이루어졌고 후퇴가 불가능할 뿐만 아니라 디지털-빅데이터 사회가 갖는 장점이 매우 큰 것을 고려할 때 “데이터 공동체”, 혹은 “연대적인 데이터 저장” 등 현재 사회학에서 제시되고 있는 방법들에 대해 성찰함으로써 개인데이터가 “상품”으로 직진하지 않도록 하기 위한 구체적인 방법을 찾아내기 위해 노력할 필요가 절실히 보인다. 젠더의 경우 디지털-빅데이터 시대가 사회 진보의 측면에서 가능성과 한계를 동시에 안고 있음을 선명하게 보여주고 있다. 한편으로 디지털 전환 자체가 젠더에 따라 다르게 경험되고 있고, 디지털 세계에서도 기존의 남성중심적인 구조가 재생산되는 경향이 분명한 것이 현실이다. 그러나 다른 한편으로 빅데이터는 2000년대 이후 정체상태에 머물러 있는 양성평등에 한걸음 더 가까이 갈 수 있는 가능성을 내포하고 있기도 하다. 젠더 격차의 문제를 빅데이터를 통해서 입체적으로 보여줄 수가 있고, 그 결과로 구체적이고 적극적인 개입이 가능하다는 것이 여러 가지 경로를 통해서 드러나고 있는 것이다. 가령

4.3 PDF 파일 다루기

4.3.2 텍스트 분할

- 의미 단위 분할(Semantic Splitting):
 - 임베딩 유사성을 기반으로 하나의 의미를 지닌 문장 단위를 찾아 분할하는 방식
 - 의미가 비슷한 텍스트는 수학적으로 유사한 임베딩 값을 가짐. 예) "고양이"와 "개"라는 단어는 임베딩 공간에서 "고양이"와 "하늘"보다 가까운 위치에 있음
 - 의미 단위 분할을 적용하면 분할할 위치가 고정적이지 않고 유동적으로 결정. 청크 분할 위치가 달라지기도 함. 분할 후 노드의 길이와 노드의 총 개수가 다른 경우가 발생
- Llama-index-embeddings-openai를 사용 (설치)

```
# embed_model 인스턴스 만들기
embed_model = OpenAIEmbedding()

# semantic_splitter 설정하기
semantic_splitter = SemanticSplitterNodeParser(
    buffer_size=10,
    breakpoint_percentile_threshold=95,
    embed_model=embed_model
)

# semantic_splitter 적용 후 노드에 담기
nodes_semantic=semantic_splitter.get_nodes_from_documents(documents)
```

- `buffer_size`: 의미적 유사성을 평가할 때, 그룹화할 문장 수. 값이 작으면 메모리 사용량은 줄지만, 데이터 처리 속도가 느려짐. 값이 크면 메모리는 더 많이 사용하지만 성능이 향상.
- `breakpoint_percentile_threshold`: 95일 경우, 문장 간 의미 차이를 계산한 뒤, 그 차이가 95% 이상인 지점에서 분할하라고 지정

연구요약

인간 사회의 전 분야를 쓰나미처럼 휩쓸고 들어오는 디지털화는 개인의 생활 방식, 개인들끼리의 교류 방식, 더 나아가 사회가 작동하는 방식을 총체적이고 근본적으로 변화시키고 있는 상황이다. 우리는 현재 디지털을 사용하는 선을 넘어서서 디지털 가운데서 살아가고 있다. 디지털이라는 매체가 인간 삶의 전반적 틀을 크게 좌우하는 이러한 상황은 여러 가지 전례 없는 현상들을 낳고 있으며, 이러한 변화가 긍정적일지 부정적일지는 현재의 우리가 어떻게 대처하느냐에 달려있을 수밖에 없다. 평자에 따라서는 농업혁명, 산업혁명 이상의 역사적인 전환점이라고도 평가되고 있는 현재 시점은 곧 새로운 사회구조의 원리와 더불어 새로운 방식의 학문이 출현하는 시점인 듯하다.

이 글에서는 먼저 개인정보의 문제와 젠더의 문제에 집중하여 디지털-빅데이터 시대가 낳은 사회적 변화의 양상을 조망하고자 했다. 인간의 생활 전반이 디지털에서 흔적으로 남고 있는 현실은 그 흔적, 즉 “trace data”를 그 자체로 “상품”으로 간주하려는 경향과 결부되고 있다. 디지털 사회로의 진입이 이미 이루어졌고 후퇴가 불가능할 뿐만 아니라 디지털-빅데이터 사회가 갖는 장점이 매우 큰 것을 고려할 때 “데이터 공동체”, 혹은 “연대적인 데이터 저장” 등 현재 사회학에서 제시되고 있는 방법들에 대해 성찰함으로써 개인데이터가 “상품”으로 직진하지 않도록 하기 위한 구체적인 방법을 찾아내기 위해 노력할 필요가 절실히 보인다.

젠더의 경우 디지털-빅데이터 시대가 사회 진보의 측면에서 가능성과 한계를 동시에 안고 있음을 선명하게 보여주고 있다. 한편으로 디지털 전환 자체가 젠더에 따라 다르게 경험되고 있고, 디지털 세계에서도 기존의 남성중심적인 구조가 재생산되는 경향이 분명한 것이 현실이다. 그러나 다른 한편으로 빅데이터는 2000년대 이후 정체상태에 머물러 있는 양성평등에 한걸음 더 가까이 갈 수 있는 가능성을 내포하고 있기도 하다. 젠더 격차의 문제를 빅데이터를 통해서 입체적으로 보여줄 수가 있고, 그 결과로 구체적이고 적극적인 개입이 가능하다는 것이 여러 가지 경로를 통해서 드러나고 있는 것이다. 가령 유엔이 클린턴 재단 등 거대 민간 재단들과 공동설립한 Data2X의 경우 기존의 데이터들을 모으는 데에서 더 나아가 이 데이터들을 해체함으로써 결핍된 데이터, 즉 여성에 대한 데이터를 채우는 것을 목표로 삼고 있다. 이와 같이 빅데이터의 입체성은 인간의 사회가 젠더의 측면에서 얼마나 기울어진 운동장인지를 보다 더 명징하게 보여주고 있고, 디지털을 통해 가능해진 과거와 비교할 수 없는 정도로 진전된 국제협력을 통해서 과거보다 더 적극적이고 구체적인 방식의 대응을 가능케 할 가능성을 내포하고 있는 것이다.

디지털화가 내포하고 있는 사회 전반의 변화는 분과학문 단위에서 접근할 수 있는 범위를 넘어서 있다. 이러한 문제적 상황에 대해 인문학과 사회과학 분야가 각각 어떻게 대응하고 있는지를 II부와 III부에서 살펴보았다. II부에서는 먼저 디지털인문학이 출현함으로써 새

연구요약 인간 사회의 전 분야를 쓰나미처럼 휩쓸고 들어오는 디지털화는 개인의 생활 방식, 개인들끼리의 교류 방식, 더 나아가 사회가 작동하는 방식을 총체적이고 근본적으로 변화시키고 있는 상황이다. 우리는 현재 디지털을 사용하는 선을 넘어서서 디지털 가운데서 살아가고 있다. 디지털이라는 매체가 인간 삶의 전반적 틀을 크게 좌우하는 이러한 상황은 여러 가지 전례 없는 현상들을 낳고 있으며, 이러한 변화가 긍정적일지 부정적일지는 현재의 우리가 어떻게 대처하느냐에 달려있을 수밖에 없다. 평자에 따라서는 농업혁명, 산업혁명 이상의 역사적인 전환점이라고도 평가되고 있는 현재 시점은 곧 새로운 사회구조의 원리와 더불어 새로운 방식의 학문이 출현하는 시점인 듯하다.이 글에서는 먼저 개인정보의 문제와 젠더의 문제에 집중하여 디지털-빅데이터 시대가 낳은 사회적 변화의 양상을 조망하고자 했다. 인간의 생활 전반이 디지털에서 흔적으로 남고 있는 현실은 그 흔적, 즉 “trace data”를 그 자체로 “상품”으로 간주하려는 경향과 결부되고 있다. 디지털 사회로의 진입이 이미 이루어졌고 후퇴가 불가능할 뿐만 아니라 디지털-빅데이터 사회가 갖는 장점이 매우 큰 것을 고려할 때 “데이터 공동체”, 혹은 “연대적인 데이터 저장” 등 현재 사회학에서 제시되고 있는 방법들에 대해 성찰함으로써 개인데이터가 “상품”으로 직진하지 않도록 하기 위한 구체적인 방법을 찾아내기 위해 노력할 필요가 절실히 보인다.젠더의 경우 디지털-빅데이터 시대가 사회 진보의 측면에서 가능성과 한계를 동시에 안고 있음을 선명하게 보여주고 있다. 한편으로 디지털 전환 자체가 젠더에 따라 다르게 경험되고 있고, 디지털 세계에서도 기존의 남성중심적인 구조가 재생산되는 경향이 분명한 것이 현실이다. 그러나 다른 한편으로 빅데이터는 2000년대 이후 정체상태에 머물러 있는 양성평등에 한걸음 더 가까이 갈 수 있는 가능성을 내포하고 있기도 하다. 젠더 격차의 문제를 빅데이터를 통해서 입체적으로 보여줄 수가 있고, 그 결과로 구체적이고 적극적인 개입이 가능하다는 것이 여러 가지 경로를 통해서 드러나고 있는 것이다. 가령 유엔이 클린턴 재단 등 거대 민간 재단들과 공동설립한 Data2X의 경우 기존의 데이터들을 모으는 데에서 더 나아가 이 데이터들을 해체함으로써 결핍된 데이터, 즉 여성에 대한 데이터를 채우는 것을 목표로 삼고 있다. 이와 같이 빅데이터의 입체성은 인간의 사회가 젠더의 측면에서 얼마나 기울어진 운동장인지를 보다 더 명징하게 보여주고 있고, 디지털을 통해 가능해진 과거와 비교할 수 없는 정도로 진전된 국제협력을 통해서 과거보다 더 적극적이고 구체적인 방식의 대응을 가능케 할 가능성을 내포하고 있는 것이다.디지털화가 내포하고 있는 사회 전반의 변화는 분과학문 단위에서 접근할 수 있는 범위를 넘어서 있다.

4.3 PDF 파일 다루기

4.3.2 텍스트 분할

- 문장 단위 분할(Sentence Splitting) : 문장의 경계를 존중하면서 텍스트를 분리하는 방식
- SentencespLitter 클래스를 임 포트하고, chunk_size를 1024토큰, chunk_overlap을 20토큰으로 설정

```
from llama_index.core.node_parser import SentenceSplitter
```

```
# semantic_splitter 설정하기  
splitter = SentenceSplitter(  
    chunk_size=1024,  
    chunk_overlap=20,  
)
```

```
# semantic_splitter 적용 후 노드에 담기  
nodes_sentence = splitter.get_nodes_from_documents(documents)
```

```
print(len(nodes_sentence))
```

연구요약

인간 사회의 전 분야를 쓰나미처럼 휩쓸고 들어오는 디지털화는 개인의 생활 방식, 개인들끼리의 교류 방식, 더 나아가 사회가 작동하는 방식을 총체적이고 근본적으로 변화시키고 있는 상황이다. 우리는 현재 디지털을 사용하는 선을 넘어서서 디지털 가운데서 살아가고 있다. 디지털이라는 매체가 인간 삶의 전반적 틀을 크게 좌우하는 이러한 상황은 여러 가지 전례 없는 현상들을 낳고 있으며, 이러한 변화가 긍정적일지 부정적일지는 현재의 우리가 어떻게 대처하느냐에 달려있을 수밖에 없다. 평자에 따라서는 농업혁명, 산업혁명 이상의 역사적인 전환점이라고도 평가되고 있는 현재 시점은 곧 새로운 사회구조의 원리와 더불어 새로운 방식의 학문이 출현하는 시점인 듯하다.

이 글에서는 먼저 개인정보의 문제와 젠더의 문제에 집중하여 디지털-빅데이터 시대가 낳은 사회적 변화의 양상을 조망하고자 했다. 인간의 생활 전반이 디지털에서 흔적으로 남고 있는 현실은 그 흔적, 즉 “trace data”를 그 자체로 “상품”으로 간주하려는 경향과 결부되고 있다. 디지털 사회로의 진입이 이미 이루어졌고 후퇴가 불가능할 뿐만 아니라 디지털-빅데이터 사회가 갖는 장점이 매우 큰 것을 고려할 때 “데이터 공동체”, 혹은 “연대적인 데이터 저장” 등 현재 사회학에서 제시되고 있는 방법들에 대해 성찰함으로써 개인데이터가 “상품”으로 직진하지 않도록 하기 위한 구체적인 방법을 찾아내기 위해 노력할 필요가 절실히 보인다.

젠더의 경우 디지털-빅데이터 시대가 사회 진보의 측면에서 가능성과 한계를 동시에 안고 있음을 선명하게 보여주고 있다. 한편으로 디지털 전환 자체가 젠더에 따라 다르게 경험되고 있고, 디지털 세계에서든 기존의 남성중심적인 구조가 재생산되는 경향이 분명한 것이 현실이다. 그러나 다른 한편으로 빅데이터는 2000년대 이후 정체상태에 머물러 있는 양성평등에 한걸음 더 가까이 갈 수 있는 가능성을 내포하고 있기도 하다. 젠더 격차의 문제를 빅데이터를 통해서 입체적으로 보여줄 수가 있고, 그 결과로 구체적이고 적극적인 개입이 가능하다는 것이 여러 가지 경로를 통해서 드러나고 있는 것이다. 가령 유엔이 클린턴 재단 등 거대 민간 재단들과 공동설립한 Data2X의 경우 기존의 데이터들을 모으는 데에서 더 나아가 이 데이터들을 해체함으로써 결핍된 데이터, 즉 여성에 대한 데이터를 채우는 것을 목표로 삼고 있다. 이와 같이 빅데이터의 입체성은 인간의 사회가 젠더의 측면에서 얼마나 기울어진 운동장인지를 보다 더 명징하게 보여주고 있고, 디지털을 통해 가능해진 과거와 비교할 수 없는 정도로 진전된 국제협력력을 통해서 과거보다 더 적극적이고 구체적인 방식의 대응을 가능케 할 가능성을 내포하고 있는 것이다.

디지털화가 내포하고 있는 사회 전반의 변화는 분과학문 단위에서 접근할 수 있는 범위를 넘어서 있다. 이러한 문제적 상황에 대해 인문학과 사회과학 분야가 각각 어떻게 대응하고 있는지를 II부와 III부에서 살펴보았다. II부에서는 먼저 디지털인문학이 출현함으로써 새

연구요약 인간 사회의 전 분야를 쓰나미처럼 휩쓸고 들어오는 디지털화는 개인의 생활 방식, 개인들끼리의 교류 방식, 더 나아가 사회가 작동하는 방식을 총체적이고 근본적으로 변화시키고 있는 상황이다. 우리는 현재 디지털을 사용하는 선을 넘어서서 디지털 가운데서 살아가고 있다. 디지털이라는 매체가 인간 삶의 전반적 틀을 크게 좌우하는 이러한 상황은 여러 가지 전례 없는 현상들을 낳고 있으며, 이러한 변화가 긍정적일지 부정적일지는 현재의 우리가 어떻게 대처하느냐에 달려있을 수밖에 없다. 평자에 따라서는 농업혁명, 산업혁명 이상의 역사적인 전환점이라고도 평가되고 있는 현재 시점은 곧 새로운 사회구조의 원리와 더불어 새로운 방식의 학문이 출현하는 시점인 듯하다.이 글에서는 먼저 개인정보의 문제와 젠더의 문제에 집중하여 디지털-빅데이터 시대가 낳은 사회적 변화의 양상을 조망하고자 했다. 인간의 생활 전반이 디지털에서 흔적으로 남고 있는 현실은 그 흔적, 즉 “trace data”를 그 자체로 “상품”으로 간주하려는 경향과 결부되고 있다. 디지털 사회로의 진입이 이미 이루어졌고 후퇴가 불가능할 뿐만 아니라 디지털-빅데이터 사회가 갖는 장점이 매우 큰 것을 고려할 때 “데이터 공동체”, 혹은 “연대적인 데이터 저장” 등 현재 사회학에서 제시되고 있는 방법들에 대해 성찰함으로써 개인데이터가 “상품”으로 직진하지 않도록 하기 위한 구체적인 방법을 찾아내기 위해 노력할 필요가 절실히 보인다.젠더의 경우 디지털-빅데이터 시대가 사회 진보의 측면에서 가능성과 한계를 동시에 안고 있음을 선명하게 보여주고 있다. 한편으로 디지털 전환 자체가 젠더에 따라 다르게 경험되고 있고, 디지털 세계에서든 기존의 남성중심적인 구조가 재생산되는 경향이 분명한 것이 현실이다. 그러나 다른 한편으로 빅데이터는 2000년대 이후 정체상태에 머물러 있는 양성평등에 한걸음 더 가까이 갈 수 있는 가능성을 내포하고 있기도 하다.

4.3 PDF 파일 다루기

4.3.3 인덱싱

- 인덱싱: 데이터를 구조화하여 빠르게 검색할 수 있도록 하는 과정
- 라마인덱스는 문서를 노드로 분할한 뒤, 각 노드의 의미를 벡터 임베딩으로 변환해 저장
- 라마인덱스는 다양한 유형의 인덱스를 지원
- 벡터 스토어 인덱스를 활용 해 실습을 진행
 - 벡터 스토어 인덱스는 각 노드의 텍스트를 벡터 임베딩으로 변환하고, 쿼리가 들어오면 쿼리 또한 벡터 임베딩으로 변환한 뒤, 이 두 임베딩 간의 유사도를 계산해 가장 유사한 상위 k개의 노드를 반환

```
from llama_index.core import VectorStoreIndex
index = VectorStoreIndex(nodes_sentence)
```

```
print(index)
```

```
query_engine = index.as_query_engine(similarity_top_k=5)
```

```
response = query_engine.query("디지털 인문학은 인문학을 어떤 방식과 관점으로 다루는 학문인지 알려줘")
print(response)
```

```
print([response.source_nodes[0].get_content()])
```

← 쿼리와 유사한 상위문서 5개를 사용하도록 설정.이중 0번 소스 내용을 확인하는 코드

4.4 텍스트 파일 다루기

4.4.1 기본 RAG 실습

- 텍스트(TXT) 파일을 다루는 방식은 PDF 파일을 다루는 방식과 같음
- 폴 그레이엄의 에세이를 실습에 사용

```
from llama_index.core import SimpleDirectoryReader

reader = SimpleDirectoryReader(input_files=["data/paul_graham_essay.txt"])
documents = reader.load_data()
```

```
print(len(documents))
```

- 의미기반 분할 기법 (Semantic Chunking)

```
from llama_index.embeddings.openai import OpenAIEmbedding
from llama_index.llms.openai import OpenAI
from llama_index.core import Settings

llm = OpenAI(model="gpt-4o", temperature=0.2)
embed_model = OpenAIEmbedding()

Settings.llm = llm
Settings.embed_model = embed_model
```

```
model='gpt-4o'
model_name='text-embedding-ada-002'
```

4.4 텍스트 파일 다루기

4.4.1 기본 RAG 실습

```
from llama_index.core.node_parser import SemanticSplitterNodeParser

semantic_splitter = SemanticSplitterNodeParser(
    buffer_size=10,
    breakpoint_percentile_threshold=80,
    embed_model=embed_model
)
txt_nodes_semantic=semantic_splitter.get_nodes_from_documents(documents)
```

```
print("노드 개수:", len(txt_nodes_semantic))
print(txt_nodes_semantic[10].get_content())
```

```
from llama_index.core import VectorStoreIndex

txt_index = VectorStoreIndex(txt_nodes_semantic)
```

```
#쿼리 엔진 설정하기
txt_query_engine = txt_index.as_query_engine(similarity_top_k=5)

#쿼리 실행하기
txt_response = txt_query_engine.query("저자는 유년 시절에 무엇에 열정을 쏟았어? 자세히 설명해 줘")
print(txt_response)
```

Similarity_top_k=5로 설정, 쿼리와 유사한 문서를 최대 5개까지

4.4 텍스트 파일 다루기

4.4.2 인덱스 저장: 크로마 사용하기

- 파일로 로드하여 인덱싱하고, 이를 바탕으로 간단한 질의응답 과정을 실습, 이 방식은 프로그램을 실행할 때마다 문서를 새로 로드하고 임베딩을 생성해야 하므로 비효율적.
- → 벡터 저장소: 벡터 저장소는 한 번 생성된 임베딩을 저장해 두고, 필요할 때마다 불러옴. 예: 크로마 벡터 저장소
- 임베딩 저장하기

```
documents = SimpleDirectoryReader(input_files=["data/paul_graham_essay.txt"]).load_data()
```

```
db = chromadb.PersistentClient(path="./chroma_db_test")
```

```
chroma_collection = db.get_or_create_collection("quickstart_v.1")
```

- Collection “quickstart_v.1”을 생성

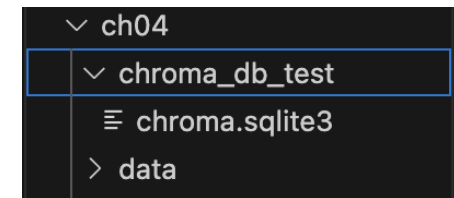
```
vector_store = ChromaVectorStore(chroma_collection=chroma_collection)  
storage_context = StorageContext.from_defaults(vector_store=vector_store)
```

- Storage_context 설정, 데이터의 저장과 검색을 관리하는 구성요소

```
index = VectorStoreIndex.from_documents(  
    documents, storage_context=storage_context  
)
```

StorageContext is a container that can hold multiple storage backends:

- vector_store
- docstore (raw documents)
- index_store (index metadata)



Vs.

```
# Chroma 벡터 스토어 생성  
vector_store = ChromaVectorStore(chroma_collection=collection)  
  
# LlamaIndex의 VectorStoreIndex 생성  
index = VectorStoreIndex.from_documents(nodes, vector_store=vector_store)  
  
# 쿼리 엔진 생성 (기본적인 검색 + 답변 생성 기능 활성화)  
! query_engine = index.as_query_engine()
```

VectorStoreIndex creates a StorageContext under the hood with that vector_store.

4.4 텍스트 파일 다루기

4.4.2 인덱스 저장: 크로마 사용하기

```
query_engine = index.as_query_engine()
response = query_engine.query("저자는 유년 시절에 어떤 작업에 열중했어? 자세히 설명해 줘")
print(response)
```

- 벡터 스토어로 생성한 임베딩을 재사용
- 저장된 임베딩을 불러올수 있음.

4.4 텍스트 파일 다루기

4.4.2 인덱스 저장: 크로마 사용하기

- 앞서 만들어둔 quickstart_v.1 collection을 불러옴

```
# 크로마 클라이언트 초기화
db = chromadb.PersistentClient(path="./chroma_db_test")

# 컬렉션 호출
chroma_collection = db.get_or_create_collection("quickstart_v.1")
```

```
# 벡터 스토어 설정
vector_store = ChromaVectorStore(chroma_collection=chroma_collection)
storage_context = StorageContext.from_defaults(vector_store=vector_store)

# 저장했던 인덱스 로드
index = VectorStoreIndex.from_vector_store(
    vector_store, storage_context=storage_context
)
```

```
query_engine = index.as_query_engine()
response = query_engine.query("저자에게 프로그래밍은 어떤 의미인가요")
print(response)
```

What is StorageContext?

Think of it as a **container** (or configuration object) that bundles together the storage backends used by your index. It can hold:

- **vector_store** → where embeddings go (e.g. Pinecone, Chroma, Weaviate, FAISS).
- **docstore** → where raw documents/nodes are saved (e.g. JSON, MongoDB, SQL).
- **index_store** → where index metadata lives (e.g. relationships between nodes, summary info).
- **graph_store** (if using graph indexes).

By wrapping these together, the index doesn't have to care *where* data is stored — it just calls the storage context.

4.5 CSV 파일 다루기

- CSV 파일에 특화된 리더인 CSVReader를 함께 사용.
- 데이터: 경찰청 미래치안정책국의 2023년 12월 31일 기준 범죄 발생 장소별 통계
- SimpleDirectoryReader 만 사용했을 경우

```
from llama_index.core import SimpleDirectoryReader

reader = SimpleDirectoryReader(input_files=["data/범죄 발생 장소별 통계_2023.csv"])
documents = reader.load_data()

print("documents 개수:", len(documents))
print(documents[0])
```

documents 개수: 1 Doc ID: 9679de78-27cd-40c0-9596-e54154578493 Text:
강력범죄,살인미수,48,66,38,16,6,0,1,4,8,0,0,0,0,0,2,0,2,17,0,0,0,0,2,1,4,3,0,0,0,0,0,0,6,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,8,1,7,2,2,0,0,0,0,1,29,4
강력범죄,살인미수등,61,67,45,19,13,1,1,16,40,0,1,3,1,2,0,0,12,13,3,1,1,0,
14,1,10,6,1,1,0,0,0,0,1,5,0,0,3,0,0,0,1,0,0,1,0,3,0,0,1,1,14,3,15,1,3,0,3,0,3,1,93,7
강력범죄,강도,36,40,29,42,9,1,1,18,42,3,4,13,35,3,0,3,23,45,1,1,...

→ document 하나, CSV 파일의 헤더가 사라짐

4.5 CSV 파일 다루기

```
from llama_index.core import SimpleDirectoryReader
from llama_index.readers.file import CSVReader

parser = CSVReader()
file_extractor = {".csv": parser}
```

```
documents = SimpleDirectoryReader(
    input_files=["data/범죄 발생 장소별 통계_2023.csv"],
    file_extractor=file_extractor
).load_data()
```

```
print("documents 개수:", len(documents))
print(documents[0])
```

documents 개수: 1 Doc ID: 0dc0e4cf-0975-4c57-9886-9fe6772eee78 Text: "범죄대분류, 범죄중분류, 단독주택_다가구_다중, 아파트, 다세대_연립, 오피스텔_원룸, 기타거주시설_기숙사 등, 고속도로, 자동차 전용도로, 일반도로, 통행로_보도_골목길, 백화점, 대형할인점, 슈퍼마켓_소매점, 편의점, 시장_노점, 창고_매장창고한정, 무인상점, 기타상점, 숙박업소_호텔_모텔_여관, 목욕탕_찜질방_사우나, 이발소_미용실, 마사지업소, 공중위생업소_기타, 음식점, 카페, 주점, 단란_유흥주점_나이트_클럽_카바레, 버스터미널_정류소, 지하철역_전철역, 기차역, 여객선터미널, 공항, 버스, 택시, 자가용자동차, 지하철_전철, 기차, 선박, 비행기, 교통수단내_기타, 공연장_극장, 체육시설, 공원_놀..."

- → document 1개, 헤더 부분 로드

4.5 CSV 파일 다루기

- 의미기반 분할

```
from llama_index.core.node_parser import SemanticSplitterNodeParser
from llama_index.embeddings.openai import OpenAIEmbedding

embed_model = OpenAIEmbedding()
semantic_splitter = SemanticSplitterNodeParser(
    buffer_size=10,
    breakpoint_percentile_threshold=80,
    embed_model=embed_model
)

csv_nodes_semantic=semantic_splitter.get_nodes_from_documents(documents)
```

```
print("노드 개수:", len(csv_nodes_semantic))
print(csv_nodes_semantic[0].get_content())
```

[illegible]

```
from llama_index.core import VectorStoreIndex

csv_index = VectorStoreIndex(csv_nodes_semantic)
```

```
csv_query_engine = csv_index.as_query_engine(similarity_top_k=5)

csv_response = csv_query_engine.query("절도범죄가 가장 많이 일어난 장소는 어디야? 수지도 알려줘")
print(csv_response)
```

4.5 HWP 파일 다루기