Supplementary Material for Characterizing the Value of Information in Medical Notes

Chao-Chun Hsu[1], **Shantanu Karnwal**[2],
Sendhil Mullainathan[3], **Ziad Obermeyer**[4], **Chenhao Tan**[2]
[1] University of Chicago,[2] University of Colorado Boulder
[3] Chicago Booth School of Business, [4] University of California, Berkeley
chaochunh@uchicago.edu
{shantanu.karnwal,chenhao.tan}@colorado.edu
sendhil@chicagobooth.edu, zobermeyer@berkeley.edu

A Experiment Setup

Filtering invalid data. We follow the data preprocessing procedure described in Harutyunyan et al. (2019). Specifically, we first collect the event records of a patient by grouping by patient id. Then, we split events of different admissions based on admission id. For all experiments, we eliminate organ donors, i.e., the patients who died already but were readmitted to donate their organs, and the patients who do not have chart event data. Additionally, events missing admission ids are eliminated. Note that not all admissions have medical notes. We only select admissions with notes for our experiments. After filtering out invalid data, we obtain 34,847/37807 patients and 44,055/48,262 admissions for 48 hours/retrospective mortality prediction.

Model Training We trained DAN and GRU-D on a single GPU (Nvidia Titan RTX) with PyTorch (Paszke et al., 2019) and we use AdamW (Reddi et al., 2019) as an optimizer. Best models are selected based on PR-AUC scores on validation set. For model trained with only notes, we set learning rate to 3e-4; for model trained with both types of features or only structured variables, we choose 1e-4 as our learning rate. We train model up to 20 epochs. We choose the epochs and learning rate that lead to the best performance on the validation set.

B Model Details

B.1 Deep Averaging Networks (DAN)

A DAN is similar to a bag-of-word classification model, but the bag-of-words features are replaced by word embeddings. First, we concatenate all notes in an admission as input text and transform input text into a list of token ids $W = \{w_1, w_2, ..., w_P\}$, for which P denotes token length. Then, we obtain word embeddings of each token from a word embedding matrix $M^{V \times D}$ where D is dimension of word embedding. After that, we calculate the mean of all word embeddings as the representation $x_{mean} \in R^D$ of the input text. Finally, we concatenate x_{mean} and e_i and feed it into final dense layer to obtain the prediction probability output by a softmax function.

B.2 DAN with Attention

Instead of computing averaging word embedding of all notes as a whole, we first generate average word embedding of each note x_t in an admission separately. Then, we compute attention weights and final text representation x as follows:

$$s_t = Wx_t + b \tag{1}$$

$$\alpha_t = \frac{exp(s_t)}{\sum_{t=1}^{T} exp(s_t)} \tag{2}$$

$$x = \sum_{t=1}^{T} \alpha_t x_t \tag{3}$$

where T denotes number of notes in the admission and $W \in R^d$ is a trainable vector. x is fed into final layer for prediction.

B.3 GRU-D

In contrast to the system described in Che et al. (2018), we have two types of input features, structured variables and notes, so the input dimension of our model are number of structured variables (767) plus the word embedding dimension (300). Note that statistical functions of different time windows on structured variables is not applicable because the input of GRU-D should be the content of a single event in the admission. Also, we do not impute missing value in note representation, because we cannot pre-compute averaged word embeddings across the training set since word em-

beddings are continually updated during the training process. As shown in Table 2, GRU-D consistently performs worse than logistic regression and DAN.

C Pairwise Comparison with DAN

Results of pairwise comparison with DAN are similar with results with logistic regression (LR). Fig. 4 shows that discharge summaries dominate other types of notes for readmission prediction. In mortality prediction, nursing notes are the most useful notes.

D Similarity Computation

We have tried to compute the tf-idf similarity of a sentence in the last note with all previous sentences instead of notes, and results are similar.

Besides max similarity value function, we also conduct experiments on average similarity with or without length normalization.

$$V_{sim\_max}(s) = \max_{x_k \in X} cossim(s, x_k)$$

$$V_{sim\_avg}(s) = \frac{1}{k-1} \sum_{k=1}^{K-1} cossim(s, x_k)$$

$$V_{sim\_max\_n}(s) = \max_{x_k \in X} cossim(s, x_k) * \sqrt{\text{length}(s)}$$

$$V_{sim\_avg\_n}(s) = \frac{1}{k-1} \sum_{k=1}^{K-1} cossim(s, x_k) * \sqrt{\text{length}(s)}$$

where K denotes number of notes in the admission. The value functions for dissimilar sentences are as follows:

$$V_{dis\_max}(z) = -V_{sim\_max}(z)$$
$$V_{dis\_avg}(z) = -V_{sim\_avg}(z)$$
$$V_{dis\_max\_n}(z) = -V_{sim\_max\_n}(z)$$
$$V_{dis\_avg\_n}(z) = -V_{sim\_avg\_n}(z)$$

While compute similarity, we force selected sentences to have at least five tokens to prevent super short sentences. Results in Fig. 5 and Fig. 6 show no significant difference between max and other similarity methods.

Truncate at a given percentage. To make a fair comparison, we first calculate the total number of tokens from selected sentences which are sorted by scores from a value function. Then, we compute number of tokens given a percentage as $n = n_{total} * \text{percentage}$ and we keep pushing sentences to the output list with descending order of scores until the number of tokens in output list exceed n. Last, we truncate exceeding tokens in last selected sentence to obtain final output sentences.

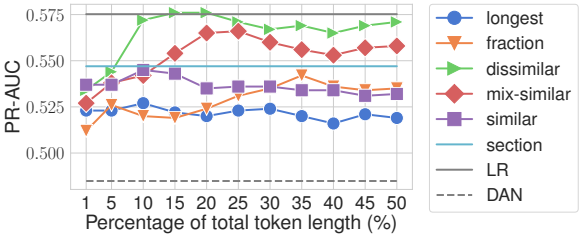


Figure 1: PRAUC scores of value functions for mortality prediction (24 hours) on LR with different **percentages of token length** using both structured information and notes. Since structured variables alone already dominate in this tasks, notes do not add additional predictive values to the results.

Structured information dominates mortality prediction task. As shown in Fig. 1, logistic regression already performs well with structured variables alone in mortality prediction. Adding part of notes does not add much predictive value over using all notes.

E Leveraging Valuable Information : Clinical-BERT

Since ClinicalBERT has input length limitation of 512 sub-word tokens, we fine-tune ClinicalBert with sentences selected by proposed value functions where sentences are truncated at 400 tokens. Note that truncation based on number of tokens instead of percentage of tokens will drastically reduce long records to too little information, and vice versa. As shown in Fig. 2, fine-tuned ClinicalBert based on selected sentences has similar performance with re-trained logistic regression model.

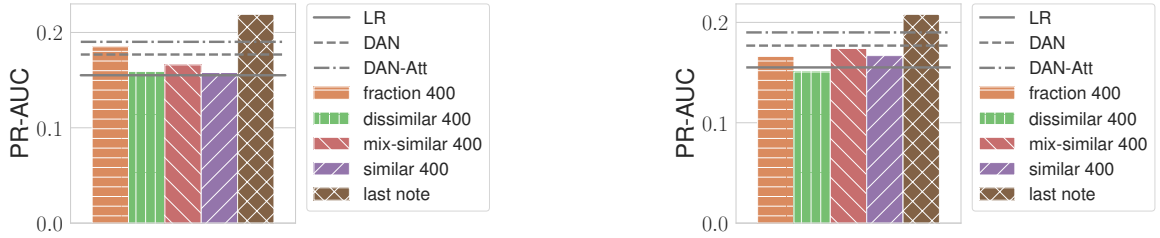
References

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. 2018. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085.

Hrayr Harutyunyan, Hrant Khachatrian, David C. Kale, Greg Ver Steeg, and Aram Galstyan. 2019. Multitask learning and benchmarking with clinical time series data. *Scientific Data*, 6(1):96.

		LR		DAN		GRU-D	
		PR-AUC	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC	ROC-AUC
M:24	S+N	0.575	0.891	0.484	0.850	0.278	0.736
	S	0.567	0.892	0.463	0.837	0.269	0.731
	N	0.288	0.754	0.323	0.767	0.15	0.585
M:48	S+N	0.558	0.903	0.520	0.888	0.254	0.764
	S	0.547	0.902	0.519	0.890	0.268	0.774
	N	0.292	0.794	0.341	0.810	0.105	0.536
M:retro	S+N	0.927	0.983	0.902	0.978	0.684	0.912
	S	0.921	0.982	0.892	0.976	0.714	0.923
	N	0.745	0.935	0.816	0.953	0.319	0.752
Readmission	S+N	0.156	0.714	0.189	0.741	0.132	0.619
	S	0.150	0.699	0.144	0.682	0.112	0.606
	N	0.155	0.730	0.176	0.744	0.085	0.552

Table 1: Results of PR-AUC/ROC-AUC scores on LR/DAN/GRU-D models in readmission prediction and mortality prediction ("M") tasks. S denotes structured variables and N is notes.



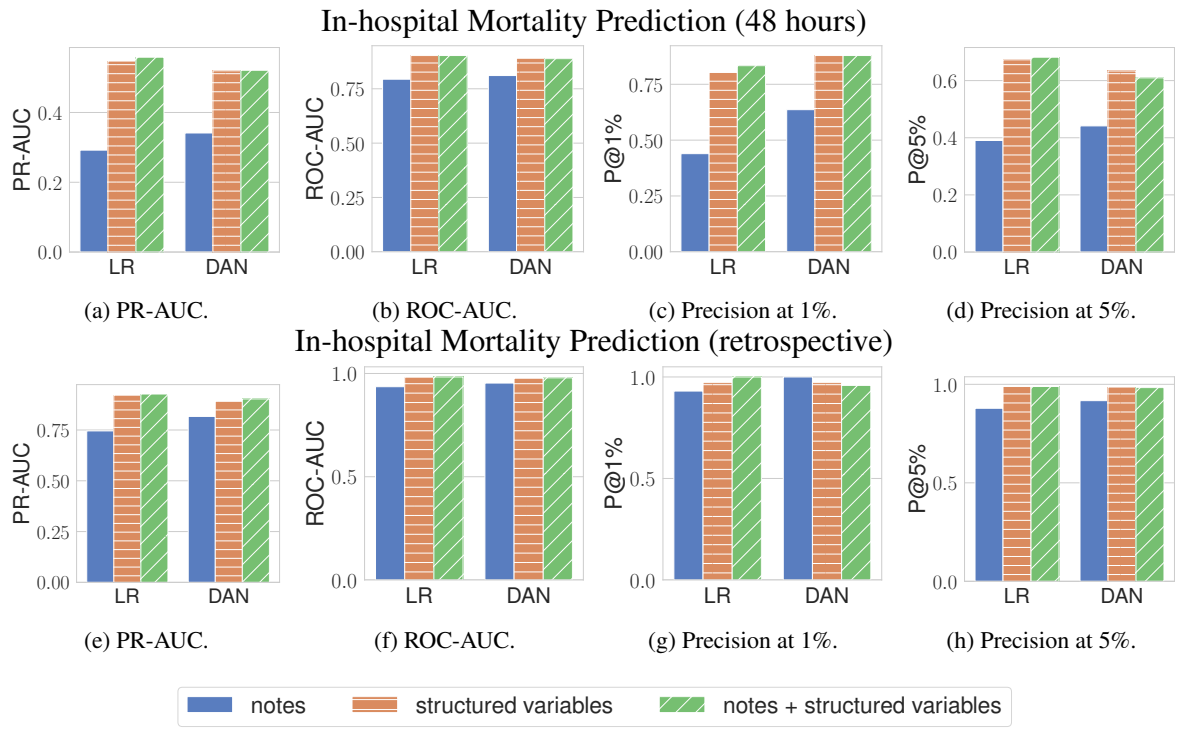
(a) Logistic regression.

(b) ClinicalBERT

Figure 2: Performance of trained models with selected valuable information (400 tokens) on logistic regression and ClinicalBert. ClinicalBERT does not necessarily provide better performance than logistic regression. The mix-similar method is the best among three similarity methods. It is different from re-training models based on **percentage** of tokens where dissimilar sentences provide highest predictive values.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. pages 8024–8035.

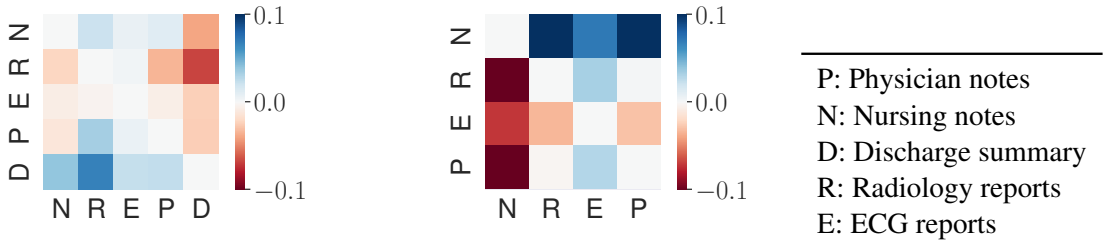
Sashank J Reddi, Satyen Kale, and Sanjiv Kumar. 2019. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*.



(a) PR-AUC. (b) ROC-AUC. (c) Precision at 1%. (d) Precision at 5%.

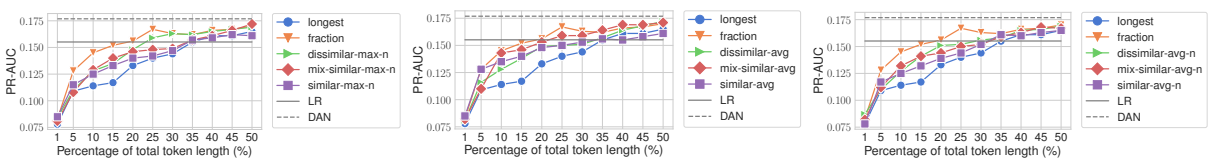
(e) PR-AUC. (f) ROC-AUC. (g) Precision at 1%. (h) Precision at 5%.

Figure 3: Results of PR-AUC/ROC-AUC/Precision at 1%/Precision at 5% scores on LR/DAN models in mortality prediction (48 hours and retrospective) tasks. Notes are marginally valuable in mortality prediction.



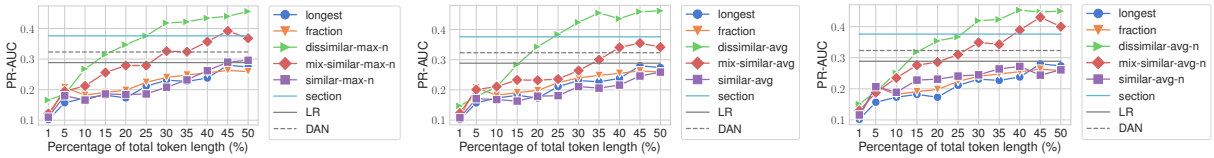
(a) Readmission prediction (b) Mortality prediction (24 hrs)

Figure 4: Pairwise comparisons between different types of notes on DAN (each grid shows $\text{PR-AUC}(f_{\text{all}}(s_{t_{\text{row}}}), y) - \text{PR-AUC}(f_{\text{all}}(s_{t_{\text{column}}}), y)$). To account for the differences in length, we sub-sample two types of notes under comparison to be the same length and report the average values of 10 samples. Discharge summaries dominate all other types of notes in readmission prediction, while nursing notes are most useful for mortality prediction.



(a) Max norm. (b) AVG. (c) AVG norm.

Figure 5: PRAUC scores of other similarity value functions for readmission prediction on LR with different percentages of tokens.



(a) Max norm. (b) AVG. (c) AVG norm.

Figure 6: PRAUC scores of other similarity value functions for mortality prediction with 24 hours period on LR with different percentages of tokens.