Code Paper https://doi.org/10.23287/KJDH.2025.2.1.4

작가 박경리 관련 일간지 기사 데이터 구축 사례

A Case Study on Building a Newspaper Article Dataset Related to Author Pak Kyongni

신정은(Shin, Jeong-Eun)*
0009-0007-0726-6820

목차

초록

이 연구는 작가 박경리와 소설 『토지』의 수용을 대중의 문학 향유 행위로 보고, 대중적 문학 수용에 큰 영향을 미친 신문 기사를 연구 가능한 데이터로 구축한 사례를 소개한다. 연구자는 1955년 8월부터 2024년 8월까지 69년간 24종의 신문에 게재된 박경리와 『토지』 관련 기사를 AI 챗봇의 도움을 받아 수집·전처리하는 코드를 개발하여 데이터로 구축하였다.
이 사례는 인문학 연구자들이 디지털 도구와의 거리감을 줄이고, AI를 활용한 방법론을 통해 전통적인 문학 연구와 디지털 인문학 간의 간극을 좁힐 수 있음을 보여준다. 특히 한국문학 연구에서 상대적으로 소외되어 온 '대중 수용사' 분야에 디지털 접근을 적용함으로써 새로운 연구 방법론의 가능성을 제시하며, 정성적 접근과 정량적 접근을 연결하는 시도로서 학문적 의의를 가진다.
주제어: 박경리, 신문 기사, 데이터 구축, 대중, 수용

Abstract

This study examines the reception of author Pak Kyongni and her novel *Land (Toji)* as a form of mass literary engagement and introduces a case of building a research-ready dataset from newspaper articles that significantly influenced this reception. The researcher collected and processed articles related to Pak Kyongni and Land, published across 24 newspapers from August 1955 to August 2024, by developing data collection scripts with the help of an AI chatbot.
This case demonstrates how humanities researchers can reduce the perceived distance from digital tools and narrow the gap between traditional literary studies and digital humanities through AI-assisted methodologies. In particular, by applying digital approaches to the relatively underexplored field of "mass audience reception history" within Korean literary studies, this research suggests the potential for new methodological developments. It also highlights the academic significance of

*단독저자: 연세대학교 글로벌창의융합대학 국어국문학과 통합과정, shincha23@gmail.com

connecting qualitative and quantitative methods in literary research.
Keywords: Pak Kyongni, newspaper articles, data construction, mass audience, reception

1. 들어가며

이 글은 '누구나'이면서 '누구도 아닌', 언제나 타자로서 어떤 책임도 없으면서 자기만의 고유성도 없는[1] '대중'이 '문학'을 어떻게 수용해왔는지, 그들에게 문학은 정말 앞길을 밝히는 길이었는지, 또 문학에게 대중은 과연 수동적인 수용자이기만 한 것이었는지 면밀하게 살펴보고자 하는 기획의 첫 단계로, 대중의 수용 환경을 분석하기 위한 데이터를 연구자가 직접 구축하는 과정을 다루고 있다. 연구 결과가 아니라 데이터 구축 과정부터 소개하는 이유는 이 글에서 소개하는 데이터 구축 과정이 질적 연구 방법론에 기반한 전통적인 국문학 연구 방법을 학습한 연구자가 인공지능(Artificial Intelligence. 이하, AI) 챗봇의 도움을 통해 독자적으로 연구데이터를 구축하는 과정이라는 점에서 전통적인 질적 연구 방법론에 익숙한 인문학 연구자들이 양적 연구 방법론을 접목한 디지털 인문학적 연구 방법론에 느끼는 거리감을 감소시키고, 한국문학 연구에서 소외된 분야인 '대중 수용' 연구 방법론 개발을 촉진하는 효과가 기대되기 때문이다.

대중 수용 연구 방법론을 위해 디지털 인문학적 연구 방법론이 도입되어야 하는 이유는 대중의 문학 수용 행위가 창작자의 창작 행위와 달리 생산물을 만들지 않는다는 점에서 쉽게 포착되지 않기 때문이다. 대중-문학의 관계는 특수한 개별 사례의 합이 아니라 전체적으로 파악되어야 할 집단적 구조[2]이기에 대중의 삶에 영향을 끼치는, 수용 환경을 구성하는 간접적인 것부터 감상 기록 같은 직접적인 것, 태도의 변화와 같은 암시적인 것까지를 복합적이고 종합적으로 살펴볼 필요가 있다. "특별한 것에서 일상적인 것으로, 특출한 사건들에서 사실들의 거대한 덩어리"[3]로 대중의 문학 수용 행위를 살펴본다면 대중과 문학의 관계가 새롭게 발견될 수 있을 것이다.

문학과 대중의 과거, 현재, 미래의 관계를 다각도로 살펴보고자 하는 연구 기획은 문단과 학계와 대중 모두에게 오랜 시간 사랑 받아온 박경리의 소설 『토지』[4]를 중심으로 한다. 원고지 3만 매가 넘는, 방대한 분량의 『토지』는 연재 시작 후 55년, 완간 30주년이 된 지금까지 수백 편의 연구로도 연구 주제가 닳을 기미가 보이지 않는다. 무엇보다 『토지』가 연재된 1969년은 박정희가 3선 개헌으로 장기독재를 내다보던 시점이며, 연재가 막을 내린 1994년은 김일성이 사망하여 남북관계가 급격하게 경색되기 시작하던 시점이었다. 격동의 대한민국 현대사 한가운데에서 조선 말에서 해방까지, 이전 세기의 격동을 다룬 『토지』를 읽은 독자는 누구였을까? 1974년의 영화, 1979~1980년, 1987~1989년, 2004~2005년. 세 차례 각색된 드라마, 음악극, 만화, 연극, 오랜 기간 다양하게 변용된 작품을 감상한 수용자들은 누구였는가? 그들의 수용 행위는 『토지』에게, 그들 개인에게, 대한민국 사회에게 무엇일까?

통속화되고 퇴폐적인 것으로 여겨지곤 하는 문화에 대한 대중의 소비는 지배적 생산 권력과 그들이 구축한 사회조직의 표면적 동질성에 일련의 틈새를 만드는, 능동적인 복수[5]일 수도, 직업생활과 경제활동이 더는 삶과 자아를 나아지게 만드는 것이 아니라 그저 유지하는 것만을 가능하게 하므로 폭식이나 '멍때리기', '소확행(나아가, 직장 생활에서 업무로 인한 자기만족을 추구할 수 없는 대중적 상태가 촉발하는 '소확횡'까지)'으로 대표될 수 있는, '삶의 마멸' 단계에서야 가능한 '측면적 행위 주체성'[6]일 수도 있다. 대중의 문화 소비 행위 가운데 '문학 수용'행위을 살펴 보는 것은 독서가 누구나 쉽게 입에 올리던 취미에서 '과시적 행위'가 되어버린 이때에, 대중의 능동적 주체성 탐구 행위를 적극적으로 살펴 보고, 그 행위의 확장 가능성을 탐구하고자 하는 것이다. '아무 생각 없이 영원히 보기'라는 수동적 수용 행위를 권장하는 미디어의 시대에, 문자라는 기호

를 매개로 한 '문학'은 수용자의 주체적인 상상을 필요로 하므로, '문학 수용' 행위는 '마멸'에 기반한 현대 사회에서 닳아 없어지지 않는, 대중이 삶의 재생산을 탐색할 수 있는 거의 유일한 주체적인 행위에 가깝기 때문이다.

이 글은 작가 박경리와 소설 『토지』의 수용 행위를 대중의 문학 읽기 행위로 보고 대중의 문학 수용 전/중에 큰 영향을 끼쳤을 신문 기사를 연구 가능한 데이터로 구축하는 과정을 소개하는 글이다. 앞서 말한 이 글의 기대 효과를 극대화하기 위해 '2장 디지털 인문학적 접근과 연구 설계'에서 인문학 연구자가 디지털 인문학 연구 방법론을 연구에 도입하기까지의 과정을 살펴 보고, '3장 데이터 구축 코드 개발 과정'을 통해 별다른 프로그래밍 지식 없이 AI 챗봇의 도움으로 필요한 코드를 생성하고 자료를 가공하는 과정을 알아본다. '4장 나가며'에서는 연구 과정의 의의와 추후 보완이 필요한 점을 살펴본다.

2. 디지털 인문학적 접근과 연구 설계

2.1 연구 대상 선정

2.1.1 『토지』의 대중 수용 관련 선행 연구

『토지』의 대중 수용과 관련된 선행 연구로는 연재 시작된 1969년부터 1999년까지를 연구 대상 시기로 삼아, 연구 대상 시기 기사가 디지털화되어 키워드 검색이 가능했던 『경향신문』, 『동아일보』, 『매일경제신문』(이하 『매일경제』), 『조선일보』, 『중앙일보』, 『한겨레』 6개 신문에서 '박경리', '소설 토지', '박경리 토지'로 검색된 기사 1,000여 건 가운데 『토지』와 직접적으로 관련된 기사를 통해 대중에게 『토지』가 어떻게 인식되었는지를 분석한 것[7]이 있다. 해당 연구는 기존 문학 연구에서 소외된 대중 수용과 관련된 연구라는 점, 주요한 연구 텍스트로 여겨지지 않았던 일간지 신문 기사 1,000여 건을 연구 대상으로 삼아 작가와 문학 작품 외에서 대중 수용에 끼친 영향력을 분석해보려 했다는 점, 연구자가 기사를 직접 읽어서 기사의 주제와 종류를 분류, 분석하였다는 점에서 의의를 지닌다. 다만, 해당 연구 방법론은 연구자가 직접 모든 연구 자료 전문을 읽고 분석해야 한다는 점에서 연구 확장에 물리적인 한계가 있다.

2.1.2 연구 대상 설정

본 연구자는 선행 연구의 성과를 바탕으로 하되, 객관적이고 종합적인 대중의 문학 수용 환경을 살펴보기 위해 연구 대상과 검색어를 확대하여 더 방대한 일간지 기사를 연구 데이터로 구축하고자 했다.

선행 연구의 분석 대상 텍스트는 "1969년~1999년 기사에 대한 키워드 검색이 가능한 『경향신문』, 『동아일보』, 『매일경제신문』, 『조선일보』, 『중앙일보』, 『한겨레』 6개"[8]이다. 본 연구는 이를 바탕으로 데이터 수집 대상 텍스트를 '연구 시작 당시(2024년 3월) 디지털화된 주요 일간지 신문 기사'로 확대하였다. '주요 일간지'는 대표성과 발행량을 고려하여 1) 10대 중앙 일간지 2) 9대 지역 일간지 3) 중앙 일간지에 버금가는 발행량을 보이는 전문지 및 기관지(이하, 전문·기

관지) 4종 4) 국가기간뉴스통신사 『연합뉴스』 1종, 총 24종으로 선정하였다.

데이터 수집 대상 시기는 선행 연구의 1969년 9월~1999년 12월 31일에서 1955년 8월~2024년 8월까지 69년으로 확대하였다. 작가 박경리의 위상이 대중의 『토지』 수용에 영향을 주었기 때문에, 기존 1969년 9월, 소설 『토지』 연재 시작 월에서 박경리 등단 월인 1955년 8월로 데이터 수집 대상 시기를 앞당겼다.

선행 연구는 <네이버 뉴스 라이브러리>와 『중앙일보』만을 분석 대상 일간지로 선정했었기 때문에 데이터 수집이 <네이버 뉴스 라이브러리>에 구축된 날짜까지만 가능했다. 본 연구는 구축 대상 데이터를 확대함으로써 <네이버 뉴스 라이브러리>에 한정된 연구 시기에서 자유로울 수 있었다. 대상 시기를 2024년 8월로 한정한 것은, 2024년 8월이 『토지』 완간 30주년이기 때문이다. 『토지』 완간 30주년을 기준으로 삼은 것은 『토지』의 대중 수용을 분석하기 위해서는 다른 보편적인 시계열보다 『토지』를 중심으로 한 시계열이 우선적으로 고려되어야 할 것으로 판단했기 때문이다. 완간 20주년인 2014년의 경우, 마로니에북스에서 2012년에 출간한 『토지』 전집이 대중 수용자에게 충분히 보급되었다고 보기 어렵기 때문에, 마로니에북스 판 『토지』 전집이 대중 수용자에게 충분히 보급될 시간을 확보하기 위해서 2024년 8월까지를 연구 대상으로 확대하였다. 확대된 연구 대상 시기는 추후 판본별 인식의 변화 등 보다 세부적인 수용 환경 분석도 가능하게 할 것이다.

작가 박경리는 1955년 등단하여 「불신시대」로 1957년 현대문학 신인문학상 수상, 1959년 『표류도』로 제2회 내성문학상 수상, 1964년 『시장과 전장』으로 제2회 여류문학상을 수상하며 등단 초기부터 10여 년 동안 문단에서 꾸준히 문학성을 인정받은 작가였다. 그뿐만 아니라 1960년 연재한 『성녀와 마녀』는 1969년에, 1962년에 출간한 『김약국의 딸들』은 다음 해인 1963년에 영화화되며 『토지』 연재가 시작되는 1969년에 이미 대중에게 잘 '읽히는' 인기 작가였다. 선행 연구에서는 소설 『토지』와 관계없이 '작가 박경리'만 언급된 기사는 분석 대상에 포함하지 않았음에도, 『토지』 연재 초기 대중의 작품 수용에 작가 박경리의 위상이 크게 영향을 끼쳤음을 확인[9]할 수 있었다. '작가 박경리'에 대한 기사를 최대한 많이 수집하기 위하여 '朴景利, 박경리, 박경이'를 검색어에 추가하였다.

선행 연구에서 제외되었던 변용 작품(영화, 드라마 등)이 역으로 『토지』 수용에 영향을 끼칠 수 있다. 이를 반영하기 위해 『토지』 변용 작품 관련 기사를 수집하기 위해 '작가 토지, 문학 토지, 예술 토지, 작품 토지, 드라마 토지, 영화 토지, 라디오 토지, 음악극 토지, 만화 토지('토지', '소설', '문학', '작품', '작가'의 경우 한자 '土地', '小說', '文學', '作品', '作家'도 검색어에 포함.)'를 검색어에 추가하였다.

확정된 연구 대상과 검색어를 정리하면 다음과 같다.

1. 연구 대상 : 2024년 3월까지 디지털화된 주요 일간지 신문 기사 가운데 서사 <토지>와 연관된 모든 기사

1) 10대 중앙 일간지 : 『경향신문』, 『국민일보』, 『동아일보』, 『문화일보』, 『서울신문』, 『세계일보』, 『조선일보』, 『중앙일보』, 『한겨레』, 『한국일보』

2) 9대 지역 일간지 : 『강원일보』, 『경남신문』, 『경인일보』, 『광주일보』, 『대전일보』, 『매일신문』, 『부산일보』, 『전북일보』, 『제주일보』

3) 중앙 일간지에 버금가는 발행량을 보이는 전문지 및 기관지 : 『농민신문』, 『매일경제』, 『한국경제』, 『스포츠조선』

4) 국가기간뉴스통신사 : 『연합뉴스』

2. 연구 대상 시기 : 1955년 8월~2024년 8월까지 69년간.

3. 검색어 : '박경리, 박경이, 작가 토지, 소설 토지, 문학 토지, 예술 토지, 작품 토지, 드라마 토지, 영화 토지, 라디오 토지, 음악극 토지, 만화 토지'('박경리', '토지', '소설', '문학', '작품', '작가'의 경우 한자 '朴景利', '土地', '小說', '文學', '作品', '作家'도 검색어에 포함.)

2.2 데이터 구축과 AI 챗봇 활용의 필요성

2.2.1 자체적인 데이터 구축의 필요성

필요한 일간지 기사 자료를 별도의 코딩 프로그램 활용 없이 수집하는 방법은 2025년 현재 크게 두 가지가 있다. 하나는 한국언론진흥재단에서 구축한 <빅카인즈>[10]를 활용하는 방법과 포털 뉴스의 오픈 API(application programming interface, 이하 API)[11]를 활용하는 방법이다.

1990년부터 2025년 현재까지 104개 매체의 약 1억여 건 뉴스콘텐츠 분석이 가능한 정형화된 데이터를 무료로 제공하는 <빅카인즈>는 별도의 데이터 분석 기술 학습이 없어도 방대한 양의 기사 데이터를 홈페이지 안에서 분석할 수 있다는 점에서 전통적인 질적 연구 방법론에 익숙한 인문학자가 이용하기에 유용한 서비스이다. 연구 대상 시기가 1955년 8월~2024년 8월까지 69년간인 본 연구자의 경우, <빅카인즈>로는 연구 대상 시기의 2/3에 해당하는 44년 간의 데이터를 구할 수 없다는 점에서 한계가 있다. 이 44년 간의 시기는 특히 박경리가 신인 작가에서 중견, 인기 작가로 입지가 변화하던 때와 『토지』 연재 기간 대부분을 포함한다는 점에서 <빅카인즈>에서 확보·분석 가능한 데이터는 『토지』를 중심으로 문학의 대중 수용사를 살펴보고자 하는 연구 기획에 적합하지 않다.

포털 뉴스의 오픈 API를 이용하는 방법은 대표적으로 네이버 검색 API[12]를 이용하는 것이다. 네이버가 제공하는 검색 API는 분야별로 검색 결과를 반환할 수 있기 때문에 연구자가 손쉽게 기사 데이터를 확보할 수 있다. 네이버 검색 API를 활용하는 방법의 가장 큰 문제는 현재 API에 검색을 진행할 기간이 매개 변수(파라미터)로 설정되어 있지 않다는 것이다. 네이버 검색 API는 한 번에 표시할 수 있는 검색 결과 개수의 최댓값을 100, 검색 시작 위치의 최댓값을 1000으로 제한하고 있는데, 이는 이론적으로 한 검색어의 검색 결과가 1,100건이 넘는다고 해도 검색어 당 결과를 정확도, 날짜순 내림차순으로 정렬하여 1,100건까지만 확보할 수 있다는 것을 의미한다.

'토지'는 일반명사로 많이 사용되기 때문에 1,100건 이내의 결과만으로는 유의미한 결과를 도출하기 어렵다. 추후 기간이 매개 변수로 추가된다고 하더라도, 현재 네이버 뉴스에서 수집할 수 있는 1990년~1999년의 기사는 『연합뉴스』뿐이며 그 외의 신문사는 2000년 이후의 기사만 수집할 수 있어 확보할 수 있는 자료의 시기 범주가 <빅카인즈>보다 적다. 이처럼 현재 구축된 일간지 데이터 분석 서비스나 포털 검색 API를 활용하는 것 모두 1990년 이후라는 최근 30여 년 여의 자료만을 연구 대상으로 삼을 수 있기 때문에, 1955년에 등단한 작가가 1969년부터 연재하기 시작한 소설의 수용사를 연구하기 위해서는 연구 대상 시기의 데이터를 자체적으로 수집하고, 최대한 기사 자료의 누락이 없게 하기 위해서 반복적인 중복 제거 작업 등 후작업을 거쳐 자체적인 데이터를 구축해야 했다.

2.2.2 AI 챗봇 활용의 필요성

인문학 연구 방법론에 익숙한 연구자는 '디지털 인문학 방법론을 사용한 분석'에 들어가기 전에 '컴퓨터가 읽을 수 있는 방식'으로 기존의 자료를 수집하고 정리해야 하며, 그를 위해 새로운 방법론의 기초부터 학습해야 한다. 문제는 방법론의 기초라고 할 수 있는 프로그래밍을 학습할 수 있는 교재나 강좌는 대체로 범용적인 목적에서 만들어졌기에 프로그래밍 기초를 학습한다고 하여도 특정 시기, 특정 분야의 특정 정보를 분석하고자 하는 연구 목적에 맞게 활용하기에는 한참 부족하다는 것이다. 이 단계에서 다른 연구자와 협업하거나 전문가에게 학습을 받아 연구 목적에 맞는 방법론을 학습·심화하는 방식으로 연구를 진행할 수도 있다.

단일한 집단이라고 할 수 없는 '대중'의 다양한 '수용' 과정을 긴 시기에 걸쳐 살펴 봄으로써 '문학'과 '읽기' 행위의 의의를 살펴보고자 하는 연구에서 '24종'이라는 다수의 일간지 기사를 살펴보는 것이 이미 분석할 수 있는 상태로 기사 데이터가 존재하는 특정 신문사 기사들만을 살펴보는 것보다 얼마나 의미가 있는지에 대해서 현재 연구된 바가 없다. 어떠한 결과도 예측할 수 없는 상황에서, 협업을 위해서라도 연구의 필요성을 확인하는 작업이 필요했기에 기초 학습 과정을 생략하고 당시(2024년 5월) 막 서비스되기 시작한 AI 챗봇인 'ChatGPT-4o'[13]를 활용하여 자료를 수집하였다.

6개월간 프로그래밍 기초 지식 없이 AI 챗봇을 활용하여 자료를 수집하며, 스스로 새로운 코드를 만들 수는 없어도 생성된 코드를 점검할 수 있는 정도까지는 코딩 과정에 익숙해질 수 있었다. 다만, AI 챗봇의 지원에도 연구자가 짧은 시간에 프로그래밍 전문가처럼 모든 과정을 능숙하게 다루기는 현실적으로 어려웠다. 이는 프로그래밍 학습 자체의 높은 진입 장벽을 보여주는 동시에, 그 간극을 메우는 데 AI 챗봇이 얼마나 긴요한지를 역설적으로 드러낸다. AI 챗봇의 적극적인 활용[14]은 본 연구자의 경우뿐만 아니라 인문학 연구자의 '디지털 인문학'에 대한 거리감을 좁히는 데에 도움이 될 것으로 보인다.

2.3 구축 대상 데이터와 방법

2.3.1 구축 대상 데이터

데이터베이스화되어 있는 24종의 신문사 기사 모두를 데이터로 구축하고자 먼저 <네이버 뉴스 라이브러리>[15], 각 신문사 홈페이지[16], <동아 디지털 아카이브>[17], <조선 뉴스 라이브러리 100>[18] 네이버 뉴스[19]에서 기사를 수집하였다.

『동아일보』와 『조선일보』의 1955년 8월~1999년 12월 기사는 <네이버 뉴스 라이브러리>와, 각 신문사가 구축한 기사 아카이브 홈페이지에서 중복적으로 수집되고, 『동아일보』의 경우 1996년, 『조선일보』의 경우 1993년부터의 기사는 신문사 홈페이지에서도 수집되었다. 『매일경제』의 1966년 10월~1999년 12월까지의 기사도 <네이버 뉴스 라이브러리>와 신문사 홈페이지에서 중복적으로 수집되었다. 박경리로 검색했을 때 <네이버 뉴스 라이브러리>에서는 『동아일보』 1959년 12월 26일 자 기사 "신문학 이래 수확의 해 <금년도작 <베스트·텐> 기타>"가 나오지만, <동아 디지털 아카이브>에서는 나오지 않는 등의 한 수집처에서만 기사를 수집할 경우 누락되는 기사가 발생하는 바, 최대한 누락을 줄이기 위해 중복되더라도 가능한 수집처에서 모두 수집한

후 중복된 기사를 제거하는 방식으로 데이터를 구축했다.

모든 기사는 각 신문별 수집 가능한 가장 오래된 기사에서부터 2024년 8월 31일 발행 기사까지만 수집하였다. 67,000건 이상의 기사가 2.1에서 확정된 검색어를 통해 수집되었고, 기사 본문에 박경리가 등장하지 않은 기사와 동명이인 등을 제외한 작가 박경리가 언급된 기사는 총 12,996건이다. 많은 검색어로 검색하고 많은 페이지를 순환하여 기사를 수집했기 때문에, 수집으로 인한 서버의 부하를 줄이기 위해 모든 작업은 신문사 당 하루 1시간 내외로만 진행했다. 기사 본문에 박경리가 등장하지 않은 기사는 실제 기사가 소설 『토지』나 변용 작품과 관련되지 않아도 '토지'라는 글자가 들어가 있어서 수집된 경우가 많아 현시점에서 확정할 수 없다.

작가 박경리가 언급된 신문별 데이터 수집 시작 시기와 최종 구축된 데이터의 기사 건수는 다음 표 1과 같다. 신문 분류 번호는 각각 1. 10대 중앙 일간지 2. 9대 지역 일간지 3. 전문·기관지 4. 국가기간뉴스통신사를 의미한다.

표 1 신문 목록과 기사 수집 가능 시기 및 박경리 언급 기사 건수

분류 번호	매체명	수집 시작 연월	기사 건수
1	『경향신문』	1955.08	896
1	『국민일보』	1990.01	435
1	『동아일보』	1955.08	1132
1	『문화일보』	1996.01	414
1	『서울신문』	1990.01	516
1	『세계일보』	1990.01	476
1	『조선일보』	1955.08	1109
1	『중앙일보』	1965.10	846
1	『한겨레』	1988.05	726
1	『한국일보』	1990.01	506
2	『강원일보』	1990.01	727
2	『경남신문』	1999.01	588
2	『경인일보』	1999.01	105
2	『광주일보』	2000.11	173
2	『대전일보』	2000.10	101
2	『매일신문』	2004.06	264
2	『부산일보』	1955.08	539
2	『전북일보』	2000.01	138
2	『제주일보』	2002.10	68
3	『농민신문』	1999.01	46
3	『매일경제』	1966.07	641

3	『스포츠조선』	2011.06	49
3	『한국경제』	1989.03	547
4	『연합뉴스』	1990.01	1954

2.3.2 구축 도구와 방법

데이터를 구축하는 기본 도구로 프로그래밍 언어 'Python'을 사용하였으며, 개발 환경으로 'Visual Studio Code'를 사용하였다. 활용한 AI 챗봇은 'ChatGPT-4o'이다. 구축하고자 한 기사 데이터의 정보는 '신문사, 기사 제목, 부제, 발행일자, 본문, 기사 주소(이하, URL)'이며, 데이터 구축 코드를 개발하는 과정은 크게 기사 수집 코드 개발과 전처리 과정으로 나뉜다. 기사 수집 코드를 개발하는 과정은 구체적인 웹페이지의 구성에 따라 변경되지만, 기본적인 과정은 다음과 같다.

① AI 챗봇에게 수집하고자 하는 URL과 수집 항목의 구조를 전달하고 기사 목록 수집하는 초기 코드 생성 요청

이때, URL 기준 중복 기사는 저장하지 않도록 요청하고, 여러 검색어를 순차적으로 입력할 것임도 미리 알려야 한다.

② ① 코드를 실행·수정하여 원하는 기사 목록 파일 생성

③ 개별 기사 페이지에서 수집해야 하는 정보가 포함된 태그와 광고와 기자 소개 등 수집에서 하는 제외해야 하는 정보가 포함된 태그를 확인

④ AI 챗봇에게 목록 파일에 포함된 URL 페이지에서 ③에서 확인한 태그 정보 중 필요한 정보만 수집하여 추가하는 초기 코드 생성 요청

⑤ ④ 코드를 코드를 실행·수정하여 원하는 기사 데이터 수집

3장에서는 웹사이트 구성에 따른 데이터 수집 코드 개발 과정과 전처리 코드 개발 과정을 살펴본다.

3. 데이터 구축 코드 개발 과정

3.1 일간지 기사 수집 코드

3.1.1 목록 수집: 주소형, 더보기형, 자바스크립트형

검색 결과 페이지를 변경하는 방식에 따라 각 신문사 웹사이트를 '주소형', '클릭형', '스크롤형' 3가지 유형으로 나누어 볼 수 있다. 신문사 웹사이트 변경 등으로 각 신문사에서 기사를 수집하는 방법은 언제든지 다른 방법으로 변경될 수 있다. 하지만 기사 목록 자체가 '사용자가 검색어를 일정 조건으로 검색하면, 그 결과를 조건에 맞추어 보여준다'는 과정의 결과물인 이상, 기사 목록에서 제공하는 정보와 검색 결과 페이지를 변경하는 방식은 크게 위의 세 가지 유형에 속할 수밖에 없다. 기사 목록을 수집할 때 먼저 원하는 웹사이트가 어떤 방식으로 목록을 제공하는지 확인

한 후, 그에 맞는 정보를 찾는 것이 중요하다. 아래에서 각각의 형태의 웹사이트에서 기사 목록을 저장하는 코드를 개발하는 과정과 주의할 점을 알아본다.

'주소형'은 검색어와 목록 페이지 수 등 검색 조건이 URL에 드러나 있어 주소에 페이지 숫자를 입력하여 기사 목록 페이지를 변경할 수 있는 유형이다. 가장 많은 신문사 웹사이트가 '주소형'을 택하고 있으며, URL에서 변수만 확인하면 되기 때문에 코드를 개발하기도 가장 쉬운 유형이다. 이 유형에 속하는 신문은 10대 중앙 일간지에서는 『경향신문』, 『동아일보』, 『문화일보』, 『서울신문』, 『조선일보』, 『중앙일보』, 『한겨레』, 『한국일보』, 9대 지역 일간지에서는 『강원일보』, 『경남신문』, 『경인일보』, 『광주일보』, 『대전일보』, 『부산일보』, 『전북일보』 『제주일보』, 전문·기관지 『농민신문』, 『스포츠조선』, 『한국경제』이며, <네이버 뉴스 라이브러리> 소재 신문들도 이 방식으로 목록을 수집할 수 있다.

AI 챗봇을 활용하여 '주소형' 웹사이트에서 목록을 수집하는 코드를 개발하는 과정은 다음과 같다.

① 검색어와 검색 날짜, 정렬 조건[20]를 설정해 검색한 다음, 해당 목록 페이지의 URL 확보

② 기사 목록 페이지 구조를 확인하여 기사 목록, 개별 기사 제목, 개별 기사 URL, 작성 날짜의 태그 확인

③ AI 챗봇에게 ① 주소와 검색어, 검색 날짜, 페이지 변경 방식과 ② 태그를 제공하고 기사 목록을 수집하는 기초 코드 생성 요청

④ ③ 코드를 실행·수정하여 원하는 기사 목록 파일 생성

기사 목록을 수집하는 코드를 생성하는 과정을 『경향신문』 홈페이지의 경우로 살펴본다.

① 『경향신문』 홈페이지(https://www.khan.co.kr/)에서 '박경리'를 검색한 후, 출력된 기사 목록 밑의 '전체 보기' 버튼을 누르면 검색 조건을 수정할 수 있다. 검색 설정의 기본 검색 범위가 '제목'으로 설정되어 있으므로, 범위를 '전체 범위'로 바꾸고 '선택 적용'을 하여 기사 목록 페이지의 URL을 확보한다. URL은 다음과 같다.

https://search.khan.co.kr/?q=%EB%B0%95%EA%B2%BD%EB%A6%AC&media=khan&page=1&section=0&term=5&startDate=2000-01-01&endDate=2024-08-31&sort=1

'주소형'의 기사 목록 페이지 URL에서 검색어, 검색 날짜, 최신순, 오래된순 등 정렬 조건은 대체로 검색어는 'query', 날짜는 'date', 정렬 조건은 'sort'로 표현되며 해당 단어 다음 = 이후 것이 연구자가 입력한 검색어나 날짜, 조건이다. 위 URL에서 검색어는 'query'의 약어인 'q'로 표현되어 있으며, 입력한 검색어인 '박경리'는 UTF-8 방식으로 인코딩되어 "%EB%B0%95%EA%B2%BD%EB%A6%AC"로 나타나 있다. 예시에서 오래된순으로 정렬된 조건은 한 번 설정하면 바꿀 일이 없기 때문에, 한 번 설정한 후 변수로 지정하지 않는다. 위 주소에서는 'q=', 'startDate=', 'endDate=' 다음을 변수로 설정하여 원하는 검색어, 검색 기간을 변경하여 기사 목록을 추출할 수 있다.

② 기사 목록 페이지 구조는 기사 목록 페이지에서 F12 키를 통해 확인할 수 있다. 위 URL의 페이지 구조 중 첫 번째 기사의 태그 내용은 다음 그림 1과 같다.

그림 1. '박경리'로 검색한 『경향신문』 홈페이지 기사 목록 1 페이지 1 번째 기사 HTML 구조 캡처

위 그림 1에서 확인할 수 있듯이, 기사 목록 페이지에서 기사들은 <section> 태그 내부의 <ul class="list"> 태그에 포함되어 있다. 각각의 기사는 <li> 태그에 담겨 있으며, 'ul > li > article'의 중간 구조를 거쳐 태그 내부의 <a> 태그에 제목과 URL이, <p class="etc last"> 태그에 작성 날짜가 담겨 있다. 그림 1의 기사는 따로 카테고리가 명시되어 있지 않지만, 카테고리가 지정된 기사들도 있으므로 <p class="category"> 태그로 기사의 카테고리도 추출할 수 있다.

③ AI 챗봇에게 기사 목록을 수집하는 기초 코드를 요청한다. 요청 프롬프트는 다음과 같다.

https://search.khan.co.kr/?q=%EB%B0%95%EA%B2%BD%EB%A6%AC&media=khan&page=1&section=0&term=5&startDate=2000-01-01&endDate=2024-08-31&sort=1

여기서 기사 목록을 엑셀로 저장하는 파이썬 코드를 짜줘. 셀레니움을 활용하고, 검색어, 검색 날짜는 변수로 지정해서 내가 나중에 변경할 수 있도록 해 줘.

저장할 내용은 제목, 개별 기사 URL, 기사 작성 날짜야.

위 그림 1에서 확인할 수 있듯이, 기사 목록 페이지에서 기사들은 <section> 태그 내부의 <ul class="list"> 태그에 포함되어 있어. 각각의 기사는'li > ul > li > article'의 중간 구조를 거쳐 태그 내부의 <a> 태그에 제목과 URL이, <p class="etc last"> 태그에 작성 날짜가 담겨 있어. 그림 1의 기사는 따로 카테고리가 명시되어 있지 않지만, 카테고리가 지정된 기사들도 있으므로 <p class="category"> 태그로 기사의 카테고리도 추출할 수 있어.

위의 프롬프트로 AI 챗봇이 제공한 기초 코드는 다음과 같다.

```
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.options import Options
import pandas as pd
import time
from urllib.parse import quote

# ✅ 설정: 검색어, 날짜, 최대 페이지 수
query = "박경리"
```

```
start_date = "2000-01-01"
end_date = "2024-08-31"
max_pages = 5  # 검색할 최대 페이지 수 (필요시 늘릴 수 있음)

# ✅ 셀레니움 옵션 설정
options = Options()
options.add_argument('--headless')  # 창 안 띄우고 실행
options.add_argument('--no-sandbox')
options.add_argument('--disable-dev-shm-usage')

# ✅ 크롬 드라이버 경로 (필요시 수정)
driver = webdriver.Chrome(service=Service(), options=options)

# ✅ 결과 저장용 리스트
results = []

# ✅ URL 생성
encoded_query = quote(query)
base_url = f"https://search.khan.co.kr/?q={encoded_query}&media=khan&section=0&term=5&startDate={start_date}&endDate={end_date}&sort=1"

# ✅ 페이지 순회
for page in range(1, max_pages + 1):
    url = f"{base_url}&page={page}"
    driver.get(url)
    time.sleep(2)

    articles = driver.find_elements(By.CSS_SELECTOR, "ul.list > li")
    if not articles:
        break

    for article in articles:
        try:
            a_tag = article.find_element(By.CSS_SELECTOR, "div > a")
            title = a_tag.text.strip()
            link = a_tag.get_attribute("href")

            date_tag = article.find_element(By.CSS_SELECTOR, "p.etc.last")
            date = date_tag.text.strip()

            # (선택) 카테고리 추출
            try:
                category_tag = article.find_element(By.CSS_SELECTOR, "p.category")
                category = category_tag.text.strip()
            except:
                category = ""

            results.append({
                "제목": title,
                "URL": link,
                "작성일": date,
                "카테고리": category
            })
        except:
```

```
                continue

    # ✅ 브라우저 종료
    driver.quit()

    # ✅ 엑셀로 저장
    df = pd.DataFrame(results)
    df.to_excel("khan_articles.xlsx", index=False)
    print(f"{len(df)}건 저장 완료.")
```

④ 원하는 데이터를 저장하기에 적합한지 기초 코드를 점검한다. 기초 코드는 검색어 1개, 기사 목록은 5페이지까지만 저장하게 되어 있으므로 목적에 맞게 코드 수정을 요청한다. 1차 수정 요청 프롬프트는 다음과 같다.

> 페이지는 더 이상 검색어의 기사가 없을 때까지나 새로운 기사가 하나도 없을 때까지 탐색하도록 하고,
> 검색어는 "박경리", "박경이", "작가 토지", "소설 토지", "문학 토지", "예술 토지", "작품 토지", "드라마 토지", "영화 토지", "라디오 토지", "음악극 토지", "만화 토지", "朴景利", "土地", "小說 土地", "文學 土地", "作品 土地", "作家"이고, 각각 돌아가면서 추출하도록 해 줘.
> url을 비교하여 중복된 기사는 한 번만 저장하게 해 줘.

AI 챗봇이 제공한 수정된 코드를 실행한 결과물에서 문제점을 확인한다. 해당 코드의 문제점은 한 페이지 당 기사를 1개만 추출하는 것, 어떤 기사들은 작성일 대신 작성자가 저장된다는 것이다. 이는 작성자가 입력된 기사의 경우, <p class="etc last"> 태그 안에 태그가 2개 존재하며, 첫 번째 태그에 작성자가 입력되었기 때문이다. 또한, 작성일이 연월일뿐만 아니라 작성 시간까지 추출되는 문제도 발견되었다. 이를 수정한 코드를 AI 챗봇에 요구한다. 2차 수정 요청 프롬프트는 다음과 같다.

> 현재 구조에서는 한 페이지 당 기사 1개만 저장하고 있어. 기사 전부 저장을 하게 바꾸고, 다른 기사를 보니까 etc 태그에서, span 태그가 2개 나오기도 해. 두 개일 경우 첫 번째 span은 작성자, 두 번째 span은 작성일이니까 그렇게 각각 추출하게 수정해 줘. span이 하나일 경우는 무조건 작성일이야. 작성일에서 시간 부분은 필요 없으니까, 띄어쓰기 앞 부분까지, 2024.08.25 15:52면 2024.08.25까지만 추출하게 해 줘.

2차 수정 코드를 실행한 결과, 작성자와 작성일은 제대로 추출되었으나, 여전히 페이지당 기사 1건만 저장하는 문제가 발생하였다. 이는 실제 기사 리스트 구조가 'ul.list > li > ul > li'임에도, 처음 프롬프트에서 제대로 구조를 설명하지 못하여 추출 대상 셀렉터가'ul.list > li'로 지정되어 발생한 문제이다. 구조를 연구자가 문장으로 설명하기보다 구조를 직접 캡처한 화면을 AI 챗봇에게 제공하는 등의 방법을 사용하면 위와 같은 문제 발생을 줄일 수 있다. 다만, HTML 구조를 직접 캡처하여 제공하여도 AI 챗봇의 문제로 셀렉터가 잘못 설정되는 문제가 적지 않게 발생하므로, 원하는 정보가 제대로 추출되지 않을 때에는 셀렉터가 제대로 지정되었는지 우선적으로 살펴볼 필요가 있다.

셀렉터를 수정한 3차 수정 코드로 원하는 기사 목록을 추출했다면, 저장하는 파일 제목의 형식, 수집한 항목명을 지정하여 최종적으로 코드를 정비한다. 다른 홈페이지에서 같은 방식으로 기사 목록을 추출하는 코드를 생성할 때, 정비된 코드를 프롬프트와 함께 AI 챗봇에 제공하면 AI 챗봇이 HTML 구조를 제외한 다른 항목을 통일하여 코드를 생성해주어 코드 생성 과정을 단축할 수 있다.

‘주소형’ 웹사이트에서 기사 목록을 수집하면서 몇 가지 주의해야 하는 부분은 다음과 같다. 1페이지 URL에서 검색 날짜나 정렬 조건을 지정하는 변수가 노출되지 않을 수도 있다. 1페이지에는 드러나지 않더라도 직접 목록 번호를 클릭해 다른 페이지로 넘어가면 드러나기도 하기 때문에, ①의 목록 URL을 확보하는 과정에서 모든 조건이 URL에 드러나는지 확인할 필요가 있다.

페이지 루프를 종료하는 방법은 웹사이트 구조에 따라 기사 리스트에 기사가 없을 경우 종료하거나 새로운 기사가 없을 때 종료하는 두 가지 방법을 적용할 수 있다. 어떤 방법을 사용할 것인지는 검색어의 마지막 페이지+1페이지를 입력해보고, 기사가 아예 없는 것으로 표시되는지, 마지막 페이지가 반복해서 표시되는지 확인할 필요가 있다. 마지막 페이지가 반복 표시되는 웹사이트에서는 기사가 없을 경우 종료하는 방법을 적용할 경우 영원히 페이지 숫자를 늘려 수집하기 때문에, 수집한 기사의 개별 URL을 기준으로 새로 추가된 기사의 숫자를 확인해서 새로운 기사가 없으면 루프를 종료하는 방법을 적용해야 한다.

‘클릭형’은 일정한 숫자의 기사가 페이지별로 제공되는 것이 아니라 페이지 번호나 ‘더보기’ 버튼을 직접 클릭하여 기사를 추가로 로드하는 유형이다. 페이지 번호를 클릭하는 유형은 『국민일보』, 『세계일보』, 『매일신문』이고, ‘더보기’ 버튼을 클릭하는 유형은 『매일경제』이다.

AI 챗봇을 활용하여 ‘클릭형’ 웹사이트에서 목록을 수집하는 코드를 개발하는 과정은 다음과 같다.

① 검색어와 검색 날짜, 정렬 조건을 설정해 검색한 다음, 해당 목록 페이지의 URL 확보

② 기사 목록 페이지 구조를 확인하여 기사 목록, 개별 기사 제목, 작성 날짜, 클릭해야 하는 버튼의 태그 확인

③ AI 챗봇에게 ① 주소와 검색어, 검색 날짜, 페이지 변경 방식과 ② 태그를 제공하고 기사 목록을 수집하는 기초 코드 생성 요청

④ ③ 코드를 실행·수정하여 원하는 기사 목록 파일 생성

‘클릭형’에서 주의할 점은 사용자가 직접 클릭하는 것처럼 작업을 진행해야 하기 때문에, 원하는 정보가 충분히 로드될 수 있도록 충분한 대기 시간을 주어야 한다는 것이다. 정보가 로드되는 시간은 유동적이므로, time.sleep()보다는 selenium.webdriver의 WebDriverWait를 추천한다.

직접 페이지 번호를 클릭하여 페이지를 루프하는 ‘클릭형’은 하나의 태그가 반복적으로 사용되는 ‘더보기’형보다 많은 정보를 확인하고 페이지 루프가 잘 진행되는지 주의 깊게 확인해 보아야 한다. 보통 페이지 번호와 태그 번호가 일치하지만, 시각적으로 표시되는 정도까지만 그렇게 진행되고, 시각적으로 표시되는 정보 이상의 값을 넣어도 해당 페이지로 이동하지 않는다. 예를 들어, 페이지 번호가 1, 2, … 9, 10까지 표시되고 11부터는 ▶ 버튼을 눌러야 표시가 되는 구조라면, 1에서 10까지는 태그에 번호를 1씩 더해서 클릭하고, 10n+1번째에는 ▶를 누를 수 있도록 해주어야 한다는 것이다. 페이지 루프를 종료하는 방법은 더는 다음 페이지가 없거나, 현재 페이지가 끝 페이지(보통 버튼으로 마지막 페이지에 갈 수 있게 설정되어 있다.)와 같을 때 종료하는 조건을 설정하면 된다. ‘주소형’ 마찬가지로 적용하는 태그에 번호 1을 더했을 때 새 기사가 없거나, 기존 기사와 동일한 기사 목록이 뜨는지에 따라 맞는 조건을 설정하면 된다.

‘스크롤형’은 스크롤을 내리면 새로운 기사가 로드되는 유형이다. 네이버 뉴스 탭이 ‘스크롤형’으로 구성되어 있다. 네이버 뉴스를 통해서만 10년 이상의 기사를 수집할 수 있는 『연합뉴스』가 이 유형에 속한다.

AI 챗봇을 활용하여 ‘스크롤형’ 웹사이트에서 목록을 수집하는 코드를 개발하는 과정은 다음과 같다.

① 검색어와 검색 날짜, 정렬 조건을 설정해 검색한 다음, 해당 목록 페이지의 URL 확보
② 기사 목록 페이지 구조를 확인하여 기사 목록, 개별 기사 제목, 작성 날짜 태그 확인
③ AI 챗봇에게 ① 주소와 검색어, 검색 날짜와 ② 태그를 제공하고 스크롤을 끝까지 내려서 모든 기사를 로드한 다음에 기사 목록을 수집하는 기초 코드 생성 요청
④ ③ 코드를 실행·수정하여 원하는 기사 목록 파일 생성

'스크롤형'에 해당하는 네이버 뉴스 탭에서 기사를 수집할 때 주의점은 목록은 네이버 뉴 스에서 수집할 수 있지만 대다수의 개별 기사가 네이버 뉴스 내부가 아니라 개별 신문사로 연결된다는 점이다. 목록을 먼저 만들고 개별 기사 본문을 수집하는 이유는 코드의 실행과 수정 과정을 최소화하기 위함인데, 개별 기사 URL이 개별 신문사로 연결된다면 각 신문사에 구조에 맞는 본문 수집 코드를 새로 개발해야 한다는 점에서 번거롭다. 네이버 뉴스 탭에서 수집할 수 있는 대다수의 기사를 각 신문사 웹사이트, <빅카인즈> 등에서도 수집할 수 있으므로, 수집 날짜로부터 10년 이전의 『연합뉴스』 기사처럼 네이버 뉴스에서만 수집할 수 있는 기사만 네이버 뉴스 탭을 이용하는 것이 좋다.

기사 목록만 필요하다면 '스크롤형'의 코드를 활용하는 것이 다른 방법보다 간단하다. 검색 URL 'news_office_checked=1001'에서 '1001'이 『연합뉴스』의 코드를 의미하므로, 원하는 신문사의 코드만 확인하여 교체하면 하나의 코드로 기사 목록을 손쉽게 수집할 수 있다.

'주소형', '클릭형', '스크롤형'으로 기사 목록을 수집하는 코드를 개발하는 과정을 살펴보았다. 이는 기사 목록을 탐색하는 방식만을 기준으로 나눈 것으로 웹사이트별로 로그인이 필요하거나, 검색어를 직접 검색창에 입력부터 해야 하는 경우 등이 존재한다. 이때는 앞선 3가지 유형의 코드 생성을 기본으로 하고, 각 웹사이트 구조와 조건에 맞게 해당 방법을 추가하는 코드를 AI 챗봇에 요청하면 된다.

다음은 수집된 기사 목록을 바탕으로 기사 본문을 수집하는 코드를 개발하는 과정을 살펴본다.

3.1.2 본문

<네이버 뉴스 라이브러리>를 제외한 신문사 기사 본문 수집 코드 개발은 유형을 나누었던 기사 목록 코드와 달리 유형을 나눌 필요가 없다. 코드 개발 과정은 간단하지만, 필요한 정보만 수집하기 위해 코드를 실행·확인·수정을 반복적으로 수행해야 한다는 점에서 사실상 가장 많은 시간이 걸리는 작업이다.

AI 챗봇을 활용하여 기사 본문을 수집하는 코드를 개발하는 과정은 다음과 같다.

① 개별 기사 페이지에서 기사 부제와 본문, 삭제해야 할 내용(사진 캡션 등)의 태그를 확인
② AI 챗봇에게 ①의 태그를 제공하고 기사 부제와 본문을 나누어서 수집하는 기초 코드 생성 요청
③ ② 코드를 실행·수정하여 원하는 기사 파일 생성

주의할 점은 같은 신문사라 하여도 웹페이지 구조가 변경된 경우 본문 구조가 다를 수 있다는 점이다. 본문 태그 및 삭제해야 할 내용을 담은 태그는 최소 10개 이상의 서로 다른 시기 기사를 통해 확인한 후 모두 반영할 수 있도록 코드를 개발해야 한다. 『조선일보』의 경우처럼 본문이 담긴 태그가 시기나 기사의 종류에 따라 굉장히 다양한 경우나 『중앙일보』처럼 사진 설명, 광고 태그가 다양한 경우에는 태그들을 리스트로 만드는 것이 추후 코드를 수정하기 용이하다. 삭제해야 할 태그는 모든 것을 전부 삭제하는 코드를 사용하면 되지만, 본문 태그가 다양할 경우,

아래 인용과 같이 본문을 추출할 때까지 본문 태그들을 순회하도록 코드를 구성해야 한다.

```
# 여러 가지 클래스명 중 첫 번째로 찾은 본문 추출
article_classes = [
    '_article_content', 'newsct_article', 'article-txt',
    'story-news', 'content01', 'viewContentWrap',
    'atic_txt1', 'articleDiv', 'articleBody'
]
# 각 클래스가 로드될 때까지 시도 (첫 번째로 로드되는 클래스 기다림)
for class_name in article_classes:
    try:
        WebDriverWait(driver, 3).until(
            EC.presence_of_element_located((By.CLASS_NAME, class_name))
        )
        break  # 첫 번째로 로드된 클래스가 있으면 루프 종료
    except TimeoutException:
        continue  # 해당 클래스가 로드되지 않았을 경우, 다음 클래스를 시도
soup = BeautifulSoup(driver.page_source, 'html.parser')
article_body = None
for class_name in article_classes:
    article_body = soup.find('div', class_=class_name)
    if article_body:
        break  # 본문을 찾으면 루프 종료
```

따로 태그로 구성되지 않은 기사 작성 기자의 정보, 저작권 표시 등 분석하고자 하는 주제와 무관한 내용은 수집되지 않도록 처리하는 것도 중요하다. 특정 정보가 기사의 특정 부분에 반복적으로 노출이 되는 경우는 아래와 같은 방식으로 해당 내용을 수집하지 않을 수 있다.

```
 # 본문에서 "GoodNews paper ⓒ" 이후 내용을 제거합니다.
content = article_body.get_text(separator='\n', strip=True)
cutoff_phrase = "GoodNews paper"
```

부제는 신문사마다 처리 방법이 다르고, 과거 기사에서는 태그로 구분이 되어 있지 않은 경우가 훨씬 많다. 최신 기사를 기준으로 부제를 분리할 수 있는 경우는 본문과 분리하여 따로 수집하도록 개발했으나, 현재 코드상으로 따로 분석을 할 정도로 일관성이 있거나 양적으로 충분하게 수집되지 않았다. 태그로 분리되지 않은 부제들을 추출할 수 있는 방법이 추후 개발되어야 한다.

3.1.3. <네이버 뉴스 라이브러리> 본문

<네이버 뉴스 라이브러리>의 경우, 목록에서 기사를 누르면 다음 그림 1처럼 '텍스트 보기' 팝업이 뜬다.

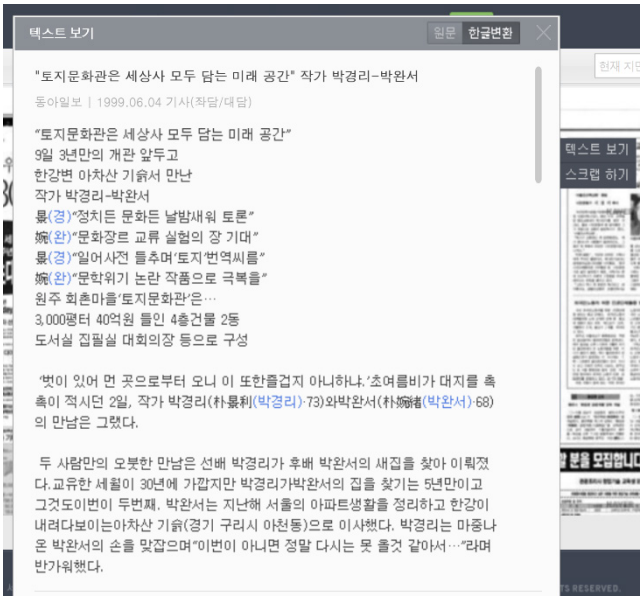

그림 2. <네이버 뉴스 라이브러리> 1999 년 6 월 4 일 13 면 기사 회면

제목 행과 신문사, 발행일, 기사 종류가 담긴 정보 행은 복사할 수 있는 텍스트 형태로 제공되지만, 아래 본문은 복사할 수 없는 그림의 형태로 제공되기 때문에 기사 페이지에서 본문을 스크랩할 수 없기 때문에, 다른 기사들과 달리 기사 검색에 사용되는 데이터에서 기사 본문을 추출해야 한다. 그림 2처럼 검색어가 기사 본문 어디에, 어떤 방식으로 표시되는지 보여주는 div.summary 태그의 내용은 API에 POST 방식으로 요청하고, JSON 형식으로 반환된다는 것은 네트워크 분석을 통해 알아낼 수 있다.

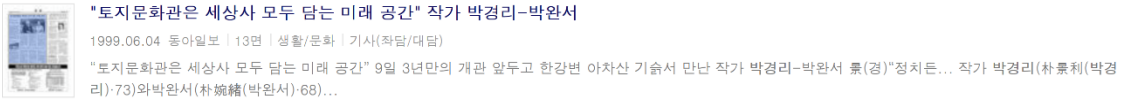

그림 3. <네이버 뉴스 라이브러리> '키워드 검색'을 사용하여 나온 기사 예시

AI 챗봇을 활용하여 <네이버 뉴스 라이브러리> 기사 본문을 수집하는 코드를 개발하는 과정은 다음과 같다.

① 기사 목록 파일의 각 기사 URL에서 개별 기사 ID 추출하고 저장하는 코드 생성 요청 후 실행

② AI 챗봇에게 각 기사에 해당하는 ID와 URL 정보를 기반으로 API에 POST 방식으로 요청을 수행하고, JSON 형식으로 반환된 응답으로부터 본문 데이터를 추출하여 저장하는 코드 요청

③ ② 코드를 실행·수정하여 원하는 기사 파일 생성

3.2 수집 데이터 전처리 코드

3.2를 통해 수집된 기사들은 다양한 수집처와 시기에 걸쳐 작성된 자료들이기 때문에 분석에 앞서 중복 기사 정리 및 사용된 특수문자와 주요 용어들을 통일하는 전처리 작업이 필요하다. 중복 기사 정리 작업은 최종적으로 연구자가 전수 검사를 통해 가려야 했으므로, 따로 코드를 생성·

실행하지 않고, ① AI 챗봇에게 신문사별 기사 파일을 전송 ② '날짜가 같은 기사 중 제목이나 내용이 50% 이상 비슷한 기사들만 따로 저장'을 요청 ③ ②의 파일들의 기사를 직접 보고 정리하는 과정으로 진행했다. 중복된 기사는 신문사 홈페이지→아카이브 홈페이지→<네이버 뉴스 라이브러리> 순서로 남기고 정리했다. 기사는 중복 기사를 정리하는 과정을 통해 아카이브 홈페이지에 있는 기사가 신문사 홈페이지에 전부 수록되어 있는 것을 확인하여, 공개하는 코드에는 아카이브 홈페이지에 없는 시기부터 사용하도록 주석을 추가했다.

기사에 사용된 특수문자와 주요 용어들을 통일하는 전처리 작업은 코드 생성 자체는 간단하지만, 추후 진행할 디지털 인문학적 분석에서 더 유의미하고 정확한 분석 결과를 만들기 위해서는 다양하게 사용된 특수문자/용어를 최대한 전부 확인하여 예외를 만들지 않는 것이 중요하다. 특수문자의 경우 작품명이 대체로 특수문자로 쌓여 있다는 점에서, 짝을 이루는 특수문자를 중심으로 통일했다. 용어는 현재 박경리의 작품명과 동시대 작품명, 주요 매체명, 박경리 관련 매체명 등으로 분류를 나누어서 통일하고자 하였다. 다만, 용어를 통일하는 작업은 기사들에서 사용된 용어를 확인하는 작업의 양이 방대하여 아직 완결되지 못했다.

용어를 통일하는 작업에서 가장 문제가 되는 것은 '한자'이다. 1955년부터 2024년까지의 긴 시간을 다루는 이 연구에서 신문사에 따라 편차가 있지만 대략 1955년~1980년대 중반까지 한자가 기사에 혼용되고, 그 이후에도 사람의 성씨나 사람의 이름은 한자로 표기되는 일이 많다. 사람의 이름 같은 경우 띄어쓰기 없이 붙은 상태가 많아 한글로 변경하면서 인물 이름별로 구분할 수 있는 방식으로 변경이 필요하다. 공개된 한자-한글 변환 라이브러리를 활용하여 변경하되, 박경리와 관련된 인물이나 단체명 등은 네트워크 분석 등의 활용을 위해 따로 라이브러리로 구축하여 깃허브(GitHub)에 공개할 예정이다.

AI 챗봇을 활용하여 특수문자/용어를 통일하는 코드를 생성하는 과정은 ① 기사 파일을 로드해서 A의 내용을 B로 전부 바꿔주는 코드를 만들어달라고 요청 ② 각 분류에 맞춰 코드를 변경하는 것으로 이루어졌다. 하지만 변경 전 다양한 사례의 양이 더 많아질 것으로 예상되므로, 코드 내부에 변경 전과 후 사례를 입력하는 방식이 아니라, 다양한 사례와 통일할 용어를 파일로 정리한 후 파일의 내용을 적용하도록 코드를 수정할 예정이다.

4. 나가며 – 수집과 구축의 성과, 그리고 과제

현재 인문학 연구자가 직접 자료를 수집하고 분석하는 데에 코드를 생성하지 않고 활용할 수 있는 <빅카인즈>, <네이버 뉴스 라이브러리>는 과거와 현재의 자료를 장기간 통시적으로 연구하기에는 어려움이 있다. 1990년대 이전을 연구 대상으로 하거나, 1990년대 이전부터 현재까지를 통시적으로 연구하고자 하는 연구자는 2025년 현재 기준으로 자신에게 필요한 데이터를 직접 수집하기 위해 코드를 개발할 필요가 있음이 이 연구를 통해 확인되었다. 코드의 개발은 프로그래밍 지식이 없는 인문학 연구자에게 단기간에 학습하고 실행할 수 있는 작업이 아닌 바, 이 과정에서 AI 챗봇을 활용할 필요가 있다.

AI 챗봇을 활용하여 코드를 개발하고 실행하기 위해서는 수집하고자 하는 대상의 사이트 구조, 스크래핑 과정에 대한 최소한의 이해는 필요하다. 자료를 수집하고 정리하는 것은 AI 챗봇의 도움으로 연구자 개인이 혼자 진행할 수 있지만, 자료를 디지털 인문학적 분석이 가능한 상태로 가공하는 전처리 작업에는 생각보다 물리적인 시간이 많이 필요하다. 하지만 전처리 작업의 규칙

등은 동시대의 텍스트에 동일하게 적용할 수 있으므로, 같은 시기나 매체를 연구하는 연구자들의 협업으로 물리적인 시간을 단축할 수 있을 것으로 예상된다. 연구자가 개발한 코드는 현재 https://github.com/ShinJeongEun/PakKyongni_NAD에서 확인할 수 있으며, 지속적으로 코드와 데이터를 업데이트할 예정이다.

AI 챗봇의 개발과 활용은 기존의 연구 대상들을 새롭게 바라보는 시각을 제공하는 것에 더해서 직접 자료를 찾기 어렵고 너무나 방대한 공간에 흩어져 있는 자료들을 모아서 정리해야 하는 관계로 문학 연구에서 불모지로 여겨진 독자 연구, 수용자 연구를 실현할 수 있다는 점에서 의의가 있다. 다만, 앞서 말한 것처럼 본격적인 분석 작업에 들어가기 전의 전처리 작업에 시간이 많이 소요되어 연구자 개인이 진행하기에는 지난한 작업이 예상되므로 연구자 간의 협업을 통해 전처리 과정의 시간을 단축하고 적합한 분석 방법을 개발·공유하는 것이 필요하다.

기사를 수집하는 과정에서 토지 관련 기사도 수집했으나, 일반명사 '토지'와 박경리의 소설 『토지』와 그 변용 작품들을 일컫는 고유명사 <토지>를 구분하는 방법에 대한 탐색이 필요하여 이 글에서는 <토지> 관련 기사를 선별·정리하는 과정을 언급하지 않았다. 관행적으로 작품명은 특수문자로 감싸서 표기하므로, '토지'라는 글자가 특수문자로 싸여 있다면 고유명사 <토지>와 관련된 기사일 확률이 높다. 특수문자로 싸이지 않은 '토지'라는 글자가 고유명사 <토지>일 수도 있으므로, 이를 위해 표기 이외의 추가적인 기사 선별 방법의 개발이 필요하다. 추후'박경리'가 포함된 기사들도 포함하여'토지'라는 단어와 '작품, 소설, 영화, 드라마' 등의 단어들의 거리를 분석하여 고유명사 <토지> 관련 기사를 선별하는 방법을 탐색할 예정이다.

일간지 기사 수집 코드 개발과 활용에서 보완해야 하는 사항을 살펴보는 것으로 이 글을 마무리하고자 한다. 여러 보완점이 있지만, 분석이 필요하나 웹사이트 구조의 차이로 인해 분석할 만큼 충분한 자료를 확보하지 못한 항목인 지면과 부제 문제를 대표적으로 살펴본다.

박경리가 언급된 기사의 수록 지면을 살펴보는 것은 박경리의 대중적인 인지도와 영향력의 파급력을 간접적으로 확인할 수 있는 중요한 작업이다. 작가 개인을 넘어서 문학 작품이 언급된 지면까지 나아갈 때, 문학 작품이 어떤 방식으로 어떤 지면에 언급되는가는 보다 직접적으로 문학의 대중 수용 환경을 살펴볼 수 있다는 점에서 반드시 필요한 작업이다. 최신 뉴스는 종이신문보다 건건이 인터넷 게시물로 읽히기 때문에 따로 종이신문의 지면을 제공하지 않거나, 섹션을 제공해도 같은 기사가 여러 섹션에 중복 게재되기에 지면이나 섹션에 따른 분류의 중요도가 낮을지 몰라도, 적어도 대중이 인터넷 신문보다 종이신문을 주로 접하던 2000년대 이전에는 지면에 대한 분석이 필수적이다. 하지만 <네이버 뉴스 라이브러리>와 몇몇 신문사를 제외한 대다수의 신문들이 수록 지면을 따로 제공하고 있지 않다.

제목과 부제는 지면 안에서 본문보다 크고 가시성이 높게 표현되어 있으므로 '훑어보는' 신문 읽기의 특성상 수용자의 인식에 미치는 영향이 크다. 제목과 함께 부제를 수집하고 분석하는 일은 대중의 문학 수용에서 중요한 작업이다. 그러나 실제 수집 과정에서 신문사별로 부제를 표시한 경우가 적었으며, 특히 과거 기사는 부제와 본문이 명확하게 구분되어 있지 않았다. 본문 도입부의 마침표 없는 문장 등을 부제로 분류하기 등 몇 가지 방법을 도입해 보았으나 실제 부제가 아닌 것들이 수집되는 일이 빈번하게 발생하여 현재의 기술적 조건에서 일관된 규칙에 따라 부제를 자동 추출할 수는 없다는 한계를 확인했다. 추후 부제 추출을 위한 규칙을 개발할 필요가 있다.

[1] Michel de Certeau. 신지은 역(2023). <일상의 발명: 실행의 기예>. 문학동네. 65. https://lod.nl.go.kr/page/KMO202347490

[2] Franco Moretti. 이재연 역(2020). <그래프, 지도, 나무: 문학사를 위한 추상적 모델>. 문학동네. 12.
[3] Franco Moretti. 이재연 역(2020). <그래프, 지도, 나무: 문학사를 위한 추상적 모델>. 문학동네. 11.
[4] 이 글에서 박경리의 소설 『토지』는 '『토지』'로, 소설 『토지』를 바탕으로 한 콘텐츠는 '<토지>'로 표기한다.
[5] Michel de Certeau. 신지은 역(2023). <일상의 발명: 실행의 기예>. 문학동네. 107. https://lod.nl.go.kr/page/KMO202347490
[6] Lauren Berlant. 박미선·윤조원 역(2024). <잔인한 낙관>. 후마니타스. 189-220. https://lod.nl.go.kr/page/KMO202443730
[7] 신정은(2023). "토지의 대중 인식 변화 연구: 1969 년~1999 년 일간지 기사를 중심으로". <현대문학의 연구> 79. 7-37. https://doi.org/10.35419/kmlit.2023..79.001
[8] 신정은(2023). "토지의 대중 인식 변화 연구: 1969 년~1999 년 일간지 기사를 중심으로". <현대문학의 연구> 79. 10. https://doi.org/10.35419/kmlit.2023..79.001
[9] 신정은(2023). "토지의 대중 인식 변화 연구: 1969 년~1999 년 일간지 기사를 중심으로". <현대문학의 연구> 79. 12-20. https://doi.org/10.35419/kmlit.2023..79.001
[10] <빅카인즈>, https://www.bigkinds.or.kr/ (접속일: 2025.03.11.)
[11] API 는 운영체제나 시스템, 애플리케이션, 라이브러리 등을 활용해 응용 프로그램을 작성할 수 있게 하는 다양한 인터페이스를 의미한다. 오픈 API 는 API 중 플랫폼의 기능 또는 콘텐츠를 외부에서 호출해 사용할 수 있게 개방한 API 를 말한다. API 정의는 "용어 정리", <네이버 Developers>, https://developers.naver.com/docs/common/openapiguide/apiterms.md#api%EC%9D%98-%EA%B8%B0%EB%B3%B8 (접속일: 2025.03.14.).
[12] "API 소개", <네이버 Developers>, https://developers.naver.com/products/intro/plan/plan.md#%EB%84%A4%EC%9D%B4%EB%B2%84-%EC%98%A4%ED%94%88-api-%EB%AA%A9%EB%A1%9D (접속일: 2025.03.14.).
[13] 이 글에서 'AI 챗봇 활용'에 대한 언급은 모두 'ChatGPT-4o'를 사용한 것이다.
[14] 현재 AI 챗봇을 활용하는 방법은 그 챗봇의 종류와 무관하게 입력한 정보를 AI 에게 '학습'시킬 수 있다는 점에서 데이터의 보안 등의 문제가 있어 사용에 주의가 필요하다.
[15] <네이버 뉴스 라이브러리>, https://newslibrary.naver.com
[16] 『경향신문』, https://www.khan.co.kr; 『국민일보』, https://www.kmib.co.kr; 『동아일보』, https://www.donga.com; 『문화일보』, https://www.munhwa.com; 『서울신문』, https://www.seoul.co.kr; 『세계일보』, https://www.segye.com; 『조선일보』, https://www.chosun.com; 『중앙일보』, https://www.joongang.co.kr; 『한겨레』, https://www.hani.co.kr; 『한국일보』, https://www.hankookilbo.com; 『강원일보』, https://www.kwnews.co.kr; 『경남신문』, https://www.knnews.co.kr; 『경인일보』, http://www.kyeongin.com; 『광주일보』, http://www.kwangju.co.kr; 『대전일보』, https://www.daejonilbo.com; 『매일신문』, https://www.imaeil.com; 『부산일보』, https://www.busan.com; 『전북일보』, https://www.jjan.kr; 『제주일보』, http://www.jejunews.com; 『농민신문』, https://www.nongmin.com; 『매일경제』, https://www.mk.co.kr; 『스포츠조선』, https://sports.chosun.com; 『한국경제』, , https://www.hankyung.com
[17] <동아 디지털 아카이브>, https://www.donga.com/archive/newslibrary
[18] <조선 뉴스 라이브러리 100>, https://newslibrary.chosun.com
[19] 네이버 뉴스에서 검색할 경우, 포털 뉴스 검색 결과로 연결되고, 기사를 클릭하면 해당 신문사 홈페이지에 탑재된 기사 페이지로 연결되기 때문에, 다른 23 개의 신문은 신문사의 홈페이지에서 기사를 수집하는 것으로 대신하였지만, 『연합뉴스』의 경우 홈페이지에서는 10 년 간의 기사만 확인할 수 있기에, 1990 년 기사부터 확인할 수 있는 네이버 뉴스 탭에서 수집하였다.
[20] 정렬 조건에 '정확도순'이 기본으로 설정된 경우, 검색을 시행할 때마다 1 페이지에 노출되는 기사가 달라질 수 있다는 점에서, 시차를 두고 기사를 수집할 때 중복 수집이나 누락이 발생할 수 있다. 기사 노출 순서가 변경될 수 없는 과거순이나 최신순으로 기사 정렬 조건을 선택한 후 수집해야 한다

참고문헌

Berlant, Lauren. 박미선·윤조원 역(2024). <잔인한 낙관>. 후마니타스. https://lod.nl.go.kr/page/KMO202443730

de Certeau, Michel. 신지은 역(2023). <일상의 발명: 실행의 기예>. 문학동네. https://lod.nl.go.kr/page/KMO202347490

Moretti, Franco. 이재연 역(2020). <그래프, 지도, 나무: 문학사를 위한 추상적 모델>. 문학동네. http://lod.nl.go.kr/resource/KMO202013161

신정은 (2023). “토지의 대중 인식 변화 연구: 1969 년~1999 년 일간지 기사를 중심으로”. <현대문학의 연구>. 79. https://doi.org/10.35419/kmlit.2023..79.001