The Journal of Open Source Software

FrESCO: Framework for Exploring Scalable Computational Oncology

Adam Spannaus [1¶], **John Gounley** [1], **Mayanka Chandra Shekar** [1], **Zachary R. Fox** [1], **Jamaludin Mohd-Yusof** [2], **Noah Schaefferkoetter** [1], **and Heidi A. Hanson** [1]

1 Oak Ridge National Laboratory, Oak Ridge, TN, United States of America **2** Los Alamos National Laboratory, Los Alamos, NM, United States of America ¶ Corresponding author

DOI: 10.21105/joss.05345

Software

- Review
- Repository
- Archive

Editor: Jacob Schreiber

Reviewers:

- @gabeerion
- @anupamajha1

Submitted: 16 March 2023
Published: 11 September 2023

Statement of Need

The National Cancer Institute (NCI) monitors population level cancer trends as part of its Surveillance, Epidemiology, and End Results (SEER) program. This program consists of state or regional level cancer registries which collect, analyze, and annotate cancer pathology reports. From these annotated pathology reports, each individual registry aggregates cancer phenotype information from electronic health records. This data is then used to create summary statistics about cancer incidence and mortality to facilitate population health monitoring. Extracting phenotypic information from these reports is a labor intensive task, requiring specialized knowledge about the reports and cancer. Automating the information extraction process from cancer pathology reports has the potential to improve data quality by extracting information in a consistent manner across registries. It can also improve patient outcomes by reducing the time from diagnosis, enabling rapid case ascertainment for clinical trials. Here we present FrESCO, a modular deep-learning natural language processing (NLP) library initially designed for extracting pathology information from clinical text documents. This repository is not solely limited to clinical medical text, but may also be used by researchers just getting started with NLP methods and those looking for a robust solution for their classification problems.

State of the Field

Other software to meet the demanding challenges of bringing ML to biomedical studies have emerged in recent years. Monai (Cardoso et al., 2022) is oriented towards ML on medical imaging data and FuseMedML (Golts et al., 2023) creates general and multimodal data structures that are useful for biomedical ML. Most similar to FrESCO is PyHealth (Zhao et al., 2021) though it is more broadly scoped, focusing on MIMIC (Medical Information Mart for Intensive Care), electronic intensive care unit (eICU), and observational medical outcomes partnership common data model (OMOP-CDM) databases. Biomedical libraries such as Med7 (Kormilitzin et al., 2021) and EHRkit (Li et al., 2022) focus on electronic health records in general and machine learning tasks such as named-entity recognition and document summarization. Our FrESCO library is singularly focused on cancer pathology reports and provides the model building workflow for auto-coding SEER pathology reports, which is a fundamental requirement in a clinical deployment environment (Harris et al., 2022).

Summary

The FrESCO codebase provides a deep-learning Python package based on PyTorch (Paszke et al., 2019) for extracting information from clinical text. While the software is designed for clinical tasks, it may also be used for typical NLP tasks such as sentiment classification. Our flexible and modular codebase provides independent modules for: (1) loading text data and creating data structures, (2) building and training deep-learning models, and (3) scoring and evaluating trained models. Provided within the code repository are three model architectures to classify text data:

1. the multi-task convolutional neural network (MTCNN) of (Alawad et al., 2020),
2. the hierarchical self-attention network (HiSAN) described in (Gao et al., 2019), and
3. the case-level context model (CLC) of (Gao et al., 2020) for hierarchical datasets.

Each of these models is available with the deep-abstaining classifier (DAC) of (Thulasidasan et al., 2019), which is presently only available as part of the CANDLE code repository (Institute, 2023). The DAC adds an additional "abstention" class to the specified model so that the classifier may choose none of the available labels for a given task. While each model may work on generic data, the HiSAN and CLC architectures were specifically designed to work with patient data and are not available in other software packages like PyHealth (Zhao et al., 2021). As an example, the CLC model uses multiple pathology reports linked to an individual patient in a hierarchical way. We have adapted the FrESCO codebase from our workflow within an airgapped system which uses patient health data that is not publicly available. This is the same tool we use internally, aside from internal consistency checks, we are making it publicly available to work with user supplied text data, the only requirement being the format of the data files, which is specified in the `README`.

We have intentionally written this library with a working knowledge of Python as the only prerequisite. Those who are just getting started or are experienced NLP researchers or practitioners will find the code easy to understand and expand upon. For example, one may create a state-of-the-art NLP model by simply editing the configuration file, without touching a line of code. Lastly, as the model definitions are independent modules, one may experiment with their own custom model definitions within the training and evaluation framework developed herein.

Acknowledgements

This manuscript has been authored by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the US Department of Energy (DOE). The US government retains and the publisher, by accepting the article for publication, acknowledges that the US government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this manuscript, or allow others to do so, for US government purposes. DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

References

Alawad, M., Gao, S., Qiu, J. X., Yoon, H. J., Blair Christian, J., Penberthy, L., Mumphrey, B., Wu, X.-C., Coyle, L., & Tourassi, G. (2020). Automatic extraction of cancer registry reportable information from free-text pathology reports using multitask convolutional neural networks. *Journal of the American Medical Informatics Association*, *27*(1), 89–98. https://doi.org/10.1093/jamia/ocz153

Cardoso, M. J., Li, W., Brown, R., Ma, N., Kerfoot, E., Wang, Y., Murrey, B., Myronenko, A., Zhao, C., Yang, D., & others. (2022). MONAI: An open-source framework for deep learning in healthcare. *arXiv Preprint arXiv:2211.02701*. https://doi.org/10.48550/arXiv.2211.02701

Gao, S., Alawad, M., Schaefferkoetter, N., Penberthy, L., Wu, X.-C., Durbin, E. B., Coyle, L., Ramanathan, A., & Tourassi, G. (2020). Using case-level context to classify cancer pathology reports. *PLoS One*, *15*(5), e0232840. https://doi.org/10.1371/journal.pone.0232840

Gao, S., Qiu, J. X., Alawad, M., Hinkle, J. D., Schaefferkoetter, N., Yoon, H.-J., Christian, B., Fearn, P. A., Penberthy, L., Wu, X.-C., Coyle, L., Tourassi, G., & Ramanathan, A. (2019). Classifying cancer pathology reports with hierarchical self-attention networks. *Artificial Intelligence in Medicine*, *101*, 101726. https://doi.org/10.1016/j.artmed.2019.101726

Golts, A., Raboh, M., Shoshan, Y., Polaczek, S., Rabinovici-Cohen, S., & Hexter, E. (2023). FuseMedML: A framework for accelerated discovery in machine learning based biomedicine. *Journal of Open Source Software*, *8*(81), 4943. https://doi.org/10.21105/joss.04943

Harris, S., Bonnici, T., Keen, T., Lilaonitkul, W., White, M. J., & Swanepoel, N. (2022). Clinical deployment environments: Five pillars of translational machine learning for health. *Frontiers in Digital Health*, *4*. https://doi.org/10.3389/fdgth.2022.939292

Institute, N. C. (2023). ECP-candle. In *CANDLE Exascale Computing Program Application*. https://github.com/ECP-CANDLE; GitHub.

Kormilitzin, A., Vaci, N., Liu, Q., & Nevado-Holgado, A. (2021). Med7: A transferable clinical natural language processing model for electronic health records. *Artificial Intelligence in Medicine*, *118*, 102086. https://doi.org/10.1016/j.artmed.2021.102086

Li, I., You, K., Tang, X., Qiao, Y., Huang, L., Hsieh, C.-C., Rosand, B., & Radev, D. (2022). Ehrkit: A python natural language processing toolkit for electronic health record texts. *arXiv Preprint arXiv:2204.06604*. https://doi.org/10.48550/arXiv.2204.06604

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., … Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Thulasidasan, S., Bhattacharya, T., Bilmes, J., Chennupati, G., & Mohd-Yusof, J. (2019). Combating label noise in deep learning using abstention. In K. Chaudhuri & R. Salakhutdinov (Eds.), *Proceedings of the 36th international conference on machine learning* (Vol. 97, pp. 6234–6243). PMLR. https://proceedings.mlr.press/v97/thulasidasan19a.html

Zhao, Y., Qiao, Z., Xiao, C., Glass, L., & Sun, J. (2021). Pyhealth: A python library for health predictive models. *arXiv Preprint arXiv:2101.04209*. https://doi.org/10.48550/arXiv.2101.04209