A Two-stage Labeling Framework for Effective Text Classification

Duke University, Statistical Science

Two-stage Labeling for Text Classification

Motivation

Goals

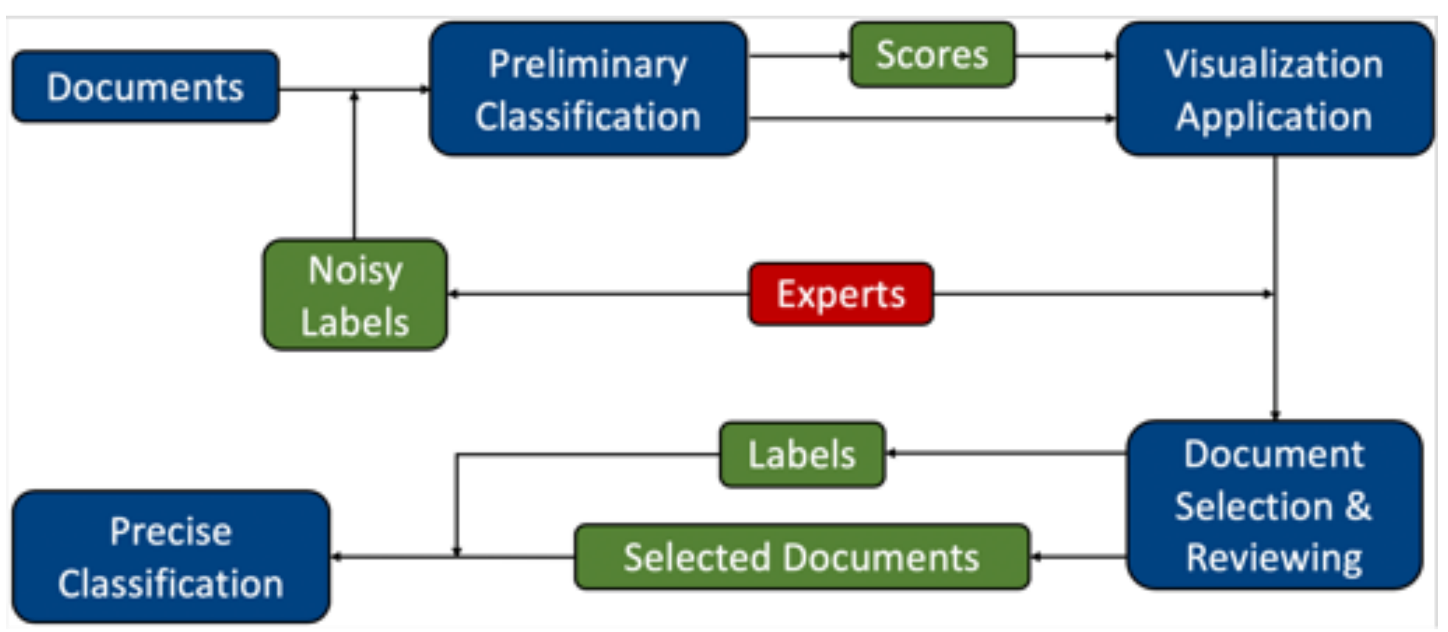
- To identify documents related to certain topics via text classification
- To obtain labels from experts for training such classifiers

Problems

- Large dataset: not feasible to label all data
- Noisy dataset: randomly sampling would not likely to collect enough relevant documents
- Imbalanced classes from the obtained labels

Architecture

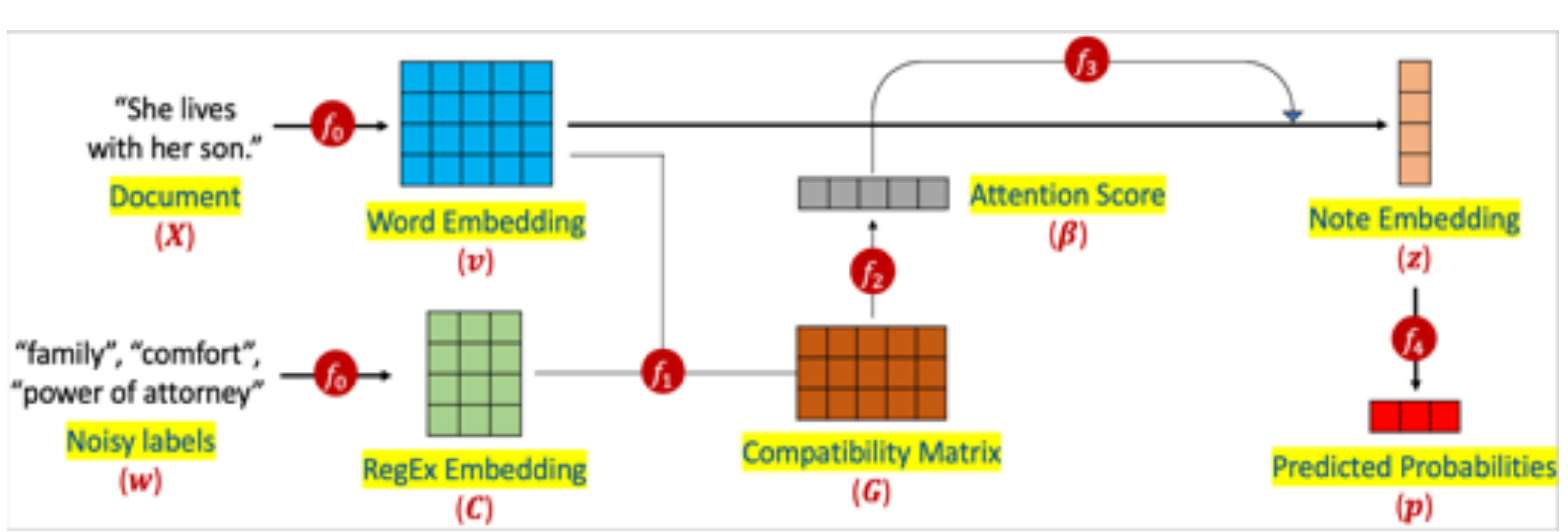
- A two-stage classification and labeling framework



Methods

Preliminary Classification

- Model: Label-Embedding Attentive Model (LEAM), a convolutional neural network (CNN) framework
- Noisy label: topic-related regular expressions (RegEx) proposed by experts **before they review the documents**



- Training objective: $\min \frac{1}{N}\sum_{i=1}^{N} \text{cross}-\text{entropy}\ (w_i, p_i)$
- Output: note embedding vectors, attention scores, and probabilities

- Functions in LEAM:

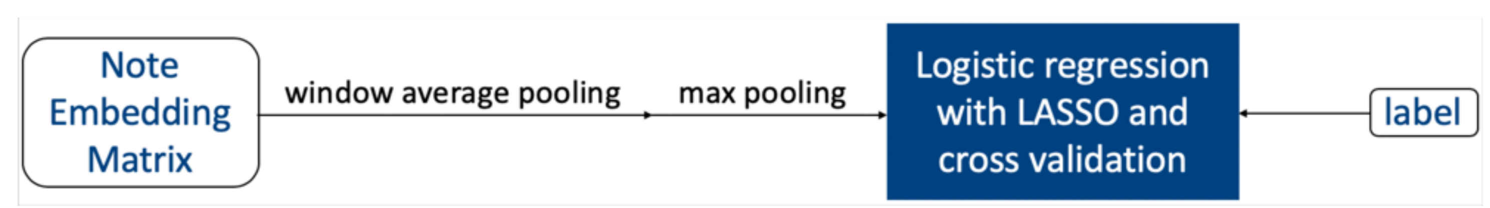
Function Name	Description
word embedding (f_0)	To convert each word in the corpus into a vector (of high dimension), thus each document into a matrix
compatibility measure (f_1)	To measure the compatibility of label-word pairs via cosine similarity: $g_{kl} = < c_k, v_l > / \lVert c_k \rVert \lVert v_l \rVert$
feature extraction (f_2)	To apply a convolution layer with non-linearity and bias, a max pooling layer and a softmax function
attentive averaging (f_3)	To average word embedding with attention scores
probability calculation (f_4)	To apply a linear layer and a sigmoid function

Document Selection and Labeling

- To apply *t-SNE* to carry out dimensionality reduction and data visualization incorporated with LEAM output
- To develop web application embedded with the visualization for experts to select candidate documents and label them

Precise Classification

- Input: selected notes' embedding vectors, labels provided by experts
- Model:



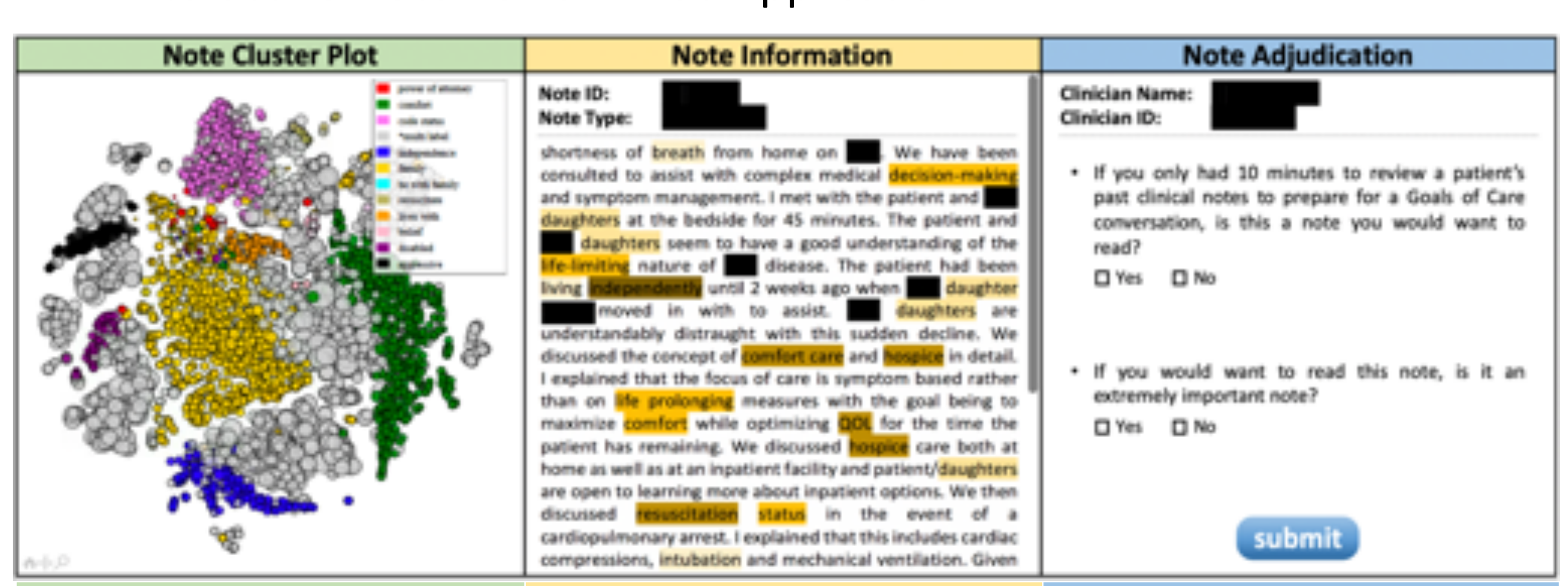
Application

Background

- Topic: Goals of Care (GoC) conversations that result in better clinical outcomes, improved quality of life and more rational utilization
- Data: over 5.3 million clinical notes from DUKE EHR for 97,402 beneficiaries in the Medicare Shared Savings Program (MSSP)

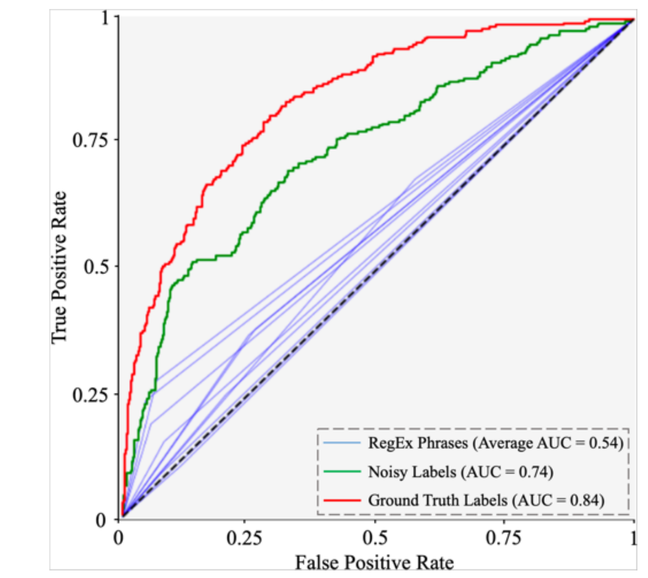
Results

- Interactive Note Review Web Application



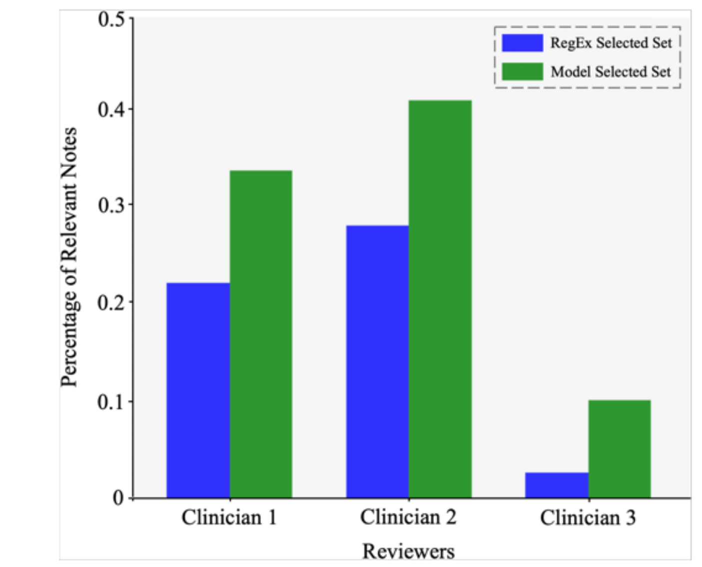
Note Cluster Plot	Note Information	Note Adjudication
• Each point represents a note and different colors represent different RegEx phrases. Users can click on points to check the corresponding notes • The size of points is proportional to the summed probabilities predicted by LEAM, which represents how relevant a note is to GoC.	• The note content pertaining to the point clicked on by users will pop up in the middle panel. • The text is highlighted according to the attention scores assigned by LEAM, and in general, the darker the highlighting color, the more important/informative the word	• After reviewing the note text, users can label the note by answering the questions in this panel for adjudication. • These labels will instantly be saved to our database for later modeling.

- Classification performance



- Blue ROC curves correspond to detecting the existence of RegEx phrases in notes.
- The green ROC curve corresponds to LEAM trained with RegEx as labels.
- The red ROC curve corresponds to the classification model trained with clinicians' adjudication (the ground truth) as labels.

- Labeling efficiency



- Labeler: clinicians with different labeling preferences from Duke Palliative Care
- Percentages of relevant notes in the RegEx selected set are22.4%, 28.4%, and 2.67% for clinician 1, 2, and 3, respectively.
- Percentages of relevant notes in the model selected set are 34.1%, 41.4%, and 10.2% for clinician1, 2, and 3, respectively.

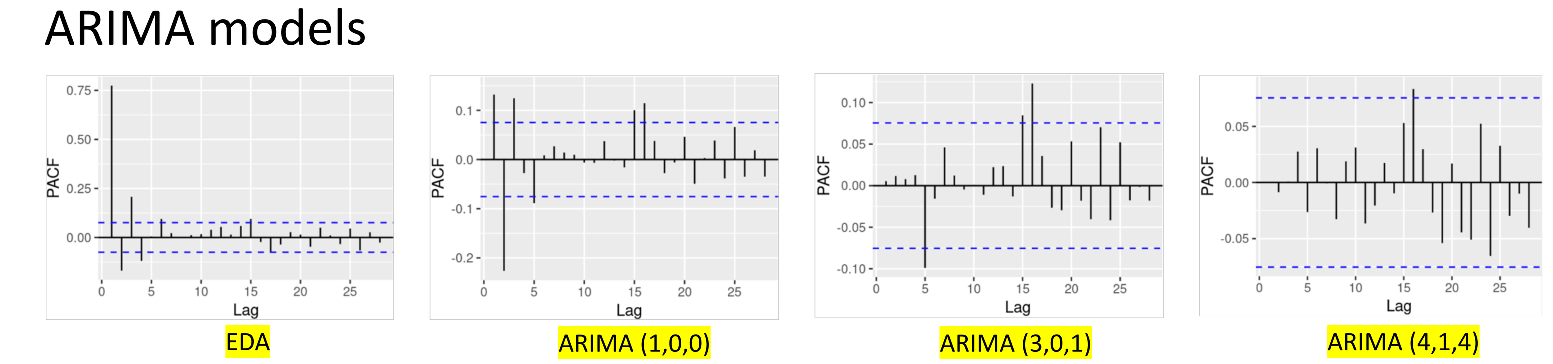
Temporal Modeling for Air Quality

Introduction

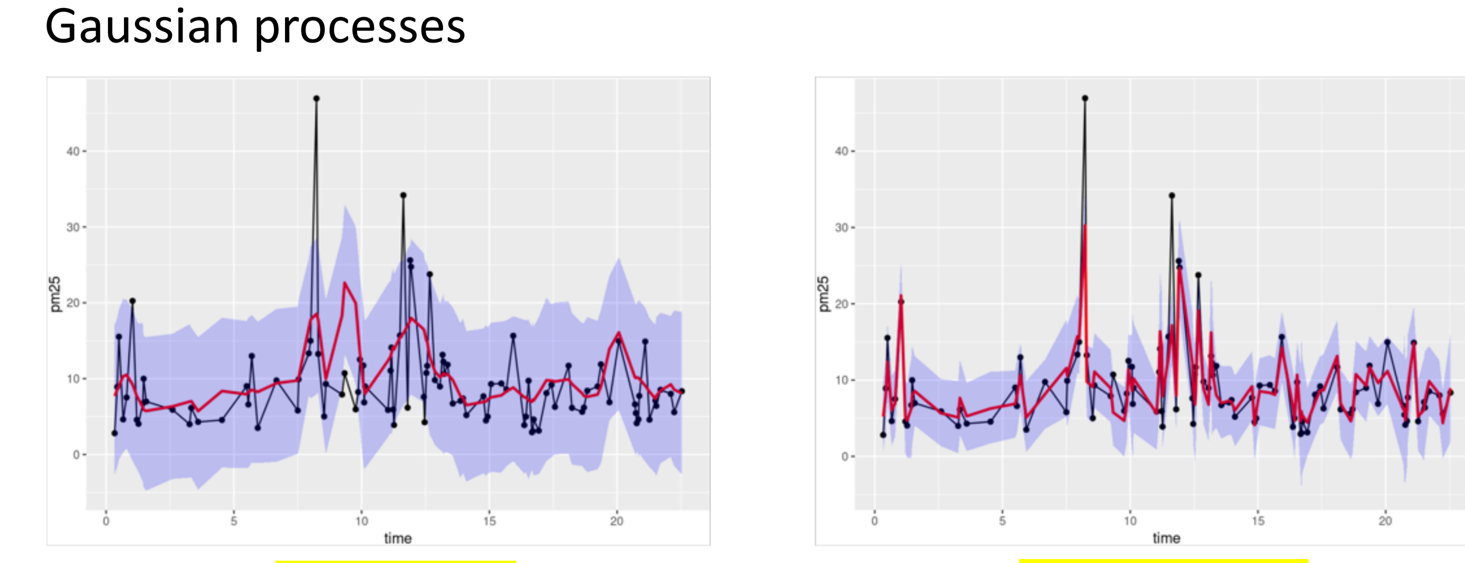
- Data: Daily PM 2.5 concentration for each county in California from January 1st 2017 to November 30th 2018
- Goal: to build time series models to explore how air quality changes over the time

Methods and Results

- ARIMA models



- Gaussian processes



- Performance

Model	ARIMA (1,0,0)	ARIMA (3,0,1)	ARIMA (1,0,0)	GP	Bayesian GP
RMSE	11.14	10.69	10.61	5.754	3.213

Conclusion

- In general, continuous models achieved better performance than discrete ones, and Bayesian Gaussian process performed the best.

References:
Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, Lawrence Carin. 2018. Joint Embedding of Words and Labels for Text Classification. ACL 2018.
Laurens van der Maaten, Geoffrey Hinton. 2008. Visualizing Data using t-SNE. Journal of Machine Learning Research.