Evaluating the utility of clinical text in learned patient representations

Conor K. Corbin[1], Gautam B. Machiraju[1]

[1]Department of Biomedical Data Science, Stanford University

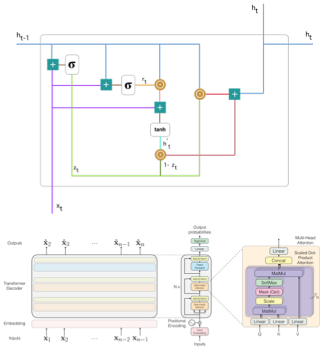
Problem

Problem: Effective use of Electronic Health Record (EHR) data promises to reveal clinical insights and improve patient care. EHR data, however, is heterogeneous, noisy, largely unstructured, and temporally asynchronous. Eighty percent of EHR data comes in the form of unstructured clinical text. Developing predictive models that perform well across health intuitions remains a challenge. While patient data in and of itself is large (Stanford's Clinical Data Warehouse houses over 3M de-identified patient timelines), often the number of patients with a clinical outcome worth predicting (development of rare disease, mortality, clinical trial eligibility)is comparatively small. Inspired by recent NLP advancements leveraging word embeddings and language models for transfer learning, we develop three neural methods to pre-train contextualized patient representations. We then leverage these representations in two downstream clinical prediction tasks (1) inpatient mortality and (2) 30-day inpatient readmission. Further, we compare the performance of learned representations that are trained on data-streams comprised of structured data only, unstructured data only, and a combination of the two

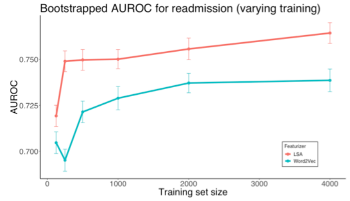
Data & Task

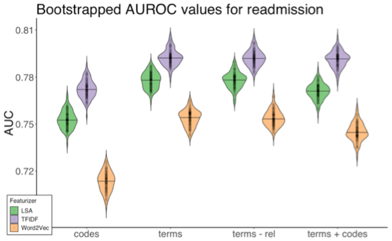
- Data/Task: We utilize Stanford's Clinical Data Warehouse (STRIDE), which houses more than 3M de-identified patient medical timelines. These timelines include both structured codes data and 70M free text clinical notes. We have access to this data through Dr. Nigam Shah in the Department of Biomedical Data Science. For comparison, have developed learned patient representations trained on the following four sets of clinical data streams and multiple (baseline and neural) featurization schemes.
- Codes / Structured EHR data comes in the form of temporally spaced sequence of tokens/coded data(diagnosis codes, procedure codes, medications, lab orders) that together form a "patient timeline". In our dataset, the temporal resolution of our timeline is a patient day. The number of codes assigned to a patient vary from day to day, and the number of recorded days varies per patient. Terms In this data stream, we leverage medical terms and concepts extracted from free text clinical notes. Medical terms are defined according to the Unified Medical Language System (UMLS)ontology library, and extracted via simple string parsing. These term mentions are grouped at patientday resolution.Codes + TermsHere we merge the structured codes data stream with medical term mentions(aggregated at day-level). Like before, tokens within a patient day are shuffled.Terms - RelationsIn this data-stream we hope to elucidate the importance of medical term contextwithin the clinical notes from which they originate. Relation extraction tools such asNegExandConTextare used to detect negated terms (ex. "patient denies cough") and terms that refer to familyhistory (ex. "patient's mother has history of breast cancer"). Like in the third data-stream, termmentions are combined with structured data. Here relations are ablated, so that we can quantify theirpredictive power in downstream tasks

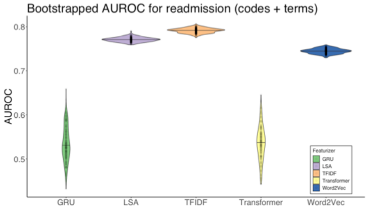
Approach



Results


Bootstrapped AUROC for readmission (varying training)


Bootstrapped AUROC values for readmission


Bootstrapped AUROC for readmission (codes + terms)

Math Model

$$z_t = \sigma(W_z \mathbf{x}_t + U_z h_{t-1} + b_z) \qquad \text{[update gate]} \tag{1}$$

$$r_t = \sigma(W_r \mathbf{x}_t + r_z h_{t-1} + b_r) \qquad \text{[reset gate]} \tag{2}$$

$$h'_t = \tanh(W_h \mathbf{x}_t + (r_t \odot U_h h_{t-1}) + b_h) \qquad \text{[current memory]} \tag{3}$$

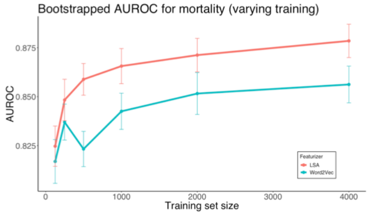
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot h'_t \qquad \text{[final memory]} \tag{4}$$

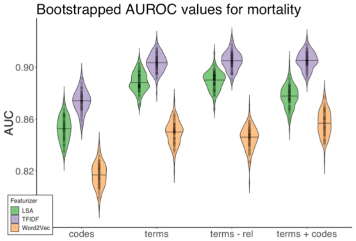
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK}{\sqrt{d_k}}\right) V \qquad \text{[Scaled dot-product attention]} \tag{5}$$

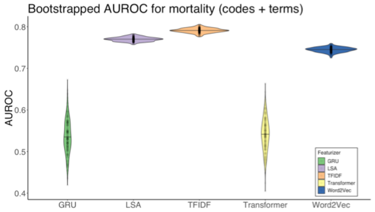
$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W^O \qquad \text{[Multi-head attention]} \tag{6}$$

$$\text{SubOut} = \text{LayerNorm}(\mathbf{x}_t + \text{Sublayer}(\mathbf{x}_t)) \qquad \text{[Sublayer output]} \tag{7}$$

$$\text{FFN}(\mathbf{x}_t) = \max(0, \mathbf{x}_t W_1 + b_1) W_2 + b_2 \qquad \text{[Feed forward layer]} \tag{8}$$


Bootstrapped AUROC for mortality (varying training)


Bootstrapped AUROC values for mortality


Bootstrapped AUROC for mortality (codes + terms)

Analysis & Conclusions

- Survey of Embedding Methods Our results suggest that simple embedding methods yield the highest predictive performance on downstream tasks. TFIDF outperforms both LSA and Word2vec. Our language models perform essentially no better than random. Given enough training data, it makes sense that of TFIDF would perform so well. Although LSA and Word2vec produce more compact patient representations than that TFIDF, they appear to lose predictive signal in the process. Both TFIDF and LSA perform better than Word2vec. This may be the case for multiple reasons. First, our Word2vec formulation relies on the assumption that code and medical term representations can be elucidated from their neighboring codes and terms. By our own admission, the immediate neighborhood of codes and terms are stochastic (we shuffle all codes and terms in a patient day because we lack the temporal resolution to properly order them). In the majority of cases this shuffling shouldn't have a large effect because the average number of codes in a patient day is small (roughly 15) and we use a context window of size 10. However some patient days carry a large number of codes (upwards of 1000). In these cases, our context window is more or less random. Secondly, we may be losing the majority of our signal in our final aggregation method. We construct dense vectors for each code and medical term using Word2vec, but to create patient representations we take the max element wise across all codes and terms in a patient timeline.

Future Aims

One avenue we could explore is to make the transformer's positional encodings dependent on patient visits dates, as opposed to recurrent time steps of t − 1, t, t + 1, etc. Tangentially, this opens the doors to other forms of hidden state aggregation functions and pooling, such as taking the weighted average of later hidden states. One could imagine that most recent hidden state is the most predictive. Furthermore, on a given visit, further aggregation functions (e.g. mean) could be explored instead of maximum. Expanding on aggregation functions, as a third baseline featurizer, we are also interested in performing simple aggregation methods (e.g. min, max, mean) of our code and term embeddings to create a fixed length vector patient representation [5].

References

[1] Miotto et al. Deep patient: an unsupervised representation to predict the future of patients from the electronic health records. Nature, 2016.

Acknowledgements

Department of Biomedical Data Science AT STANFORD MEDICINE

bmi