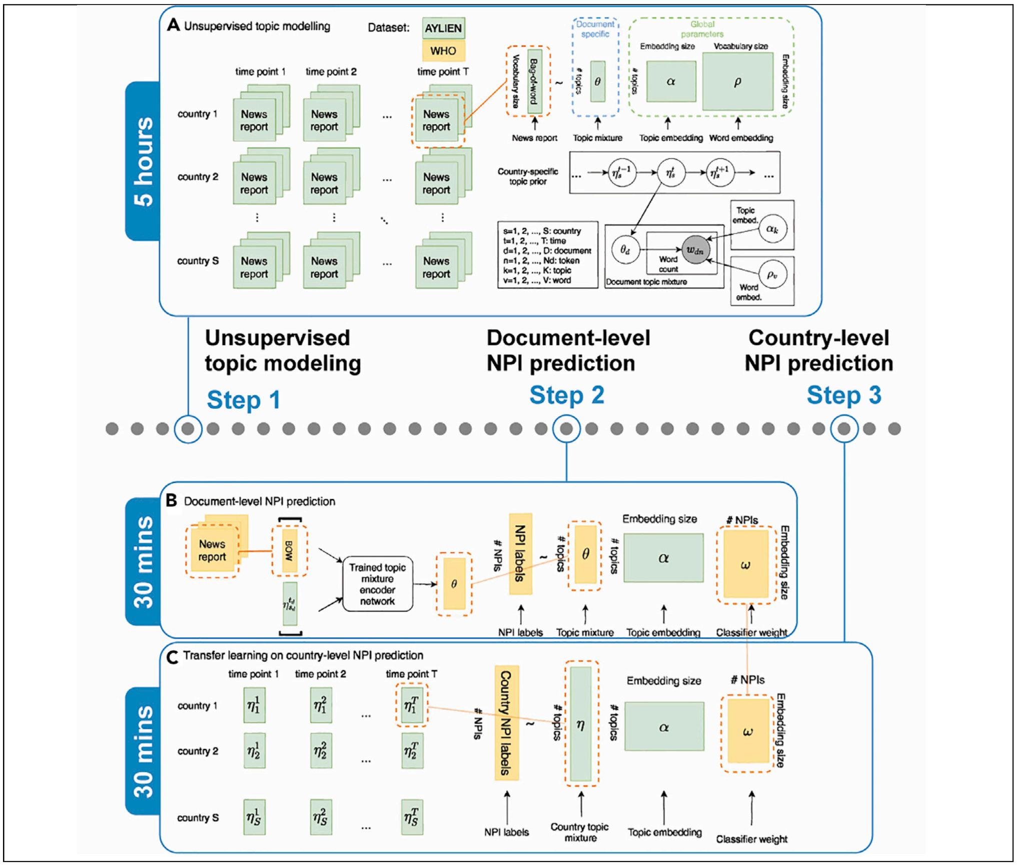
STAR Protocols

CellPress
OPEN ACCESS

Protocol

EpiTopics: A dynamic machine learning model to predict and inform non-pharmacological public health interventions from global news reports



Non-pharmacological interventions (NPIs) are important for controlling infectious diseases such as COVID-19, but their implementation is currently monitored in an ad hoc manner. To address this issue, we present a three-stage machine learning framework called EpiTopics to facilitate the surveillance of NPI. In this protocol, we outline the use of transfer-learning to address the limited number of NPI-labeled documents and topic modeling to support interpretation of the results.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Zhi Wen, Jingfu Zhang, Guido Powell, Imane Chafi, David L. Buckeridge, Yue Li

david.buckeridge@mcgill.ca (D.L.B.)
yueli@cs.mcgill.ca (Y.L.)

Highlights

Automated prediction of public health intervention from COVID-19 news reports

Inferring 42 country-specific temporal topic trends to monitor interventions

Learning interpretable topics that predict interventions from news reports

Transfer-learning to predict interventions for each country on weekly basis

Wen et al., STAR Protocols 3, 101463
June 17, 2022 © 2022 The Authors.
https://doi.org/10.1016/j.xpro.2022.101463

STAR Protocols

CellPress
OPEN ACCESS

Protocol

EpiTopics: A dynamic machine learning model to predict and inform non-pharmacological public health interventions from global news reports

Zhi Wen,[1,3] Jingfu Zhang,[1,3] Guido Powell,[2] Imane Chafi,[1] David L. Buckeridge,[2,*] and Yue Li[1,4,5,*]

[1]School of Computer Science, McGill University, Montreal, QC H3A 0G4, Canada

[2]School of Population and Global Health, McGill University, Montreal, QC, Canada

[3]These authors contributed equally

[4]Technical contact

[5]Lead contact

*Correspondence: david.buckeridge@mcgill.ca (D.L.B.), yueli@cs.mcgill.ca (Y.L.)
https://doi.org/10.1016/j.xpro.2022.101463

SUMMARY

Non-pharmacological interventions (NPIs) are important for controlling infectious diseases such as COVID-19, but their implementation is currently monitored in an ad hoc manner. To address this issue, we present a three-stage machine learning framework called EpiTopics to facilitate the surveillance of NPI. In this protocol, we outline the use of transfer-learning to address the limited number of NPI-labeled documents and topic modeling to support interpretation of the results.

For complete details on the use and execution of this protocol, please refer to Wen et al. (2022).

BEFORE YOU BEGIN

Protocol overview

This protocol will guide you through a series of steps to develop a machine learning model called EpiTopics. The method was developed to enable automatic detection from news reports of changes in the status of non-pharmacological interventions (NPI) for COVID-19. The method can be divided into 3 stages. At stage 1, EpiTopics learns country-dependent topics from a large number of COVID-19 news reports that do not have NPI labels (i.e., AYLIEN news dataset in Wen et al. (2022)). At stage 2, EpiTopics learns accurate connections between these topics and changes in NPI status from a set of labeled news reports (i.e., WHO news dataset in (Wen et al., 2022)). At stage 3, EpiTopics learns to predict country-dependent NPI changes by combining the knowledge learned from the previous two stages.

Acquiring datasets

Timing: 1 h

1. Download the WHO dataset from https://www.who.int/emergencies/diseases/novel-coronavirus-2019/phsm.

Optional: To replace the WHO dataset with other datasets of the user's choice, please ensure that the dataset of interest includes, for each sample, the text, the text's source location, the text's publication time, and the NPIs associated with the text.

Note: In addition, it is preferable to use datasets whose location and time coverages overlap significantly with the AYLIEN dataset, since only samples with overlapping locations and times can directly benefit from the topics learned during pre-training on the AYLIEN dataset.

2. Request access to the AYLIEN dataset on https://aylien.com/resources/datasets/coronavirus-dataset.

Optional: To replace the AYLIEN dataset with other datasets of the user's choice, please ensure that the dataset of interest includes, for each sample, the text, the text's source location, and the text's publication time.

Note: In addition, as this dataset is used for pre-training, generally it is preferable to use large datasets, for instance those that have more than 1 million training documents. Also, it is preferable to use datasets with significant location and time coverages overlap with the WHO dataset (i.e., the NPI-labeled documents).

Software installation

Timing: 2–4 h

3. Clone the code repository https://github.com/li-lab-mcgill/covid-npi.
4. Install packages according to the requirements file.

⚠ CRITICAL: Request to access the AYLIEN dataset might take days to be processed. We strongly recommend the usage of Graphical Processing Unit (GPU) Although it is not required, GPUs will greatly expedite training on a large corpus over CPUs. It is also desirable to have a virtual environment set up for this experiment.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
WHO dataset	https://www.who.int/emergencies/diseases/novel-coronavirus-2019/phsm	https://www.who.int/emergencies/diseases/novel-coronavirus-2019/phsm
AYLIEN dataset	https://aylien.com/resources/datasets/coronavirus-dataset	https://aylien.com/resources/datasets/coronavirus-dataset
Source code	(Wen et al., 2022)	https://github.com/li-lab-mcgill/covid-npi
Software and algorithms		
Python 3.6	https://python.org/downloads/	RRID: SCR_008394
absl-py 0.10.0	https://pypi.org/project/absl-py	https://pypi.org/project/absl-py
aiohttp 3.7.4	https://pypi.org/project/aiohttp	https://pypi.org/project/aiohttp
async-timeout 3.0.1	https://pypi.org/project/async-timeout	https://pypi.org/project/async-timeout
attrs 19.3.0	https://pypi.org/project/attrs	https://pypi.org/project/attrs
backcall 0.1.0	https://pypi.org/project/backcall	https://pypi.org/project/backcall
bleach 3.1.5	https://pypi.org/project/bleach	https://pypi.org/project/bleach
bokeh 2.0.2	https://pypi.org/project/bokeh	https://pypi.org/project/bokeh
cachetools 4.1.1	https://pypi.org/project/cachetools	https://pypi.org/project/cachetools
calmsize 0.1.3	https://pypi.org/project/calmsize	https://pypi.org/project/calmsize
captum 0.2.0	https://pypi.org/project/captum	https://pypi.org/project/captum
certifi 2020.4.5.2	https://pypi.org/project/certifi	https://pypi.org/project/certifi
chardet 3.0.4	https://pypi.org/project/chardet	https://pypi.org/project/chardet
click 7.1.2	https://pypi.org/project/click	https://pypi.org/project/click

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
configparser 5.0.1	https://pypi.org/project/configparser	https://pypi.org/project/configparser
country-list 0.1.5	https://pypi.org/project/country-list	https://pypi.org/project/country-list
cycler 0.10.0	https://pypi.org/project/cycler	https://pypi.org/project/cycler
decorator 4.4.2	https://pypi.org/project/decorator	https://pypi.org/project/decorator
defusedxml 0.6.0	https://pypi.org/project/defusedxml	https://pypi.org/project/defusedxml
docker-pycreds 0.4.0	https://pypi.org/project/docker-pycreds	https://pypi.org/project/docker-pycreds
dtw-python 1.1.6	https://pypi.org/project/dtw-python	https://pypi.org/project/dtw-python
entrypoints 0.3	https://pypi.org/project/entrypoints	https://pypi.org/project/entrypoints
epiweeks 2.1.2	https://pypi.org/project/epiweeks	https://pypi.org/project/epiweeks
et-xmlfile 1.0.1	https://pypi.org/project/et-xmlfile	https://pypi.org/project/et-xmlfile
fastdtw 0.3.4	https://pypi.org/project/fastdtw	https://pypi.org/project/fastdtw
fasttext 0.9.2	https://pypi.org/project/fasttext	https://pypi.org/project/fasttext
filelock 3.0.12	https://pypi.org/project/filelock	https://pypi.org/project/filelock
fsspec 0.8.4	https://pypi.org/project/fsspec	https://pypi.org/project/fsspec
future 0.18.2	https://pypi.org/project/future	https://pypi.org/project/future
gitdb 4.0.5	https://pypi.org/project/gitdb	https://pypi.org/project/gitdb
GitPython 3.1.9	https://pypi.org/project/GitPython	https://pypi.org/project/GitPython
google-auth 1.21.0	https://pypi.org/project/google-auth	https://pypi.org/project/google-auth
google-auth-oauthlib 0.4.1	https://pypi.org/project/google-auth-oauthlib	https://pypi.org/project/google-auth-oauthlib
grpcio 1.31.0	https://pypi.org/project/grpcio	https://pypi.org/project/grpcio
idna 2.9	https://pypi.org/project/idna	https://pypi.org/project/idna
importlib-metadata 1.6.0	https://pypi.org/project/importlib-metadata	https://pypi.org/project/importlib-metadata
ipykernel 5.2.1	https://pypi.org/project/ipykernel	https://pypi.org/project/ipykernel
ipython 7.14.0	https://pypi.org/project/ipython	RRID: SCR_001658
ipython-genutils 0.2.0	https://pypi.org/project/ipython-genutils	https://pypi.org/project/ipython-genutils
ipywidgets 7.5.1	https://pypi.org/project/ipywidgets	https://pypi.org/project/ipywidgets
jdcal 1.4.1	https://pypi.org/project/jdcal	https://pypi.org/project/jdcal
jedi 0.17.0	https://pypi.org/project/jedi	https://pypi.org/project/jedi
jieba 0.42.1	https://pypi.org/project/jieba	https://pypi.org/project/jieba
Jinja2 2.11.2	https://pypi.org/project/Jinja2	https://pypi.org/project/Jinja2
joblib 0.14.1	https://pypi.org/project/joblib	https://pypi.org/project/joblib
jsonschema 3.2.0	https://pypi.org/project/jsonschema	https://pypi.org/project/jsonschema
jupyter-client 6.1.3	https://pypi.org/project/jupyter-client	RRID: SCR_018413
jupyter-core 4.6.3	https://pypi.org/project/jupyter-core	RRID: SCR_018416
kiwisolver 1.2.0	https://pypi.org/project/kiwisolver	https://pypi.org/project/kiwisolver
lmdb 0.98	https://pypi.org/project/lmdb	https://pypi.org/project/lmdb
marisa-trie 0.7.5	https://pypi.org/project/marisa-trie	https://pypi.org/project/marisa-trie
Markdown 3.2.2	https://pypi.org/project/Markdown	https://pypi.org/project/Markdown
MarkupSafe 1.1.1	https://pypi.org/project/MarkupSafe	https://pypi.org/project/MarkupSafe
matplotlib 3.2.1	https://pypi.org/project/matplotlib	RRID: SCR_008624
mistune 0.8.4	https://pypi.org/project/mistune	https://pypi.org/project/mistune
mkl-fft 1.0.15	https://pypi.org/project/mkl-fft	https://pypi.org/project/mkl-fft
mkl-random 1.1.0	https://pypi.org/project/mkl-random	https://pypi.org/project/mkl-random
mkl-service 2.3.0	https://pypi.org/project/mkl-service	https://pypi.org/project/mkl-service
multidict 5.1.0	https://pypi.org/project/multidict	https://pypi.org/project/multidict
mwparserfromhell 0.5.4	https://pypi.org/project/mwparserfromhell	https://pypi.org/project/mwparserfromhell
nbconvert 5.6.1	https://pypi.org/project/nbconvert	https://pypi.org/project/nbconvert
nbformat 5.0.6	https://pypi.org/project/nbformat	https://pypi.org/project/nbformat
nltk 3.5	https://pypi.org/project/nltk	https://pypi.org/project/nltk
notebook 6.0.3	https://pypi.org/project/notebook	https://pypi.org/project/notebook
numpy 1.18.1	https://pypi.org/project/numpy	RRID: SCR_008633
oauthlib 3.1.0	https://pypi.org/project/oauthlib	https://pypi.org/project/oauthlib
openpyxl 3.0.4	https://pypi.org/project/openpyxl	https://pypi.org/project/openpyxl
packaging 20.1	https://pypi.org/project/packaging	https://pypi.org/project/packaging

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
pandas 1.2.2	https://pypi.org/project/pandas	RRID: SCR_018214
pandocfilters 1.4.2	https://pypi.org/project/pandocfilters	https://pypi.org/project/pandocfilters
parso 0.7.0	https://pypi.org/project/parso	https://pypi.org/project/parso
pathtools 0.1.2	https://pypi.org/project/pathtools	https://pypi.org/project/pathtools
pexpect 4.8.0	https://pypi.org/project/pexpect	https://pypi.org/project/pexpect
pickleshare 0.7.5	https://pypi.org/project/pickleshare	https://pypi.org/project/pickleshare
Pillow 7.1.2	https://pypi.org/project/Pillow	https://pypi.org/project/Pillow
plotly 4.6.0	https://pypi.org/project/plotly	RRID: SCR_013991
prometheus-client 0.7.1	https://pypi.org/project/prometheus-client	https://pypi.org/project/prometheus-client
promise 2.3	https://pypi.org/project/promise	https://pypi.org/project/promise
prompt-toolkit 3.0.5	https://pypi.org/project/prompt-toolkit	https://pypi.org/project/prompt-toolkit
protobuf 3.13.0	https://pypi.org/project/protobuf	https://pypi.org/project/protobuf
psutil 5.7.2	https://pypi.org/project/psutil	https://pypi.org/project/psutil
ptyprocess 0.6.0	https://pypi.org/project/ptyprocess	https://pypi.org/project/ptyprocess
pyasn1 0.4.8	https://pypi.org/project/pyasn1	https://pypi.org/project/pyasn1
pyasn1-modules 0.2.8	https://pypi.org/project/pyasn1-modules	https://pypi.org/project/pyasn1-modules
pybind11 2.5.0	https://pypi.org/project/pybind11	https://pypi.org/project/pybind11
Pygments 2.6.1	https://pypi.org/project/Pygments	https://pypi.org/project/Pygments
pyparsing 2.4.7	https://pypi.org/project/pyparsing	https://pypi.org/project/pyparsing
pyrsistent 0.16.0	https://pypi.org/project/pyrsistent	https://pypi.org/project/pyrsistent
python-dateutil 2.8.1	https://pypi.org/project/python-dateutil	https://pypi.org/project/python-dateutil
pytorch-lightning 1.2.7	https://pypi.org/project/pytorch-lightning	https://pypi.org/project/pytorch-lightning
pytorch-memlab 0.1.0	https://pypi.org/project/pytorch-memlab	https://pypi.org/project/pytorch-memlab
pytz 2020.1	https://pypi.org/project/pytz	https://pypi.org/project/pytz
PyYAML 5.3.1	https://pypi.org/project/PyYAML	https://pypi.org/project/PyYAML
pyzmq 19.0.0	https://pypi.org/project/pyzmq	https://pypi.org/project/pyzmq
regex 2020.6.8	https://pypi.org/project/regex	https://pypi.org/project/regex
requests 2.24.0	https://pypi.org/project/requests	https://pypi.org/project/requests
requests-oauthlib 1.3.0	https://pypi.org/project/requests-oauthlib	https://pypi.org/project/requests-oauthlib
retrying 1.3.3	https://pypi.org/project/retrying	https://pypi.org/project/retrying
rsa 4.6	https://pypi.org/project/rsa	RRID: SCR_006095
sacremoses 0.0.43	https://pypi.org/project/sacremoses	https://pypi.org/project/sacremoses
scikit-learn 0.22.1	https://pypi.org/project/scikit-learn	RRID: SCR_002577
scipy 1.4.1	https://pypi.org/project/scipy	RRID: SCR_008058
seaborn 0.10.1	https://pypi.org/project/seaborn	RRID: SCR_018132
Send2Trash 1.5.0	https://pypi.org/project/Send2Trash	https://pypi.org/project/Send2Trash
sentencepiece 0.1.91	https://pypi.org/project/sentencepiece	https://pypi.org/project/sentencepiece
sentry-sdk 0.19.0	https://pypi.org/project/sentry-sdk	https://pypi.org/project/sentry-sdk
shortuuid 1.0.1	https://pypi.org/project/shortuuid	https://pypi.org/project/shortuuid
six 1.14.0	https://pypi.org/project/six	https://pypi.org/project/six
smmap 3.0.4	https://pypi.org/project/smmap	https://pypi.org/project/smmap
subprocess32 3.5.4	https://pypi.org/project/subprocess32	https://pypi.org/project/subprocess32
tensorboard 2.2.0	https://pypi.org/project/tensorboard	https://pypi.org/project/tensorboard
tensorboard-plugin-wit 1.7.0	https://pypi.org/project/tensorboard-plugin-wit	https://pypi.org/project/tensorboard-plugin-wit
terminado 0.8.3	https://pypi.org/project/terminado	https://pypi.org/project/terminado
testpath 0.4.4	https://pypi.org/project/testpath	https://pypi.org/project/testpath
tokenizers 0.8.0rc4	https://pypi.org/project/tokenizers	https://pypi.org/project/tokenizers
torch 1.5.0	https://pypi.org/project/torch	https://pypi.org/project/torch
torchmetrics 0.2.0	https://pypi.org/project/torchmetrics	https://pypi.org/project/torchmetrics
torchvision 0.6.0	https://pypi.org/project/torchvision	https://pypi.org/project/torchvision
tornado 6.0.4	https://pypi.org/project/tornado	https://pypi.org/project/tornado
tqdm 4.46.0	https://pypi.org/project/tqdm	https://pypi.org/project/tqdm
traitlets 4.3.3	https://pypi.org/project/traitlets	https://pypi.org/project/traitlets
transformers 3.0.0	https://pypi.org/project/transformers	https://pypi.org/project/transformers

(*Continued on next page*)

Continued

REAGENT or RESOURCE	SOURCE	IDENTIFIER
typing-extensions 3.7.4.2	https://pypi.org/project/typing-extensions	https://pypi.org/project/typing-extensions
urllib3 1.25.9	https://pypi.org/project/urllib3	https://pypi.org/project/urllib3
wandb 0.10.25	https://pypi.org/project/wandb	https://pypi.org/project/wandb
watchdog 0.10.3	https://pypi.org/project/watchdog	RRID: SCR_018355
wcwidth 0.1.9	https://pypi.org/project/wcwidth	https://pypi.org/project/wcwidth
webencodings 0.5.1	https://pypi.org/project/webencodings	https://pypi.org/project/webencodings
Werkzeug 1.0.1	https://pypi.org/project/Werkzeug	https://pypi.org/project/Werkzeug
widgetsnbextension 3.5.1	https://pypi.org/project/widgetsnbextension	https://pypi.org/project/widgetsnbextension
wikipedia2vec 1.0.4	https://pypi.org/project/wikipedia2vec	https://pypi.org/project/wikipedia2vec
yarl 1.6.3	https://pypi.org/project/yarl	https://pypi.org/project/yarl
zipp 3.1.0	https://pypi.org/project/zipp	https://pypi.org/project/zipp

STEP-BY-STEP METHOD DETAILS

Data preprocessing

Timing: 10 min

This section describes 1) The removal of white spaces, special characters and non-English words 2) The removal of stop words as in (Dieng et al., 2020) 3) The extraction of information that is relevant to us from AYLIEN and WHO datasets 4) The removal from WHO dataset of documents whose country or source are not observed in the AYLIEN data.

1. Preprocess AYLIEN data.
   a. modify the script run_data_process.sh to include the correct path to the AYLIEN dataset, stop words file, and country NPIs file.
   b. set ‘aylien_flag’ to 1 and ‘label_harm’ to 1.
   c. execute ‘run_data_process.sh’ from the command line.
2. Preprocess WHO data.
   a. modify the script run_data_process.sh to include the correct path to the WHO dataset, stop words file, and country NPIs file.
   b. set ‘label_harm’ to 1.
   c. execute ‘run_data_process.sh’ from the command line.
3. The program will store the processed data (e.g., bag-of-words) in the output directory specified by save_dir.
   a. Take note that this should also be the input directory for running MixMedia (Li et al., 2020) (see below).
   b. More specifically, check that the output directory contains:
      i. Text file that contains the mappings between labels and their ids.
      ii. Text file that contains countries and their assigned ids.
      iii. Time_stamps and their ids.
      iv. 43 pickle (.pkl) files that mainly feature the pickled vocabulary and embeddings and bag-of-word representations of tokens.

Running MixMedia

Timing: 5 h

Pretraining of the MixMedia (Li et al., 2020) framework on the larger AYLIEN dataset as part of our transfer learning scheme.

4. Modify the script run_MixMedia.sh.

a. set K = 25 (the number of desired topics).
b. set cuda = {the indices to the GPUs that are available to you}.
c. set dataset = "AYLIEN".
d. set datadir = path to your AYLIEN files.
e. set outdir = path to the output directory of your choice.
f. set wemb = path to the output directory of your choice, this contains the embeddings that are needed for stage 3.
g. set mode = "train".
h. set batch_size = "128".
i. set lr = "1e-3".
j. set epochs = "400".
k. set min_df = "10".
l. set train_embeddings = "1".
m. set eval_batch_size = "128".
n. set time_prior = "1".
o. set source_prior = "1".

Execute './run_MixMedia.sh' from the command line.

5. The program will save the outputs to a folder under save_path: save_path/<timestamp>, and
   a. The timestamp records the time this script starts to run, and is in the format of {month}-{day}-{hour}-{minute}.
   b. The program saves the trained model.
   c. The program saves the learned topics (e.g., the topic embedding α, the word embedding ρ, LSTM weights for topic prior η, etc).
6. Monitor the progress with Tensorboard or Weights & Biases by setting "logger".

Transfer learning for NPI prediction

Timing: 1 h

After MixMedia is trained on AYLIEN, we can use the learned topics for NPI prediction via transfer learning. This consists of three consecutive stages: inferring WHO documents' topic mixtures, training a classifier on document-NPI prediction, transferring the classifier to country-NPI prediction.

7. Infer WHO documents' topic mixtures.
   a. Populate the 'save_dir', 'data_dir' and 'model_dir' entries of the 'infer_theta.sh' file according to the instructions within the file.
   b. The program saves the output to a folder under save_dir: save_dir/{timestamp}, where the timestamp records the time this script starts to run, and is in the format of {month}-{day}-{hour}-{minute}.
   c. The program also saves the document topic mixtures theta.
8. Train a classifier on document-NPI prediction.
   a. Within classify_npi.sh, set 'mode' to zero-shot or finetune or from-scratch based on the type of result that currently needs to be reproduced.
      i. For document-level NPI prediction, set mode to "doc" and provide who_label_dir and theta_dir.
      ii. For zero-shot transfer, set mode to zero_shot and provide cnpi_dir and ckpt_dir;
      iii. For fine-tuning, set mode to finetune and provide cnpi_dir and ckpt_dir.
   b. Set eta_dir to the directory where you saved your outputs in step 5.
   c. Specify save_ckpt. When set, the program saves the results reported in Wen et al. (2022) to a subfolder under save_dir: save_dir/mode/{timestamp}

   i. The timestamp records the time this script starts to run, and is in the format of {month}-{day}-{hour}-{minute}.
   ii. For each random seed, the program saves a trained linear classifier and the corresponding test predictions, with suffixes in filenames that specify the seed.
   iii. The program also saves the aggregated results in AUPRC into a json file.

d. Repeat the above steps for the other modes.

EXPECTED OUTCOMES

The above commands will result in the following outcomes corresponding to Figure 1, Table 1, Table 2, Table 3 in Wen et al. (2022):

Figure 1: Learned topics and the top words under each topic. The sizes of the words are proportional to their topic probabilities. The background colors indicate the themes we gave to the topics.

Table 1: Area under the precision-recall curve (AUPRC) scores for document-level NPI prediction. The AUPRC scores are computed on individual NPIs, and then averaged without weighting (macro AUPRC) or weighted by NPIs' prevalence (weighted AUPRC). Both BOW+linear and BOW+feed-forward use the normalized word vector (i.e., bag of words or BOW) for each document to predict NPI label. All methods are each repeated 100 times with different random seeds. Values in the brackets are standard deviations over the 100 experiments.

Table 2: Area under the precision-recall curve (AUPRC) scores for country-level NPI prediction. Random baselines are each repeated 1000 times with different random seeds, and the rest are each repeated 100 times with different random seeds. Values in the brackets are standard deviations over the repeated experiments.

Table 3: AUPRC scores for country-level NPI prediction from topics at document and country level. Values in the brackets are standard deviations. Random baselines are each repeated 1000 times with different random seeds, and the rest are each repeated 100 times with random seeds.

All of the above will be saved to a subfolder under save_dir: save_dir/mode/{timestamp}.

QUANTIFICATION AND STATISTICAL ANALYSIS

Expected outcomes in this protocol are stochastic in nature, due to hardware, model initialization, etc. Uncertainty in the reported results is controlled and measured through repeated runs with different random seeds. For models involving training (i.e., except random baselines), the results are based on 100 runs, while results for random baselines are based on 1000 runs. To reduce the uncertainty, the user can choose to increase the number of runs subject to computational cost.

Additionally, because of the large size of AYLIEN dataset, the topic model is trained once, and therefore one set of learned topics is used throughout all subsequent experiments. The user can explore training multiple versions of the topic model using different random seeds to obtain multiple sets of topics. The user can then study the variation, or consistency, of learned topics across runs, and explore how variations in learned topics can impact NPI predictions.

LIMITATIONS

To begin, using different library versions may have an impact on the results. As a result, the program might run into errors, or the results might not be able to be exactly reproduced. Please ensure to follow requirements as closely as possible. Also, the datasets used in this protocol, i.e., AYLIEN and WHO, may be updated or removed after they were accessed in this protocol. This may lead to differences in the results, or that some results could not be reproduced. In addition, this protocol assumes the user has direct access to the computational infrastructure, not through a set of centralized computing clusters such as SLURM. To use the protocol in such scenarios, minimal changes need to be made. For example, the user can use the protocol in an interactive session. Please refer to the instructions of the specific computing system on how to modify the protocol. Finally, the type of computational resources has an impact on the results. As an example, the batch sizes and the model sizes entail a certain amount of memory, and the availability of GPUs impact the amount of time needed for training.

TROUBLESHOOTING

Problem 1

Incompatible library versions (before you begin - software installation).

Potential solution

It is best to install libraries in a virtual environment specifically created for this protocol. For instructing on managing virtual environments, please refer to https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html.

Follow the library versions in requirements as closely as possible.

Problem 2

The model with the provided hyperparameters cannot be reproduced due to GPU memory limits (any step).

Potential solution

If the user does not have enough GPU memory, the user can reduce the batch size. While doing so, in order to approximate the protocol as closely as possible, the user can accumulate gradients (i.e., calculate gradients without updating optimizer or model) across multiple batches to maintain the identical effective batch size. For example, the user can use a batch size of 64 and accumulate gradients of 2 consecutive batches to approximate an effective batch size of 128.

If the user needs to further reduce GPU memory usage, the user can reduce the model’s size, for example the numbers of layers or the hidden dimensions. Doing so would likely have a negative impact on performance.

Problem 3

AYLIEN or WHO data is updated or removed (before you begin - acquiring datasets).

Potential solution

If the AYLIEN or WHO dataset is updated to include more data, the user can retrieve the same version as in this protocol by filtering according to Wen et al. (2022). The user can also choose to use the newer version instead and obtain a model trained on a wider coverage.

If the dataset is removed, or the user wishes to obtain the exact same version as in this protocol for any other reason, the user can reach out to authors.

Problem 4

Data files or intermediate result files are not found or compatible (any step).

Potential solution

Check the paths given to the script and make sure that the files exist and they match the script’s configuration.

Problem 5

Training progress cannot be correctly logged (running MixMedia - step 4).

Potential solution

If the user is using Tensorboard as the logger, please follow the instructions here on using Tensorboard with PyTorch.

If the user is using Weights and Bias for logging, by default it requires internet connection. For logging locally, or other functionalities, please refer to the instructions here.

Problem 6

When using other custom datasets as alternatives to the WHO dataset, the NPI labels have an imbalanced distribution, resulting in poor performance on minority classes (transfer learning for NPI prediction – step 7).

Potential solution

To mitigate the issue of data imbalance and improve performance on minority classes, the user can apply several techniques. For instance, the model can be more heavily regularized via weight decay. Also, the user can assign different weights to different classes such that the loss incurred on minority classes is amplified.

Problem 7

When using other custom datasets as alternatives to the AYLIEN dataset for learning topics, the optimal number of topics changes (running MixMedia – step 4).

Potential solution

The optimal number of topics is usually specific to the dataset on which the model is trained, and therefore the user is advised to search for that number on new datasets. As an example, the user can search from 5 topics to 100 topics at an interval of 20, and then search within the best performing intervals using a small interval. The number of search steps is determined as a trade-off between the precision of the search and the compute budget.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources should be directed to and will be fulfilled by the lead contact, Yue Li (yueli@cs.mcgill.ca).

Materials availability

This study did not generate any reagents.

Data and code availability

All data and scripts of this protocol are publicly available on GitHub at https://github.com/li-lab-mcgill/covid-npi. An archived release (https://doi.org/10.5281/zenodo.6350810) can be found at https://github.com/li-lab-mcgill/covid-npi/releases/tag/v1.0.

ACKNOWLEDGMENTS

This work is supported by CIHR through the Canadian 2019 Novel Coronavirus (COVID-19) Rapid Research Funding Opportunity (Round 1) (application number: 440236).

AUTHOR CONTRIBUTIONS

Y.L. and D.L.B. conceived the study. Y.L. and Z.W. developed the model with critical help from D.L.B. and G.P. I.C. collected and processed the data. Z.W. implemented the model and ran the experiments. J.Z. experimented with the code and wrote the initial draft of the manuscript. Y.L. and D.L.B. supervised the project. All authors analyzed the results and wrote the final version of the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Dieng, A.B., Ruiz, F.J.R., and Blei, D.M. (2020). Topic modeling in embedding spaces. Trans. Assoc. Comput. Linguist. *8*, 439–453. https://doi.org/10.1162/tacl_a_00325.

Li, Y., Nair, P., Wen, Z., Chafi, I., Okhmatovskaia, A., Powell, G., Shen, Y., and Buckeridge, D. (2020). Global surveillance of COVID-19 by mining news media using a multi-source dynamic embedded topic model. In Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics, pp. 1–14. https://doi.org/10.1145/3388440.3412418.

Wen, Z., Powell, G., Chafi, I., Buckeridge, D.L., and Li, Y. (2022). Inferring global-scale temporal latent topics from news reports to predict public health interventions for COVID-19. Patterns *3*, 100435. https://doi.org/10.1016/j.patter.2022.100435.