EHRKit: A Python Natural Language Processing Toolkit for Electronic Health Record Texts

Irene Li, Keen You, Xiangru Tang, et al., Dragomir Radev

LILY Lab, Yale University, New Haven, CT

LILY Lab

Motivation

The Electronic Health Record (EHR) is an essential part of the modern medical system and impacts healthcare delivery, operations, and research. Unstructured text is attracting much attention despite structured information in the EHRs and has become an exciting research field. The success of the recent neural Natural Language Processing (NLP) method has led to a new direction for processing unstructured clinical notes. In this work, we create a python library for clinical texts, EHRKit. This library contains two main parts: MIMIC-III-specific functions and tasks specific functions. The first part introduces a list of interfaces for accessing MIMIC-III NOTEEVENTS data, including basic search, information retrieval, and information extraction. The second part integrates many third-party libraries for up to 12 off-shelf NLP tasks such as named entity recognition, summarization, machine translation, etc.

System Design and Architecture

We show our EHRKit architecture in Fig. 1. It consists of two modules, namely MIMIC-III Tasks and Wrapper Functions.

- MIMIC-III Tasks

We include some basic NLP functions for MIMICIII NOTEEVENTS text data (Johnson et al., 2016), including basic statistical functions, Information Extraction, Keyword Search, Document Retrieval, Extractive and Abstractive Summarization. A big advantage of this module is to load the original NOTEEVENTS data (this is in free text) as a more structured data source.

- Wrapper Functions

This module integrates many third-party libraries and supports up to 12 functionalities for any freetext inputs.

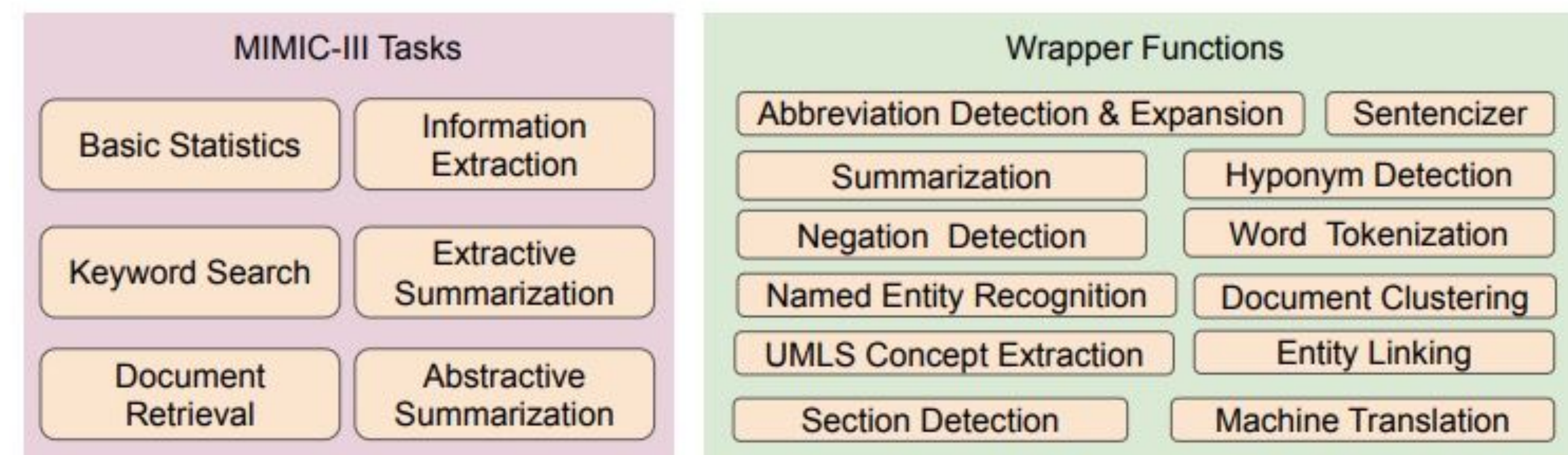

Table 1. EHRKit Architecture.

	MIMIC Related	Neural Methods	Machine Translation	Summarization
MIMIC-Extract	✓			
ScispaCy		✓		
medspaCy		✓		
Stanza Biomed		✓		
medspaCy		✓		
SciFive		✓		✓
EHRKit (ours)	✓	✓	✓	✓

Table 2. A comparison with other similar python toolkit.

	PubMed			MIMIC-CXR		
	R-1	R-2	R-L	R-1	R-2	R-L
Pegasus (Zhang et al., 2020)	45.97	20.15	28.25	65.11	**52.90**	**61.88**
BigBird (Zaheer et al., 2020)	46.32	**20.65**	**42.33**	63.85	51.09	60.55
BART (Lewis et al., 2019)	44.16	20.28	36.80	62.09	49.02	58.65
SciFive (Phan et al., 2021)	**48.83**	15.81	37.06	**65.17**	52.45	61.80

Table 3. Summarization evaluation: we evaluate selected models and report ROUGE-1, ROUGE-2 and ROUGE-L.

Dataset	Train	Valid	Test
PubMed	112K	6.6K	6.7K
MIMIC-CXR	91,544	2000	600

Table 3. Data statistics for PubMed and MIMIC-CXR summarization datasets. Words are counted before tokenization.

Performance Evaluation: Summarization

Summarization corpora from the clinical scenario are very challenging to be obtained, we chose the existing PubMed (Cohan et al., 2018) and MIMIC-CXR (Johnson et al., 2019) as our principal datasets.

PubMed dataset consists of 133k biomedical scientific publications from the PubMed database. MIMIC-CXR is a de-identified, Protected Health Information removed dataset of chest radiographs, with a DICOM format and free-text radiology reports. We use a subset from the MIMIC-CXR Dataset for the MEDIQA 2021 Radiology report summarization shared task.

We evaluate selected pretrained abstractive methods using ROUGE (Lin, 2004). Among the four models, we can observe that SciFive has a high R-1 score, but BigBird (Zaheer et al., 2020) and Pegasus (Zhang et al., 2020) achieve a better score on R-2 and R-L, respectively, on the two datasets. This evaluation shows that it is challenging to determine which model is the best in our specific scenario, though SciFive was pretrained for this purpose. In the future, more work can be done to improve automatic summarization for biomedical and clinical texts.

Conclusion

In this work, we propose a python library for clinical texts, EHRKit. This toolkit contains two main components: general API functions and MIMICspecific functions. In the future, we will investigate more EHR-NLP tasks including machine translation for more languages, multi-document summarization and question answering (Li et al., 2021b). Besides, we plan to investigate better-performed NLP models for these tasks, for example, BERTbased models (Lee et al., 2020; Li et al., 2021c) and graph-based models (Li et al., 2020, 2021d,a)