A Python Library for Measuring the Passage of time in Fiction

이시헌[1], 황태검[1], ???[1*]
[1] *협성대학교 소프트웨어공학과, 화성, 대한민국*
[*]Corresponding Author: ???

1. 서론

1.1. 연구 배경 및 목표

문학 텍스트를 자동적으로 분석(automate content analysis)하는 기존의 방법론들이 있다. 대표적인 것은 'Named entity extraction'과 'Topic modeling'이 있다. 그러나 이들 방법론은 기본적으로 '단어를 세어' 통계량을 얻는 것이고, 명백한 한계를 갖는다. 예를 들면 "소설 한 페이지에 평균적으로 얼마나 긴 시간이 흐르는가?"와 같은 질문이 있다.

"wow, it's been thirty years but feels like yesterday"

Ted Underwood 는 위와 같은 문장을 제시하며, 이 장면에서 30년의 시간이 흘렀다고 계산해서는 안된다는 점을 지적하며, GPT-4를 사용한 시간 흐름 추정이 유의미하다는 연구를 제시한다[1]. 그러나 Underwood의 연구는 그 결과물이 디지털 문학 연구에 유효함을 입증했음에도, 여전히 개념 증명 단계에 머무르고 있다. GitHub에 소프트웨어 코드가 게시되어 있지만, 어디까지나 연구 결과의 재현을 위한 코드에 불과하다. 실제 연구에 즉각적으로 사용할 수 있는 상태가 아니라는 뜻이다. 이에 따라 본 연구는, Underwood의 연구 결과를 토대로, **'사용하기 편한 연구 도구로서의, 언어모델을 사용해 인간 연구자와 유사한 시간 흐름 추정 결과를 산출하는 파이썬 라이브러리'를 제작하여 배포하는 것을 목표로 한다.**

1.2. 유사 연구 분석

기존의 자연어처리 생태계의 시간 계측 라이브러리들은 특정한 어휘를 찾는 것에 의존한다. 'Spark NLP'의 'dateMatcher'와 같은 도구는 명시적인 날짜 표현을 찾는 것으로 동작[1]하며, 행동 묘사로부터 시간 흐름을 추론하는 기능은 없다[2]. 'pyTLEX'와 같은 라이브러리는 연구자의 주석에 의존[2]한다[3].

---

[1] "DateMatcher and MultiDateMatcher extract *exact* & *normalized dates* from relative date-time phrases and convert these dates to a *provided date format*. DateMatcher can only extract one date per input document while MultiDateMatcher can multiple dates." [2]

[2] "**pyTLEX** is the Python library to perform temporal analysis of TimeML annotated texts."

한편 Underwood가 언급한 Gregory Yauney의 빈도 분석 기반의 시간 흐름 측정 연구[4]에서, 빈도 분석을 통해서도 문학사적 맥락에서 유효한 정확도의 시간 흐름 추정이 가능함을 밝힌 바 있다. 그러나 해당 연구의 시간 흐름 추정 정확도는 인간 독자와 LLM에 비해 현저히 떨어진다.

그러므로, 현 상황에서 pip 등의 패키지 매니저로 일괄 설치가 가능한, '문학 연구자를 위한 시간 흐름 측정 라이브러리'는 존재하지 않는 것으로 보인다.

1.3. 기대효과

문학 텍스트에서의 시간 흐름 계측 도구가 간단한 라이브러리로 제공되는 것으로, **시간 흐름과 관련된 새로운 인문학 연구 촉발에 기여할 것이다.**

새로운 연구는 멀리서 읽기의 거시적 방법론에서 벗어날 수 있는 도구로서 기능할 것이. 기존의 빈도 분석 기반의 연구 방법론의 정확도는 거시적인 수준에서의 분석에만 유효했으나, LLM에 기반한 시간 흐름 측정은 인간 연구자와 유사한 수준의 결과를 내놓을 수 있기 때문이다.

한편, 인문학 연구자에게 기술적 장벽이었던 API 연동, 프롬프트 엔지니어링, 데이터 파이프라인 설계 등을 허물어 준다. 낮아진 기술적 장벽으로 인해 새로운 연구가 촉발되리라 기대한다.

2. 연구 추진 계획

2.1. 시스템 설계

본 시스템은 철저하게 모듈화 되어 구축될 예정이다. 구체적으로는 다음 다섯 모듈을 모두 구축하는 것을 목표로 한다.

모듈 이름	기능
Preprocessor	사용자 입력 텍스트 정제. 불필요한 공백 제거, 비표준 문자 정규화. 텍스트 분할.
Prompter	사용자 정의 프롬프트 템플릿 저장. 시스템 메시지, 프롬프트 등의 입력 인터페이스 제공.
Gateway	외부 LLM API 통신 추상화 제공. API 키 관리, HTTP 요청 전송, 오류처리 제공.
Parser	LLM 응답에 대한 안전한 파싱 수행. 예외처리 제공. 오류 검증 수행.
Schemas	사용자 정의 출력 구성 기능 제공. 다루기 쉽게 잘 구조화 된, 깨끗한 캡슐로 데이터 출력. (Pandas DataFrame 등의 데이터 분석 도구에 즉시 통합될 수 있는 출력 형식 지정 가능.)

그리하여 Underwood가 제시한 데이터 흐름의 파이프라인을 높은 수준으로 추상화 한 단일 시스템을 구축한다. 연구자는 본 시스템이 외부에 노출한 몇 개의 함수와 변수를 조작하

는 것으로, 간편하게 문학 텍스트의 시간 흐름을 추정할 수 있게 되어야 한다.

한편으로는 각 모듈의 작동 방식을 얼마든지 수정할 수 있어야 한다. 연구자의 역량에 따라 얼마든지 다른 시스템이나 모듈과 연동할 수 있도록, 확장성 있는 구조를 갖추어야 한다는 뜻이다.

2.2. 시스템 검증

2.2.1. 재현성 검증

본 시스템이 Underwood의 실험 결과(r = .68)에 근접한 결과를 산출하는지 검증한다. Underwood가 제공한 'outputGPT4.tsv' 파일[5]의 GPT-4 추정치와, 본 시스템의 추정치가 통계적으로 구별할 수 없어야 한다. 이를 충족하지 못했다면, 시스템 설계 또는 구현 단계에서 치명적인 오류가 있었다고 간주할 수 있다.

2.2.2. 오류 시험

사용하기 편한 연구 도구로서의 라이브러리를 표방하므로, 다양한 오류에 대한 적절한 처리를 지원해야 한다. 구체적으로는 다음 항목들에 대한 적절한 대처가 준비되어 있는지, 각 대응 코드가 정상 동작 하는지 검증한다.

- 빈 문자열 입력.
- 극도로 긴 문자열 입력.
- 네트워크 오류.
- 유효하지 않은 API 키.

2.3. 연구 일정표

9월 말일까지 개발 완료를 목표로 한다.

연구 주차	연구 내용 및 성과물
9월 2주차	Preprocessor, Prompter, Gateway 모듈 구현. 잘 작동하는 모듈 파일 산출.
9월 3주차	Parser, Schemas 모듈 구현. 다섯 모듈을 통합. 잘 작동하는 단일 소프트웨어(클래스) 산출.
9월 4주차	시스템 재현성 검증 및 오류 시험 수행. Underwood의 연구를 완전히 재현한 소프트웨어 산출.
9월 5주차	PIP 저장소 등록 및 연구 공개.

3. 후속 연구 제안

정상적으로 연구가 완료되었다면, 본 라이브러리의 지원 언어를 확장하는 방식으로 후속 연구를 진행할 수 있다.

시간 흐름과 관련된 측정 결과를 시각화 하는 모듈을 추가할 수 있다. 오직 문학 텍스트 속 시간 흐름 시각화만을 위해 준비된 Matplotlib 코드를 포함하는 방식으로 구현을 진행하는 것을 제안한다. 구체적으로는 각 페이지 별 시간 흐름, 각 문단별 시간 흐름 등을 시각화 가능하다.

참고문헌

[1] Underwood, T. (2023, March 19). "Using GPT-4 to measure the passage of time in fiction." https://tedunderwood.com/2023/03/19/using-gpt-4-to-measure-the-passage-of-time-in-fiction/

[2] Gregory Yauney. (2019). "Computational Prediction of Elapsed Narrative Time". *New Literary History*. 85.2. 351-65. https://gyauney.github.io/papers/elapsed-narrative-time.pdf

[3] “pyTLEX”, *Cognition, Narrative, and Culture Laboratory (Cognac Lab),* https://cognac.cs.fiu.edu/pytlex/

[4] Saglamlar, Halil. (2023, Jan 28). “Extracting Dates From Text Using Spark NLP”. *Medium*. https://medium.com/john-snow-labs/spark-nlp-datematcher-multidatematcher-1f1c37970f90

[5] Underwood, T. (2023, March 19). https://github.com/tedunderwood/fictional-time-with-GPT4