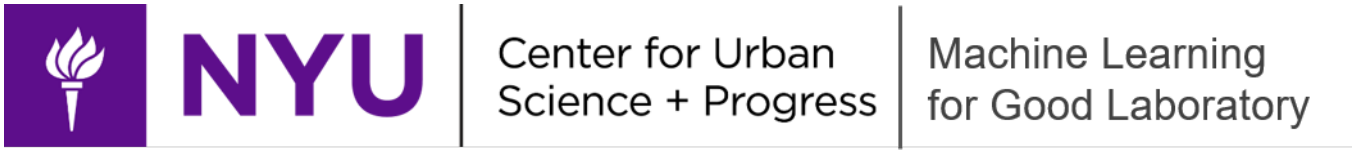
MUSES Open-Source Software for Pre-Syndromic Disease Surveillance

Daniel B. Neill, Ph.D.

Machine Learning for Good Laboratory
New York University

daniel.neill@nyu.edu

Joint work with Boyuan Chen (NYU),
Yi (Andy) Wei (NYU), and Mallory Nobles (CMU).



Our **MUSES Open-Source Software** is a freely available, open-source Python implementation of the Multidimensional Semantic Scan, our novel method for **pre-syndromic** disease surveillance.

Methodology is described in our recent publication:

M. Nobles, R. Lall, R.W. Mathes, and D.B. Neill*, *Science Advances* 8(44): eabm4920, 2022. DOI: 10.1126/sciadv.abm4920 (open access)

Code and documentation by Boyuan Chen, Yi Wei, Daniel B. Neill. Available from https://wp.nyu.edu/ml4good/pre-syndromic-surveillance.

Acknowledgements:

MUSES Open Source is derived from the original C code for MUSES by Mallory Nobles and Daniel B. Neill, with additional contributions from Kenton Murray, Abhinav Maurya, and Yandong Liu.

We wish to acknowledge Ramona Lall, Robert Mathes, and colleagues at the BCD Syndromic Surveillance Unit at NYC DOHMH for providing retrospective data and expert feedback that were highly valuable for our development of the MUSES methodology and open source software.

Our **MUSES Open-Source Software** is a freely available, open-source Python implementation of the Multidimensional Semantic Scan, our novel method for **pre-syndromic** disease surveillance.

MUSES is primarily geared toward **state and local health departments** who are already regularly collecting **Emergency Department chief complaint data** from hospitals in their jurisdiction.

To obtain maximum benefit from running MUSES, you should have:

- **Timely**, **representative**, and **high-quality** input data.
- Trained public health epidemiologist(s) who will **view** detection results regularly (e.g., daily), **respond** to relevant clusters, and **provide feedback** to continually enhance system performance.

MUSES is meant to **complement**, rather than replace, currently used systems for syndromic surveillance, providing users with **awareness of emerging threats** that other systems cannot detect.

Goals of today's training session

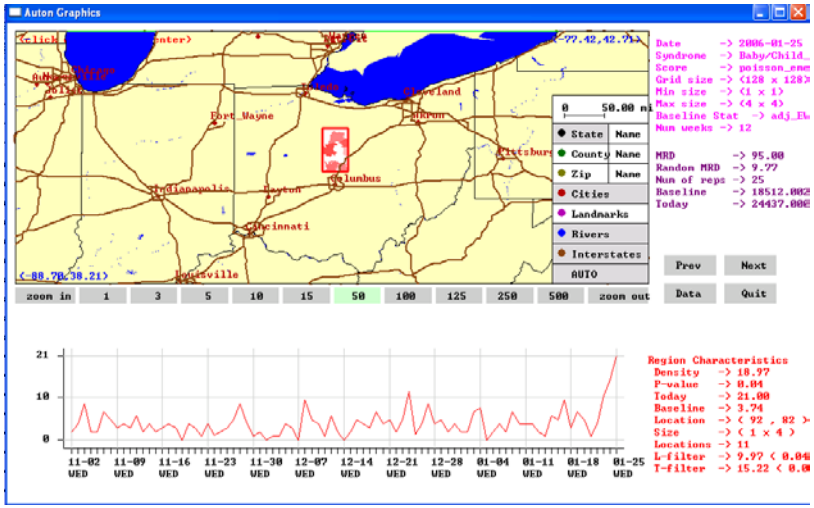
1. Understand the purpose and value of pre-syndromic surveillance, as distinct from syndromic surveillance.
2. Gain a high-level understanding of the methodology used for MUSES.
3. Download and install MUSES, and prepare data for analysis.
4. Run MUSES on sample (synthetic) data or on one's own collected data.
5. Visualize detected clusters using our graphical interface.
6. Provide feedback on newly discovered syndromes for MUSES "to monitor" or "to ignore" in future runs.
7. Interpret and use detection results.
8. Understand practical considerations and limitations for using MUSES in day-to-day practice.

Goals of today's training session

1. Understand the purpose and value of pre-syndromic surveillance, as distinct from syndromic surveillance.
2. Gain a high-level understanding of the methodology used for MUSES.
3. Download and install MUSES, and prepare data for analysis.
4. Run MUSES on sample (synthetic) data or on one's own collected data.
5. Visualize detected clusters using our graphical interface.
6. Provide feedback on newly discovered syndromes for MUSES "to monitor" or "to ignore" in future runs.
7. Interpret and use detection results.
8. Understand practical considerations and limitations for using MUSES in day-to-day practice.

Why pre-syndromic?

- **Syndromic surveillance** is effective for detecting case clusters that correspond to known types of illness.
  - Classify cases (e.g., ED chief complaints or OTC sales) into predefined syndromes (e.g., ILI).
  - For each syndrome, detect any unexpectedly high case counts, searching across space, time, and subgroups (spatial scan).



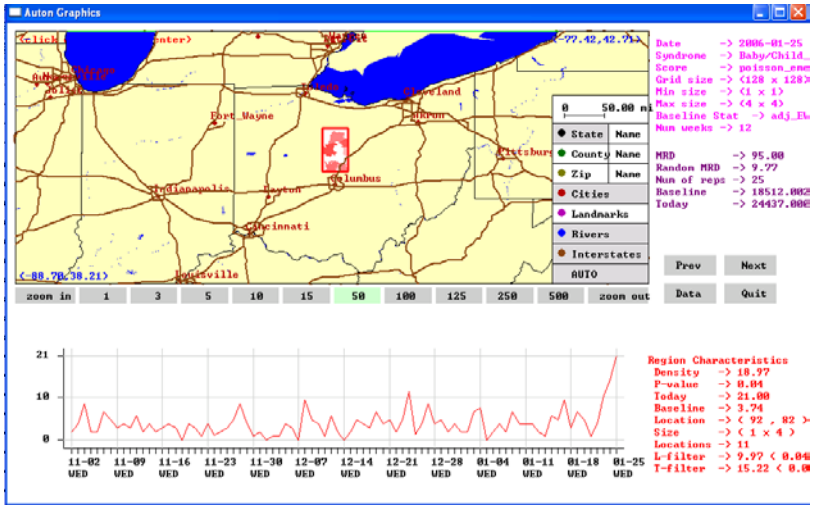
<u>Example</u>: A small, localized outbreak of GI illness was detected in OH by monitoring pediatric electrolyte sales.

How can we detect a newly emerging health threat with rare or previously unseen symptomology?

Existing methods would map these unusual cases into existing syndromes or ignore them, thus diluting the signal and delaying (or preventing) detection.

Why pre-syndromic?

- **Syndromic surveillance** is effective for detecting case clusters that correspond to known types of illness.
  - Classify cases (e.g., ED chief complaints or OTC sales) into predefined syndromes (e.g., ILI).
  - For each syndrome, detect any unexpectedly high case counts, searching across space, time, and subgroups (spatial scan).



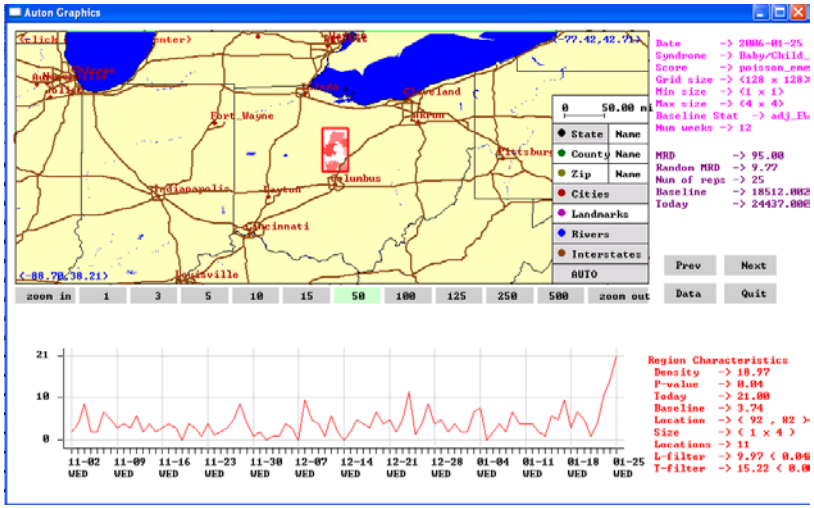
<u>Example</u>: A small, localized outbreak of GI illness was detected in OH by monitoring pediatric electrolyte sales.

How can we be aware of the many potentially relevant health-related events occurring every day?

<u>Common thread</u>: How can we detect what we're not already looking for? (How can we find case clusters that do not correspond to existing syndromes?)

Why pre-syndromic?

- **Syndromic surveillance** is effective for detecting case clusters that correspond to known types of illness.
  - Classify cases (e.g., ED chief complaints or OTC sales) into predefined syndromes (e.g., ILI).
  - For each syndrome, detect any unexpectedly high case counts, searching across space, time, and subgroups (spatial scan).



<u>Example</u>: A small, localized outbreak of GI illness was detected in OH by monitoring pediatric electrolyte sales.

Pre-syndromic surveillance is a **safety net** that can supplement existing syndromic surveillance systems by alerting public health to unusual or newly emerging threats that existing systems would fail to detect.

Goals of today's training session

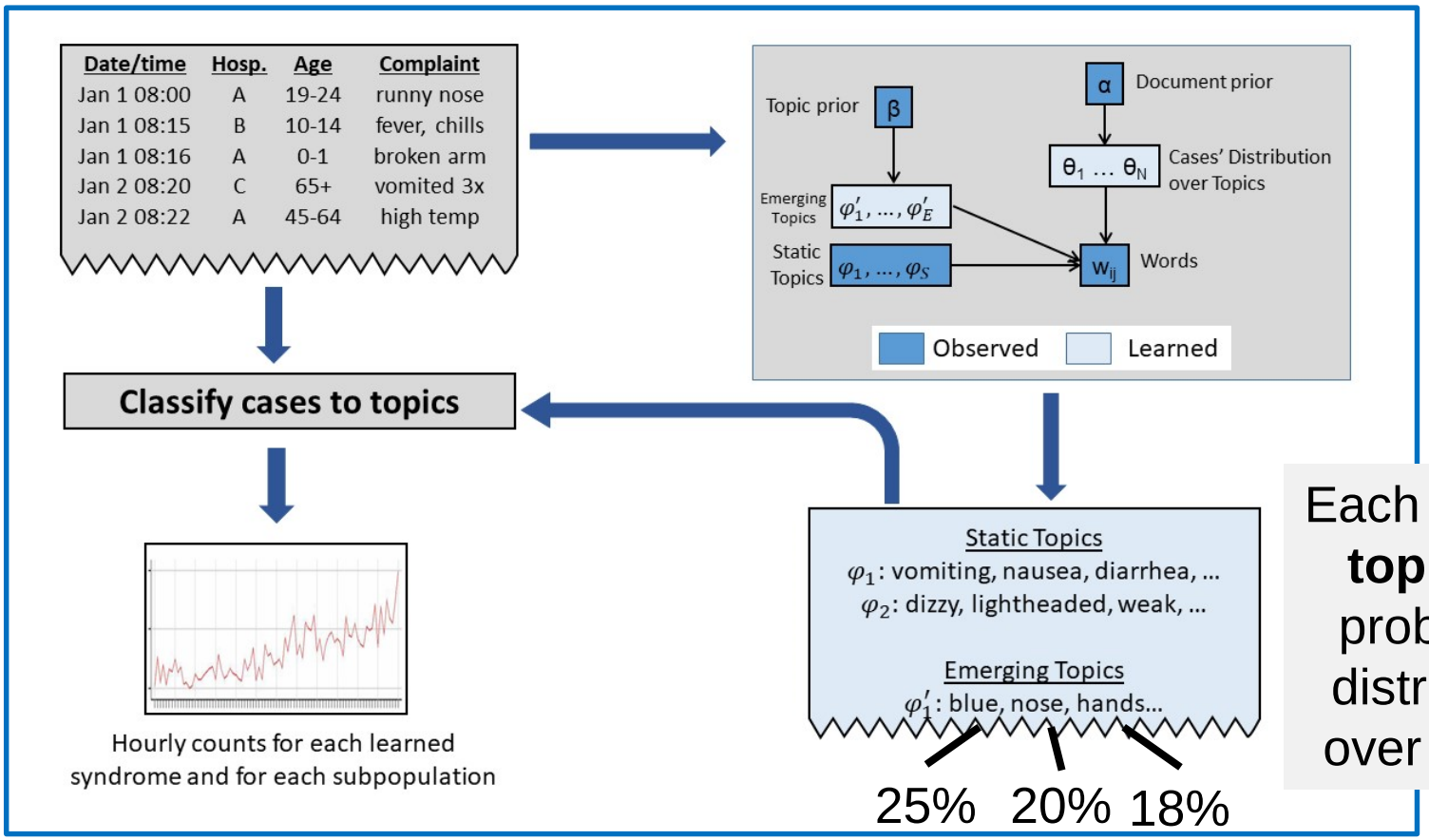
1. Understand the purpose and value of pre-syndromic surveillance, as distinct from syndromic surveillance.
2. Gain a high-level understanding of the methodology used for MUSES.
3. Download and install MUSES, and prepare data for analysis.
4. Run MUSES on sample (synthetic) data or on one's own collected data.
5. Visualize detected clusters using our graphical interface.
6. Provide feedback on newly discovered syndromes for MUSES "to monitor" or "to ignore" in future runs.
7. Interpret and use detection results.
8. Understand practical considerations and limitations for using MUSES in day-to-day practice.

MUSES methodology- overview

Four main stages of analysis:

1. Preprocessing and data cleaning
   - Spell-checking and standardization
   - Expanding ICD codes into text, e.g., “V9542X” → “forced landing of spacecraft injuring occupant”
2. Identifying newly emerging syndromes from text
   - Contrastive topic modeling
3. Detecting case clusters
   - Multidimensional spatial scan
4. Visualization and incorporating user feedback
   - Users can add new syndromes and specify if they would like MUSES to monitor or ignore them in the future.
   - This enables the system to report more relevant clusters and fewer irrelevant “false positives”.

MUSES methodology- overview

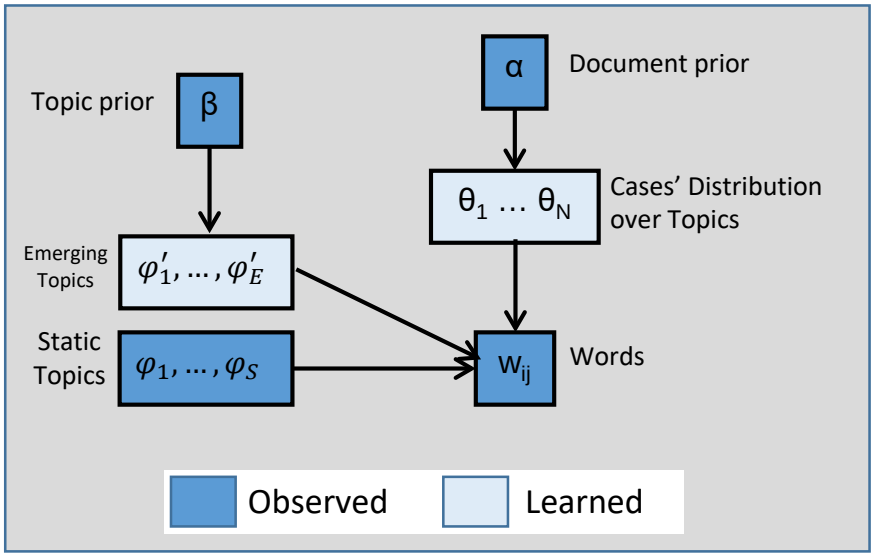


Each learned **topic** is a probability distribution over words.

MUSES learns newly emerging syndromes directly from text (ED chief complaints) using a new type of **probabilistic topic modeling**.

MUSES learns two sets of topics

- **Static Topics**
  - Designed to capture common syndromes like ILI.
  - Learned over a large set of historical data using a standard topic model (LDA).
- **Emerging Topics**
  - Capture rare diseases or novel events that are not well explained by the static topics.
  - Learned over the most recent data using a new approach.



Contrastive topic modeling answers the question, "what's going on the recent data that we typically haven't seen in the past".

Multidimensional spatial scanning

Once the data has been structured by the topic modeling step, we now have a set of learned topics (e.g., 25 static and 25 emerging topics) and have classified each case to one of these topics.

We now look at all potential **emerging case clusters** defined by:

- A given time window (typically, 1-3 hours)
- One of the emerging topics (we typically do not scan over static topics)
- One or more hospitals
- Demographics (e.g., age range)

For each potential cluster S defined in this way, we compare the **observed** number of cases C(S) with the **expected** number of cases B(S).

The **score** F(S) is an expectation-based Poisson log-likelihood ratio statistic: that is, high scores mean that the observed number of cases is much higher than expected.

Goals of today's training session

1. Understand the purpose and value of pre-syndromic surveillance, as distinct from syndromic surveillance.
2. Gain a high-level understanding of the methodology used for MUSES.
3. Download and install MUSES, and prepare data for analysis.
4. Run MUSES on sample (synthetic) data or on one's own collected data.
5. Visualize detected clusters using our graphical interface.
6. Provide feedback on newly discovered syndromes for MUSES "to monitor" or "to ignore" in future runs.
7. Interpret and use detection results.
8. Understand practical considerations and limitations for using MUSES in day-to-day practice.

Download and installation

System Requirements

- Python 3.8 or above.
- Scripts for preprocessing, topic modeling and spatial scan can run on Windows, MacOS, and Linux. However, the graphical user interface for visualization can only run on Windows.

Code on Github

https://github.com/danielbneill/pre-syndromic-surveillance
(Also accessible via wp.nyu.edu/ml4good/pre-syndromic-surveillance)

- BSD (3-clause) open source license – freely available, can modify
- Full documentation in PDF file
- Top-level code directory
  - requirements.txt file
  - synthetic data for sample run: chief complaints sampled at random from ICD lookup table, plus a “novel health threat” injected into the foreground data.
  - code directories (preprocessing, topic_modeling, visualization)

Download and installation

System Requirements

- Python 3.8 or above.
- Scripts for preprocessing, topic modeling and spatial scan can run on Windows, MacOS, and Linux. However, the graphical user interface for visualization can only run on Windows.

Installation from command line

> git clone https://github.com/danielbneill/pre-syndromic-surveillance

[Or download and unzip from GitHub page.]

> cd "pre-syndromic-surveillance/MUSES Open Source Software Code"

> pip install -r requirements.txt [This installs all necessary packages.]

Sample script to run MUSES on synthetic data

> cd .. [Now in directory: pre-syndromic-surveillance.]

> sample_run_for_synthetic_data.bat

[I will walk you through each line of this file and how to modify it for your own data.]

Preparing data

Two main files:

- Background (historical) data used to learn static topics
- Foreground (recent) data for detecting emerging clusters

(These can be the same file - then different date ranges would be used.)

Required attributes: cc, time, date (MM/DD/YY), hospcode, agegroup
Optional attributes: icd, VISITID, sex
(Column names can be different, and/or in a different order. We'll specify this in the "attributes_in_sequence" parameter of our pre-processing script.)
(Hint: if you don't have a required attribute like agegroup in your data, Just create a new column with the same value, like "x", for everyone.)

date	time	hospcode	agegroup	cc
8/15/2021	23:51:00	A	18-64	unspecified monoarthritis ankle and foot
1/24/2021	12:03:00	A	00-17	miosis persistent not due to miotics
8/21/2021	10:28:00	A	18-64	open fracture of unspecified part of femur
11/14/2021	0:23:00	B	00-17	allergy to other foods

Preparing data

Two main files:

- Background (historical) data used to learn static topics
- Foreground (recent) data for detecting emerging clusters

(These can be the same file - then different date ranges would be used.)

Two additional files for the scanning step:

- Age groups
- Hospital codes

(Each line of the file is a comma-separated list to be searched over.)

Typically, we search over a single hospital at a time, so the hospital file just consists of all hospitals in the data, one per line. For the synthetic data:

A
B
C

If we had three age categories (00-17, 18-64, 65-99), we might search all non-empty subsets:

00-17
18-64
65-99
00-17, 18-64
00-17, 65-99
18-64, 65-99
00-17, 18-64, 65-99

Goals of today's training session

1. Understand the purpose and value of pre-syndromic surveillance, as distinct from syndromic surveillance.
2. Gain a high-level understanding of the methodology used for MUSES.
3. Download and install MUSES, and prepare data for analysis.
4. Run MUSES on sample (synthetic) data or on one's own collected data.
5. Visualize detected clusters using our graphical interface.
6. Provide feedback on newly discovered syndromes for MUSES "to monitor" or "to ignore" in future runs.
7. Interpret and use detection results.
8. Understand practical considerations and limitations for using MUSES in day-to-day practice.

Preprocessing the background data

> cd "MUSES Open Source Software Code/preprocessing"

> python preprocessing.py

--functionality=clean_data

--input_file=../data/synthetic_data/synthetic_data_2021.csv [Point to your background data file.]

--sep="," [This is for comma-separated (CSV) files. Change to "\t" for tab-separated.]

--attributes_in_sequence="date time hospcode agegroup cc" [Describe each column in your dataset, in order, as one of {cc, time, date, agegroup, hospcode, icd, VISITID, sex, x}, where x means ignore.]

--search_group_attributes="hospcode agegroup"

--search_group_file_names="../data/synthetic_data/Settings/single_hospcode_searchgroups.csv ../data/synthetic_data/Settings/age_search_groups_all.csv" [Point to the hospcode and agegroup files.]

--icd_map_file="./dicts/icd9_to_text_list.txt" [If you're using icd10, change to icd10_to_text_list_concat.txt.]

--start_date="01/01/2021 00:00" [Change this line and the next to the desired start and end dates.]

--end_date="12/31/2021 23:59"

--chunksize=100000

--correcting_misspell="./dicts/correct_common_mistakes_list.txt" [Used for spell-checking.]

--remove_word_list="./dicts/remove_word_list.txt" [Used for stopword removal.]

--output_file_name="../data/synthetic_data/processed_2021.csv" [Four new output files are created by the preprocessing script. This one contains the preprocessed data.]

--output_all_search_groups="../data/synthetic_data/sg_full.csv" [Used for the spatial scanning step.]

--output_sg_dict="../data/synthetic_data/sg_dict_full.pickle" [Used for the spatial scanning step.]

--output_word_index_dict_file="../data/synthetic_data/word_dict_2021.txt" [This is the vocabulary file, containing all distinct words in the background data.]

--word_index_load_file=""

Preprocessing the foreground data

> python preprocessing.py

--functionality=clean_data

--input_file=../data/synthetic_data/synthetic_data_Jan_2022.csv [Point to your foreground data file.]

--sep=","

--attributes_in_sequence="date time hospcode agegroup cc"

--search_group_attributes="hospcode agegroup"

--search_group_file_names="../data/synthetic_data/Settings/single_hospcode_searchgroups.csv ../data/synthetic_data/Settings/age_search_groups_all.csv"

--icd_map_file="./dicts/icd9_to_text_list.txt"

--search_group_attributes="hospcode agegroup“

--start_date="01/01/2022 00:00“ [Desired start and end dates for the foreground data.]

--end_date="01/31/2022 23:59"

--chunksize=100000

--correcting_misspell="./dicts/correct_common_mistakes_list.txt“

--remove_word_list="./dicts/remove_word_list.txt“

--output_file_name="../data/synthetic_data/processed_Jan_2022.csv“ [Name the output file for the preprocessed foreground data.]

--output_all_search_groups=""“ [Don’t need to create this file a second time.]

--output_sg_dict=""“ [Don’t need to create this file a second time.]

--output_word_index_dict_file="../data/synthetic_data/word_dict_full.txt“ [This is the output vocabulary file, which will contain all distinct words in the background and foreground data.]

--word_index_load_file="../data/synthetic_data/word_dict_2021.txt“ [Load in the vocabulary from the background data.]

Learning static topics

> cd ../topic_modeling

> python train_background.py

--data_file="../data/synthetic_data/processed_2021.csv“ [Input file: preprocessed background data.]

--dict_file="../data/synthetic_data/word_dict_full.txt“ [Input file: vocabulary.]

--num_static_topics=25 [Change if you want a smaller, or larger, number of static topics.]

--static_iters=3000 [Run time is proportional to number of iterations; at least 1000 recommended.]

--verbose=100 [Print a line every 100 iterations to update the user on progress.]

--checkpoint="../data/synthetic_data/checkpoint/static_2021_25topics.npz” [Output file: will contain the learned static topics.]

Note: this run could take hours, depending on the size of the background dataset! (But just a couple of minutes for the synthetic data.) Once you've run it once, the static topics can be reused for daily runs, and re-learned monthly or quarterly to account for gradual changes in ED case distribution.

Viewing the learned topics (optional):

> python view_topics.py

--checkpoint="../data/synthetic_data/checkpoint/static_2021_25topics.npz“ [Input file: learned static topics]

--word_dict="../data/synthetic_data/word_dict_full.txt“ [Input file: vocabulary.]

--display_words=10 [Show top 10 words for each topic.]

[Left column is the word's index in the vocabulary file.]

[Right column is the word's probability in that topic distribution.]

	word	weight
2519	sp	0.117470
232	back	0.066265
1457	knee	0.060241
972	fall	0.057229
2429	shoulder	0.047591
2679	swelling	0.040964
1749	neck	0.039759
70	ago	0.027109
2923	upper	0.025904
1554	lower	0.022289

Emerging topics + spatial scanning

> python semantic_scan.py

--full_scan_file="../data/synthetic_data/processed_Jan_2022.csv“ [Input file: preprocessed foreground data.]

--dict_file="../data/synthetic_data/word_dict_full.txt“ [Input file: vocabulary.]

--sg_dict="../data/synthetic_data/sg_dict_full.pickle“ [Input file, created by preprocessing script.]

--sg_list="../data/synthetic_data/sg_full.csv“ [Input file, created by preprocessing script.]

--static_checkpoint="../data/synthetic_data/checkpoint/static_2021_25topics.npz“ [Input file: static topics.]

--import_monitored=""“ [Optional input file, containing user-specified topics “to monitor” or “to ignore”.]

--num_foreground_topics=25 [Change if you want a smaller, or larger, number of emerging topics.]

--static_iters=1000

--contrastive_iters=1000

--verbose=False

--step_size=1

--window_size=3 [Default is to step through each hour of each day, using a 3-hour moving window. For each such window, MUSES it will first learn emerging topics using the most recent 3 hours, then spatial scan over windows between 1 and 3 hours in duration.]

--baseline_size=21

--score_threshold=4.0 [Can increase score threshold to reduce runtime and return fewer clusters.]

--start_date="2022/01/29“ [Choose start and end dates for the run.]

--end_date="2022/01/31"

--topic_weight=False

--cluster_dir="../data/synthetic_data/clusters“ [Specify directory where output files will be created.]

--concatenate_agegroup=True [set to True if age groups are adjacent numeric age ranges, False otherwise.]

Emerging topics + spatial scanning

- Running MUSES creates eight result files (four for “novel” clusters, four for “monitored”) in the output cluster directory.
  - “Monitored” files will be empty, unless you provide monitored topics.
  - These files are best explored with our visualization tool, but novel_raw.csv and monitored_raw.csv can also be viewed in Excel.
- Clusters are indexed in the leftmost column (0, 1, …). All rows with that index are cases belonging to that cluster.
- Each row has: cluster index, cluster score, cluster topic #, chief complaint, ICD code(s), VISITID, time, date, sex, agegroup, hospcode, cluster topic (distribution over words).

index	score	topic	cc	icd	VISITID	time	date	sex	agegroup	hospcode	word_dist
0	9.40944	14	ENCOUNTER FOR EXAMINATION AND OBSERVATION FOLLOWING OTHER ACCIDENT	V87.7XXA		3:12	3/17/2021	F	41-45	HOSP30	encounter_0.16_ examination_0.15_ **...**

Goals of today's training session

1. Understand the purpose and value of pre-syndromic surveillance, as distinct from syndromic surveillance.
2. Gain a high-level understanding of the methodology used for MUSES.
3. Download and install MUSES, and prepare data for analysis.
4. Run MUSES on sample (synthetic) data or on one's own collected data.
5. Visualize detected clusters using our graphical interface.
6. Provide feedback on newly discovered syndromes for MUSES "to monitor" or "to ignore" in future runs.
7. Interpret and use detection results.
8. Understand practical considerations and limitations for using MUSES in day-to-day practice.

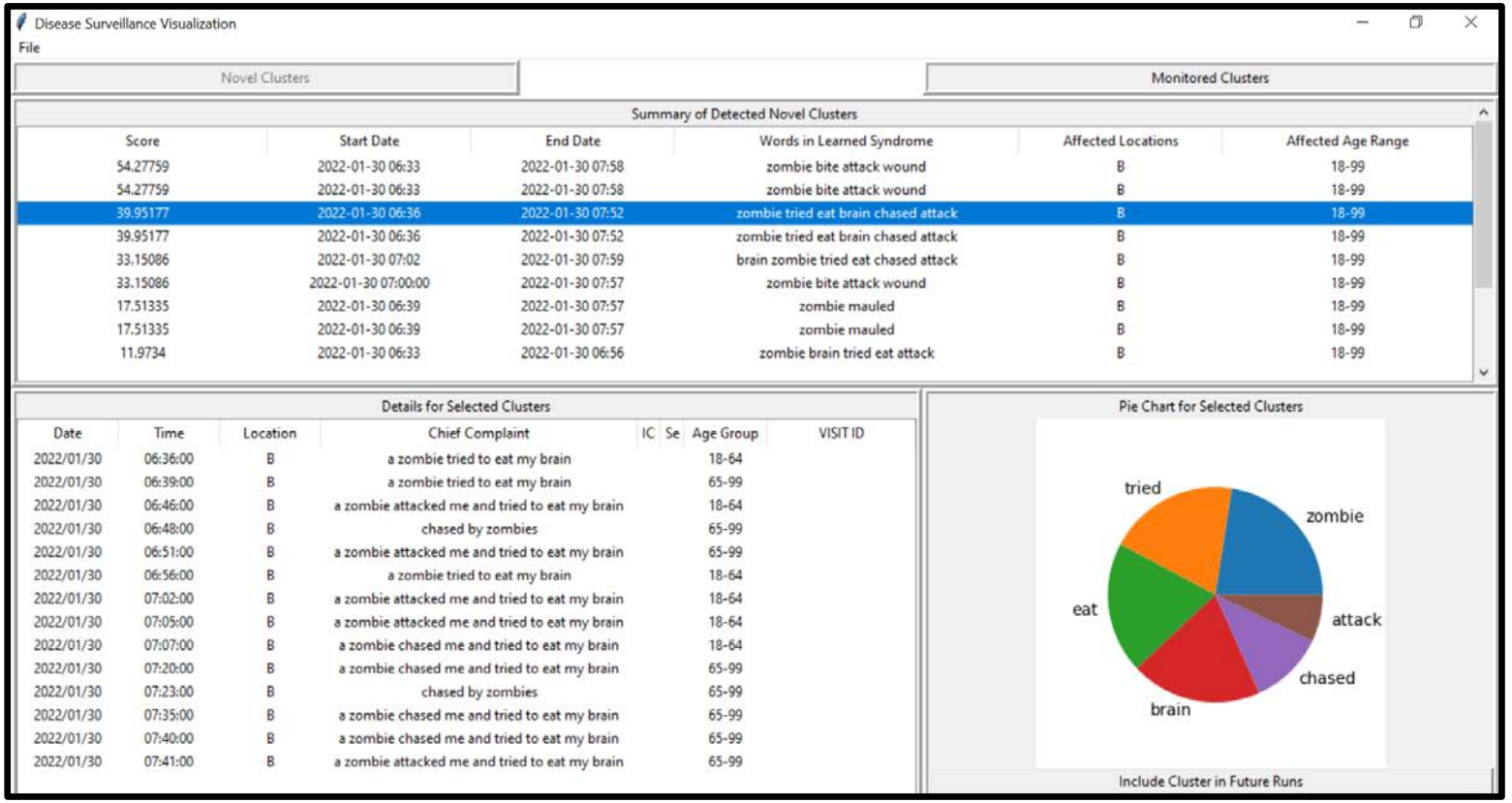
Visualizing detected clusters

> cd ../visualization

> python visualize.py

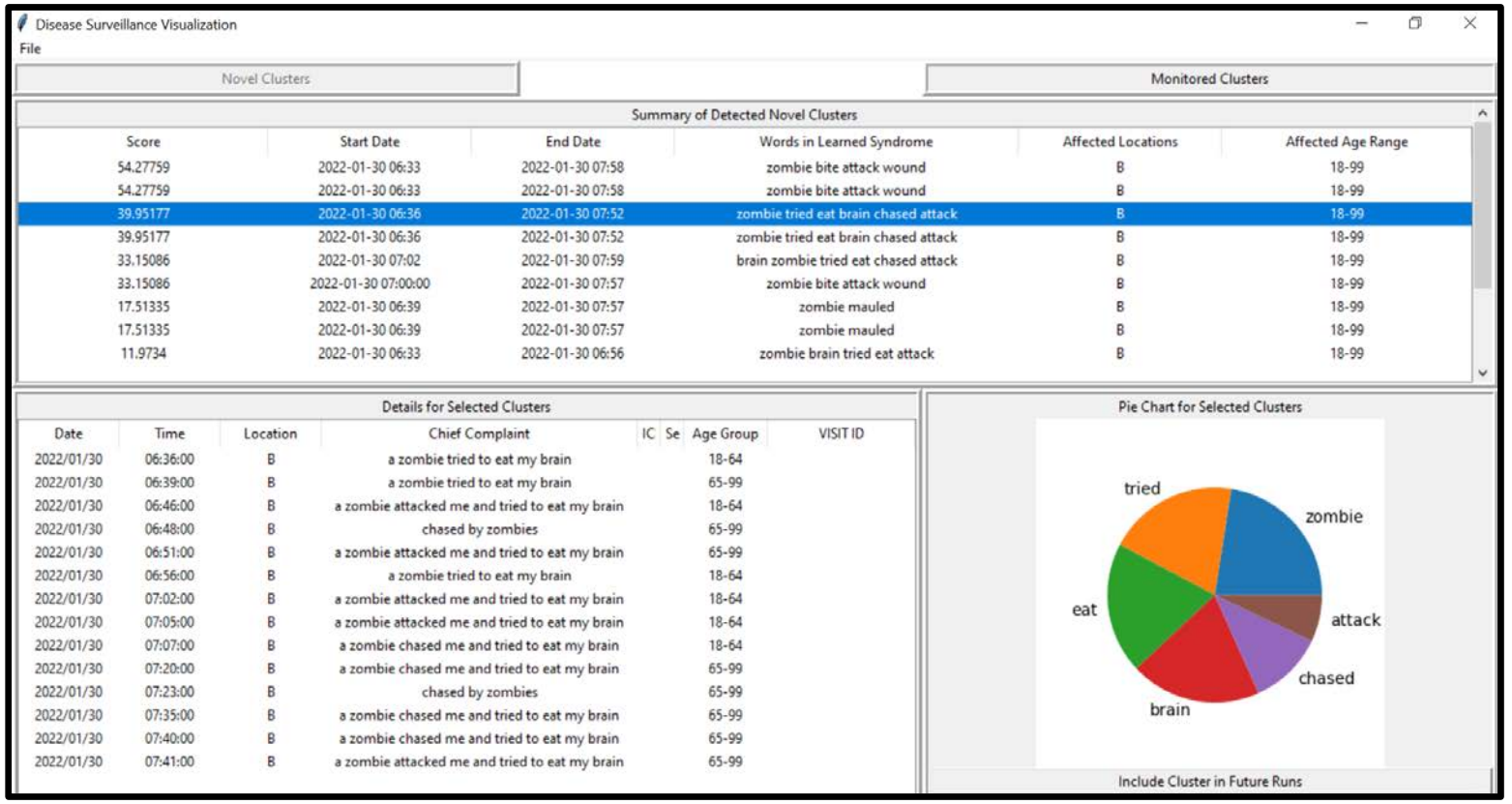
--results_folder="../data/synthetic_data/clusters“ [Input file directory.]

--monitored_topic_file="../data/synthetic_data/monitored_topics.csv“ [Output file: topics “to monitor” or “to ignore”. Will create this file if not present, or append to it if present.]



Visualizing detected clusters

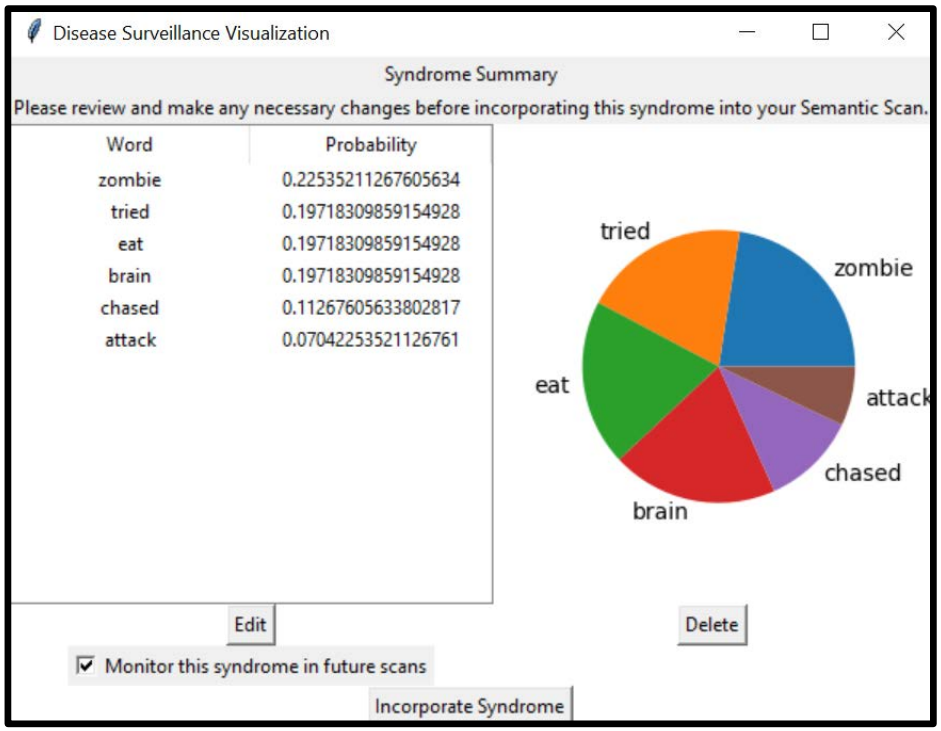
- Two tabs: one for novel clusters and one (currently empty) for monitored clusters.
- Top panel: list of clusters, with score, start and end dates, topic, hospital codes, and age range for each.
- Click on any column to sort (ascending or descending) → I recommend sorting by score (descending).
- Click on any row of the top panel to show its corresponding case list in the bottom panel and its topic in the (lower-right) pie chart.
- Bottom panel: date, time, hospital code, chief complaint, ICD, sex, age group, and visit ID for each case.



Incorporating user feedback

When looking at a detected “novel” cluster in the visualization tool, you can add that topic to the monitored topic file, as a new topic “to monitor” or “to ignore”. To do so, click on “Include Cluster in Future Runs” (in the lower right corner).

This does not affect visualization of the current results, but will be incorporated into the topic model for future runs, giving higher power to detect clusters of the monitored topics and fewer detected clusters of the ignored topics.



Clicking on “Include Cluster in Future Runs” pops up a dialog box like this one. You should then:

1) Edit the set of words and probabilities, if desired. (Probs renormalize automatically.)
2) Decide whether to monitor or ignore the topic, and either check or uncheck “Monitor this syndrome in future scans”.
3) Click “Incorporate Syndrome”.

Incorporating user feedback

When looking at a detected “novel” cluster in the visualization tool, you can add that topic to the monitored topic file, as a new topic “to monitor” or “to ignore”. To do so, click on “Include Cluster in Future Runs” (in the lower right corner).

This does not affect visualization of the current results, but will be incorporated into the topic model for future runs, giving higher power to detect clusters of the monitored topics and fewer detected clusters of the ignored topics.

More about the monitored topic file:

- Each line of the file is of the form “1_word_prob_word_prob_...” for a topic “to monitor” or “0_word_prob_word_prob_...” for a topic “to ignore”.
- These are treated as additional static topics in the emerging topic modeling step: “novel” topics will be distinguished from these as well as the original static topics.
- In the spatial scan step, we scan over the “emerging topics” and the topics “to monitor”, but not the original static topics nor the topics “to ignore”.

Goals of today's training session

1. Understand the purpose and value of pre-syndromic surveillance, as distinct from syndromic surveillance.
2. Gain a high-level understanding of the methodology used for MUSES.
3. Download and install MUSES, and prepare data for analysis.
4. Run MUSES on sample (synthetic) data or on one's own collected data.
5. Visualize detected clusters using our graphical interface.
6. Provide feedback on newly discovered syndromes for MUSES "to monitor" or "to ignore" in future runs.
7. Interpret and use detection results.
8. Understand practical considerations and limitations for using MUSES in day-to-day practice.

Some limitations of MUSES*

1) Lags in data collection, preprocessing, analysis, or communication of results may affect timeliness, and thus effective use of MUSES depends on a well-developed data infrastructure and the availability of public health practitioners to respond rapidly to detected clusters.
2) False positive clusters could result from repeated typographical errors by a triage nurse, hospital EHR changes and upgrades, or the use of new or unusual terminology to describe cases within a given hospital.
3) False negatives could result from sampling bias and recording bias, since not everyone who has a particular symptom presents at the ED; patients may describe symptoms differently, or they may be recorded differently.

*See the discussion section of our Science Advances paper for more details.

Practical considerations*

1) Day-to-day use: a script could be set up to run MUSES each night on the previous 24 hours of data, so that results are ready for perusal by public health each morning.
2) Static topics could be re-learned from the past year of data at regular intervals (e.g., quarterly). (Don't re-learn daily!)
3) "Monitored" and "ignored" clusters will improve detection, but having too many will lead to longer run times.
   - Don't add everything, just frequently occurring irrelevant clusters (e.g., car accidents) and those which are particularly high-priority.
4) Again, pre-syndromic surveillance should be used as a **complement** to existing public health practices (notifiable disease reporting, syndromic surveillance, etc.) since it focuses on detecting clusters that other systems cannot.

*See the discussion section of our Science Advances paper for more details.

Thanks for listening!

More details and MUSES open-source software on our project page:
https://wp.nyu.edu/ml4good/pre-syndromic-surveillance

Or e-mail me at:
daniel.neill@nyu.edu