Combining LSTM and Latent Topic Modeling for Mortality Prediction

Yohan Jo, Lisa Lee, Shruti Palaskar {YOHANJ,LSLEE,SPALASKA}@CS.CMU.EDU

Abstract

There is a great need for technologies that can predict the mortality of patients in intensive care units with both high accuracy and accountability. We present joint end-to-end neural network architectures that combine long short-term memory (LSTM) and a latent topic model to simultaneously train a classifier for mortality prediction and learn latent topics indicative of mortality from textual clinical notes. For topic interpretability, the topic modeling layer has been carefully designed as a single-layer network with constraints inspired by LDA. Experiments on the MIMIC-III dataset show that our models significantly outperform prior models that are based on LDA topics in mortality prediction. However, we achieve limited success with our method for interpreting topics from the trained models by looking at the neural network weights.

1. Introduction

Many intensive care units (ICUs) suffer from a shortage of nurses and doctors to care for patients in critical conditions. As caregivers inevitably have to prioritize patients based on the severity of their conditions, it is essential to leverage patient data—collected from laboratory tests and clinical notes—to help determine the most efficient ICU resource allocation, e.g., by estimating patient mortality. The problem of mortality prediction involves several challenges. (1) Mortality prediction requires dealing with time series data, where a progressive analysis of the patients' conditions is preferred to cross-sectional analysis. While prior work has conducted time series analysis of clinical notes (Jo & Rosé, 2015; Grnarova et al., 2016) and other measurements (Lipton et al., 2015) for predicting mortality/diagnoses, there is no standard methodology that yields high accuracy in mortality prediction. (2) Algorithmic accountability is also critical, as doctors cannot blindly accept the machine's decisions without knowing the rationales behind them. While neural network techniques have demonstrated promising performance in prediction tasks (Grnarova et al., 2016; Lipton et al., 2015), they often lack interpretability and suffer from data sparsity (e.g., text vocabulary), especially in the case of insufficient data.

We present two-layer joint models for mortality prediction that combine the advantages of long short-term memory (LSTM) and latent topic modeling. The LSTM layer captures long-range dependencies in sequential data, trained for mortality prediction, and this information propagates back to the topic modeling layer to learn topics that are predictive of mortality. We tried three different structures for the topic modeling layer. The Encoder only structure has a single-layer network that encodes a bag-of-words to a latent document vector (topic distribution) using a trainable word-topic weight matrix. The Encoder+Decoder structure reconstructs the input vector from the document vector using a trainable topic-word weight matrix that aims to associate each latent dimension with cohesive words. The Encoder+Transcoder+Decoder structure has an intermediate Transcoder layer that converts a document representation to a sparse vector to mimic the sparsity constraints in LDA.

We evaluate our model on the MIMIC-III dataset. The prediction accuracy of our models is compared to that of LDA-based models. The learned topics are evaluated by examining top words associated with each latent dimension. We also analyze the quality of learned topics in terms of their similarity to LDA topics.

2. Background & Related Work

2.1. Clinical Outcome Prediction

The three main types of clinical data that have been used for clinical outcome prediction are textual notes (Ghassemi et al., 2014; Jo & Rosé, 2015; Grnarova et al., 2016), real-valued measurements (e.g., laboratory/physiologic measurements) (Lipton et al., 2015), and categorical measurements (e.g., medical codes).

Textual clinical notes contain qualitative information that cannot be found in numeric measurements, such as insights from nurses and doctors, patients' progress, and social context (e.g., relationships with family and friends). Clinical

Project for 10-708: Probabilistic Graphical Models. Spring 2017.

notes were found to be helpful for long-term prediction, but not as much for short-term prediction (Jo & Rosé, 2015). Numeric measurements provide useful insight into the patients current health condition and health record.

Some of the main challenges in analyzing textual clinical notes include unstructured text, incomplete sentences/phrases, irregular use of language, abundant abbreviations, and rich medical jargons and their variations. These characteristics make it difficult to apply NLP tools, such as part-of-speech taggers, dependency parsers, named-entity recognition, etc. Topic modeling is one of the techniques applied to tackle these problems. Ghassemi et al. (2014) did a cross-sectional analysis of topics to predict mortality, and later, Jo and Rosé (2015) did a time series analysis using a joint model of HMM and LDA to find patients' latent states and state transition patterns.

Both studies offer good interpretability due to the topics learned from the notes, but the time series analysis with Markovian assumption in the latter study achieved only a minor improvement from the former cross-sectional analysis. Our model can hopefully capture more complex time dependencies in patient trajectories by using an LSTM. Recently, a joint model of LSTM and convolutional neural network (CNN) was used to predict mortality and found phrases that are highly related to mortality (Grnarova et al., 2016). This model outperforms the previous models, but it is cross-sectional and has limited interpretability. Our work may improve this model by introducing progression analysis and interpretable topics.

For real-valued time series measurements, Chia and Syed (2014) predicted mortality using the variability of ECG signals for each patient measured by dynamic time warping between every pair of consecutive heart beats. Chia and Syed (2013) also used time series heart rate patterns for mortality prediction, by binning each heart rate (per minute), clustering subsequences of the bins, and choosing clusters that are indicative of mortality and survival respectively. Recently, LSTM without feature engineering was used for diagnosis prediction, where 13 types of time series data (e.g., blood pressure, blood oxygen saturation) were resampled to an hourly rate by taking the mean value and then put into an LSTM for prediction (Lipton et al., 2015). However, this study found evidence that even simple statistics (e.g., max, min, mean) of the measurements throughout the entire timeline of a patient achieve almost comparable accuracy with a simple MLP.

2.2. Neural Network for Topic Modeling

Our goal is to design a neural network architecture, where the upper layer is LSTM for predicting mortality, and the lower layer feeds the LSTM at each time point with a latent topic distribution learned from clinical notes. In this section, we review prior work on modeling topic distributions using neural networks.

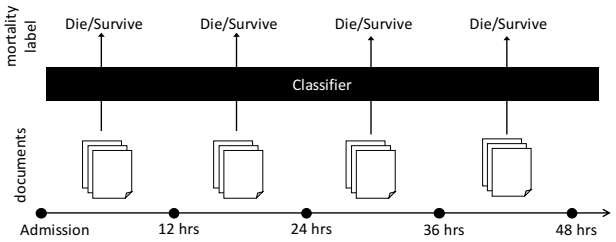

Figure 1. Framework of our mortality prediction task. We define time points as 12 hour-long time segments of a patient's timeline from admission (e.g., time point 1 is the first 12 hours, time point 2 is the next 12 hours, etc.) We aggregate all clinical notes in each time point into one document. At each time point t, we use all documents up to t to predict the mortality of the patient at t.

One of the simplest and earliest approaches is a restricted Boltzmann machine (Srivastava et al., 2013; Hinton & Salakhutdinov, 2009). The activation probability of each hidden node is a sigmoid function on a weighted sum of the frequency of individual words. Hence, a hidden node can be interpreted as a latent topic that has a weight associated with each word. Inspired by these models, a deep sigmoid belief network has been proposed to learn topic distributions in a supervised way given a bag-of-words, and there has been an attempt to interpret the hidden layers (Zhang et al., 2016). In the same vein, we try to interpret topics from the hidden layers of our joint models LSTM+E and LSTM+E+D.

In a more recent model, TopicRNN, the topic distribution of text is assumed to be drawn from a normal distribution whose mean and variance are computed from the input BoW using a neural network (Dieng et al., 2016). This topic distribution serves as the global semantics of the text, and is fed into an RNN to help predict the following word.

Topics learned by the above models are based on unigrams, but general n-grams may capture better topics for some tasks, as shown in (Zhai & Boyd-Graber, 2013; Hardisty et al., 2010; Wallach, 2006). A neural network model has been proposed that learns the association between an arbitrary n-gram and topics (Cao et al., 2015). This model, however, takes individual n-grams of interest as separate inputs, which makes training tricky. Recently, hierarchical LSTM has been proposed (Chung et al., 2016). Our other joint model, the hierarchical LSTM, exploits this architecture and explores the idea of using a lower LSTM layer for learning topic distributions from a sequence of words without limiting itself to unigrams.

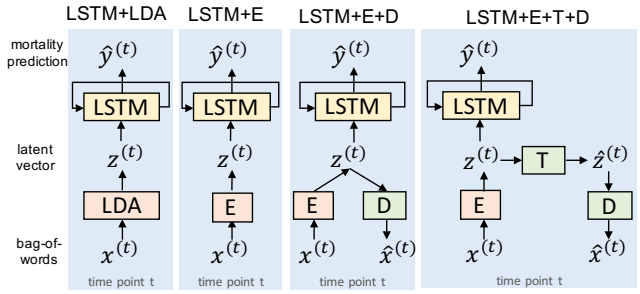

Figure 2. Overview of the LSTM models. In all four models, the input is a bag-of-words representation $x^{(t)}$ of the documents at time point t, which is then converted to a latent document embedding $z^{(t)}$ by either a pre-trained LDA model (as in LSTM+LDA) or an Encoder (E) network. The LSTM layer inputs z_t and generates a predicted outcome $\hat{y}^{(t)}$. In LSTM+E+D, we add a Decoder (D) that reconstructs bag-of-words data from $z^{(t)}$. In LSTM+E+T+D, we add an intermediate Transcoder (T) that maps the latent vector $z^{(t)}$ to a sparse topic vector $\hat{z}^{(t)}$, before the Decoder (D) converts $\hat{z}^{(t)}$ to a bag-of-words vector $\hat{x}^{(t)}$. The models are trained to minimize the prediction loss $H(\hat{y}^{(t)}, y)$ and (for LSTM+E(+T)+D) the reconstruction loss $H(\hat{x}^{(t)}, x^{(t)})$ over all time points.

3. Methods

Our task is to build a classifier for predicting a patent's mortality given his/her clinical notes written so far (Figure 1). Throughout the report, we define time points as 12 hour-long time segments of a patient's timeline from admission (e.g., time point 1 is the first 12 hours, time point 2 is the next 12 hours, etc.) (Ghassemi et al., 2014). For each patient, we aggregate all notes in each time point t into a bag-of-words representation x_t, normalized to sum to 1.

In this section, we present each of our models (see Fig. 2 for an overview). First, we present and compare two LDA baseline models to analyze the benefit of using LSTM over linear SVM for capturing long-term dependencies (Section 3.1). Then we present three end-to-end models which jointly learn topic models and mortality prediction (Section 3.2).

3.1. LDA baselines

Here, we present two baseline methods that use LDA, which has been used to infer topic distributions in textual clinical notes. Note that these models are not a joint model of topic modeling and mortality prediction, because they use a separate LDA model to train topics.

SVM+LDA is the model proposed by Ghassemi et al. (2014). This model builds a linear SVM for each time point that predicts a patient's mortality given the patient's clinical notes written up to that time point. To obtain the input for the classifier for time point t, we first compute the topic distribution of each note using LDA and then aggregate all topic distributions up to time point t by simply averaging the topic distribution vectors.

LSTM+LDA replaces the linear SVM with LSTM, allowing us to evaluate the ability of LSTM to capture time dependencies for mortality prediction. The LSTM inputs a sequence of topic distribution vectors $\boldsymbol{z} = (z^{(1)}, \ldots, z^{(T)})$, generates a predicted outcome $\hat{y}^{(t)}$ at each time point t, and is trained to minimize the prediction loss $H(\hat{y}^{(t)}, y)$ over all time points, where $y \in \{0, 1\}$ is the mortality label of the patient and $H(p, q)$ is the weighted cross-entropy defined as

$$H(p, q) = -C^{FN} q \log p - (1 - q) \log(1 - p), \qquad (1)$$

where C^{FN} is the cost for false negative classification.

Unlike the SVM baseline, LSTM+LDA does not build separate classifiers for different time points, and instead is trained to predict the correct outcome at any time point. By comparing these baselines, we try to establish whether an LSTM can infer relevant semantic information based on topic distributions. (As shown in Fig. 3, LSTM+LDA is more robust than SVM+LDA in predicting the mortality of patients who stay longer in the hospital.)

3.2. Joint Models

Next, we present three end-to-end models which jointly learn topic modeling and mortality prediction.

The **LSTM Encoder (LSTM+E)** model replaces the pre-trained LDA topic model in LSTM+LDA with an Encoder (E) network. The Encoder is a single-layer neural network that is expected to find latent topics. Due to the Encoder structure, the Encoder weights θ_E are reminiscent of LDA word-topic weights in that we can interpret the Encoder as a graphical model that relates words in $x^{(t)}$ to the latent topics in $z^{(t)}$ (see Section 5.2 for our analysis).

The Encoder of LSTM+E does not guarantee that cohesive words fall into the same hidden node, which may make hidden nodes difficult to interpret. Hence, **LSTM Encoder-Decoder (LSTM+E+D)** adds a Decoder (D), a single-layer network with no biases that tries to reconstruct the original bag-of-words data by minimizing the reconstruction loss $H(\hat{x}^{(t)}, x^{(t)})$ across all time points. Our rationale behind the Decoder is that the words with the highest weights for each hidden node are likely to be generated together in the same document through the Decoder, thus finding cohesive topics like LDA topics. We can interpret topics from the Decoder weights θ_D in the same way we do from the Encoder weights.

One potential drawback of LSTM+E+D is that latent document vectors z are tied to both the LSTM network and the

Decoder, thereby being optimized neither for mortality nor for topic interpretation. To relax this tie, **LSTM Encoder-Transcoder-Decoder (LSTM+E+T+D)** adds an intermediate Transcoder (T) that maps the latent vector $z^{(t)}$ to a sparse topic vector $\hat{z}^{(t)}$, before the Decoder (D) converts $\hat{z}^{(t)}$ to a bag-of-words vector $\hat{x}^{(t)}$.

Inspired by LDA's sparsity constraints on the document's probability distribution over topics and the topic's probability distribution over words, we impose L1 regularizations on the learned topic vector $\hat{z}^{(t)}$ and the Decoder's weights θ_D, respectively. Thus, our final loss function for LSTM+E+T+D is

$$\sum_{t=1}^{T}\left[\underbrace{H(\hat{y}^{(t)},y)}_{\text{prediction}}+\lambda_1\underbrace{H(\hat{x}^{(t)},x^{(t)})}_{\text{reconstruction}}+\lambda_2\|\hat{z}^{(t)}\|_1\right]+\lambda_3\|\theta_D\|_1 \tag{2}$$

where $\lambda_1, \lambda_2, \lambda_3$ are hyperparameters to control the weight of each loss.

Unlike LSTM+LDA, which uses a pre-trained topic model to generate topic vectors for documents at each time point, these joint LSTM models are end-to-end in that they jointly learn the topic models and the contexts between documents. As a result, they learn a better latent document representation $z^{(t)}$ for mortality prediction and outperform LSTM+LDA on all three mortality prediction tasks (see Fig. 3).

4. Experiments

4.1. Evaluation Metrics

Following Ghassemi et al. (2014; 2016), we evaluate our model on three mortality prediction tasks: in-hospital mortality, 30-day post-discharge mortality, and 1-year post-discharge mortality. These three tasks cover both short-term and long-term mortality. We measure accuracy via the Area Under ROC Curve (AUC) metric (Rakotomamonjy, 2004), which captures how well a trained classifier discriminates between positive instances and negative instances.

4.2. Data and Preprocessing

We use the MIMIC-III dataset which contains data about patients admitted to critical care units at a large tertiary care hospital (Johnson et al., 2016). For data preparation, we followed Ghassemi et al. (2014)'s work as closely as we could, as their setting has been widely used (Jo & Rosé, 2015; Grnarova et al., 2016). We also excluded patients who are less than 18 years old; these patients (mostly infants) show very different trajectories from adult patients (Jo & Rosé, 2015). We used all textual clinical notes except discharge summaries because they explicitly mention the patient's outcomes. We randomly split patients into training, validation, and test sets with the ratio of 6:2:2. Since the classes are severely skewed toward negative (i.e., survival), the negative instances in the training set are downsampled such that negative instances constitute no more than 70% of the training set. The validation and test sets are not downsampled.

# patients	36,218	# unique words	83,176
Seq. len (median)	13	Seq. len (max)	1,145
Doc. len (median)	113	Doc. len (max)	2,507

Table 1. Data statistics for MIMIC-III after preprocessing. Here, "Seq. len" refers to the last time point of a patient's timeline from admission, and "Doc. len" refers to the number of words in the concatenation of the notes at a time point t.

For preprocessing of the text, we first normalized some text into categories to cluster meaningful text. We replace deidentified information in text with the given category (e.g., "[** First Name 3 **]" → "##firstname##"). We also replace times and numbers with "##time##" and "#" using regular expressions. Next, we tokenized the text with non-alphanumeric letters. We removed stop words using the Onix stop word list[1].

To reduce the vocabulary size, we retained at most 500 words that have the highest tf-idf among all documents for each patient in the training set, and only included these words into the vocabulary. We excluded subjects from the training set if the total length of the clinical notes is less than 100 words. The statistics of the final data after preprocessing are listed in Table 1.

For our LSTM models, we handle missing time points by using zero vectors for documents $x^{(t)}$. Note that this setting does not affect the prediction and reconstruction loss.

4.3. Models and Parameters

For the LDA baseline, the number of topics is set to 50 (Ghassemi et al., 2014). The cost C and the weight w for the positive class ("died") in libsvm are explored on the grid of $C = \{2^{-5}, 2^{-3}, 2^{-1}, 2^{1}, 2^{3}, 2^{5}, 2^{7}, 2^{9}, 2^{11}, 2^{13}, 2^{15}\}$ and $w = \{1, 3, 5, 7, 9\}$ and tuned on the validation set. The weight for the negative class ("survived") is fixed to 1.

The network configuration of our models is summarized in Table 2. The batch size is 10, the number of training steps is 100,000, and the learning rate is 0.001. As in the SVM, we explored different false negative costs (Eq. 1) $C^{FN} = \{2^0, 2^1, 2^2, 2^3\}$ and chose the optimal value based on the validation set.

[1] http://www.lextek.com/manuals/onix/stopwords1.html

	LSTM+LDA	LSTM+E	LSTM+E+D	LSTM+E+T+D
# nodes in the topic layer	50	50	50	50
# nodes in the LSTM hidden layer	128	128	128	128
LSTM activation	Softmax	Softmax	Softmax	Softmax
Encoder activation	-	ReLU	ReLU	ReLU
Transcoder activation	-	-	-	ReLU
Decoder activation	-	-	Softmax	Softmax

Table 2. Network configurations for each of our models.

For the loss function (Eq 2), we explored $\lambda_1 = \{10^{-2}, 10^{-1}, 10^{0}, 10^{1}\}$ and $\lambda_2, \lambda_3 \in \{0, 1\}$. Ultimately, we found that $\lambda_1 = 10^{-2}$, $\lambda_2 = 0$, $\lambda_3 = 1$ resulted in the highest AUC score for mortality prediction on the validation set, and used these settings for our experiments.

5. Results

We evaluate our models on both mortality prediction (Section 5.1) and the quality of learned topics (Sections 5.2 and 5.3).

5.1. Mortality Prediction Accuracy

In Fig. 3, we plot the AUC score of each model for each time point on the three mortality prediction tasks. The joint models LSTM+E(+T)(+D) outperform the LDA baselines on all three mortality prediction tasks. For example, LSTM+E outperforms LSTM+LDA by about +4% (Hospital), +2% (30Days), +7% (1Year) on average.

The LSTM-based approaches show less accuracy drop over time than the SVM+LDA baseline, notably for the in-hospital prediction. One of the reasons for SVM+LDA's performance drop is having fewer training instances for later time points, as patients who have died or been discharged at a certain time point are excluded from the training set (Ghassemi et al., 2014). According to the in-hospital prediction, LSTM seems to be able to relieve this problem. We suspect that the way we define our cost function—the average of losses at previous time points—might have compensated for the data loss. Another reason for performance drop is that it is more difficult to predict the destiny of patients who stay in an ICU for long. Long-term time dependencies captured by LSTM might have help make stable prediction.

We introduced the cost C^{FN} for false negative classification to compensate for the fewer number of positive instances, but this cost had inconsistent effects. For example, we found that a cost greater than 1 improved the accuracy for in-hospital and 1-year post-discharge mortality prediction, but decreased the accuracy for the 30-day post-discharge prediction.

The latent document vectors learned by our joint models have a better ability to separate documents by mortality rate. This is demonstrated in Fig. 4, where we visualize the latent document vectors $z^{(t)}$ for each model, and see that the joint models, LSTM+E(+D), have documents with the same mortality label clustered more closely.

5.2. Topic Interpretation

A popular method for qualitatively analyzing latent topics is to examine the top words of each topic. We expected each hidden node in the Encoder to represent a cohesive topic that is indicative of mortality. We tried interpreting the hidden nodes by examining the words that have the highest weights for each hidden node. As shown in Table 3b, the top words consist largely of typos and infrequent words and thus hardly represent an interpretable notion of topics compared to LDA topics in Table 3a. We conclude that (I) mortality is related with certain words individually rather than as a group like an LDA topic, and (II) the Encoder weights cannot find cohesive topics partly because words in the same document are not encouraged to be tied to the same hidden node.

We introduced the Decoder of LSTM+E+D to alleviate (II). We tried interpreting topics from the Decoder weights in the same way we did from the Encoder weights. As shown in Table 3c, the top words consist of more frequent words than do the top words from the Encoder weights. Yet, there still appear many typos and infrequent words, and the topics are quite difficult to interpret. This does not necessarily mean that topics are not cohesive. Due to the limited time, we could not further investigate the learned topics, which consist of a lot of medical terms and jargons. We leave this for future work.

The topics interpreted from the Decoder weights of LSTM+E+T+D are shown in Table 3d. We expected that the sparsity constraints for $\hat{z}^{(t)}$ and θ_D imposed on this model would produce more interpretable, cohesive, LDA-like topics, but the learned topics do not show significant difference from those from LSTM+E+D. Moreover, we found that L1 regularization on $\hat{z}^{(t)}$ caused the AUC to fall, while L1 regularization on the Decoder weights

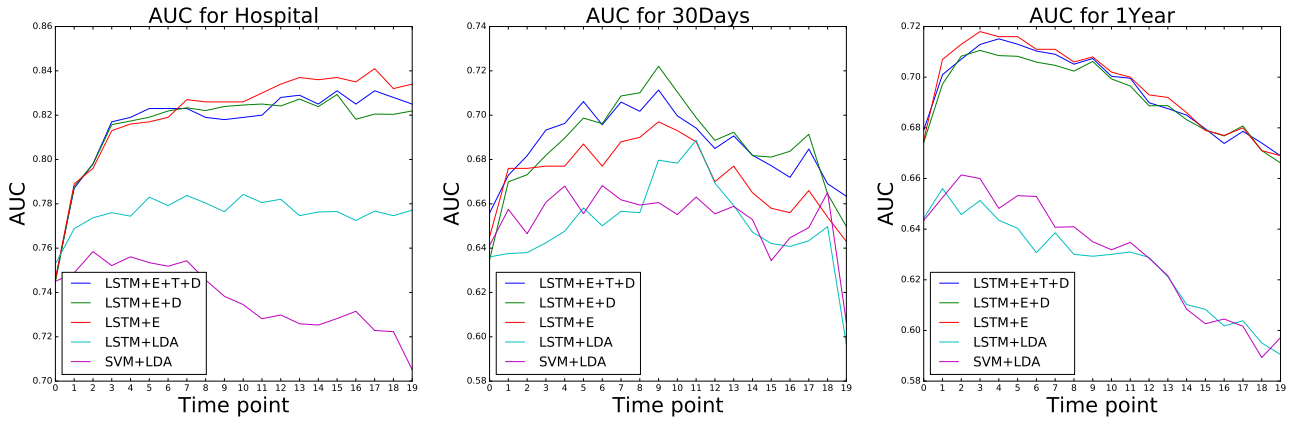

Figure 3. Accuracy on predicting in-hospital mortality (left), 30-day post-discharge mortality (center), and 1-year post-discharge mortality (right) at every time point. We use the Area Under ROC Curve (AUC) metric. The LSTM+E(+D) models outperform the LDA baselines on all three mortality prediction tasks.

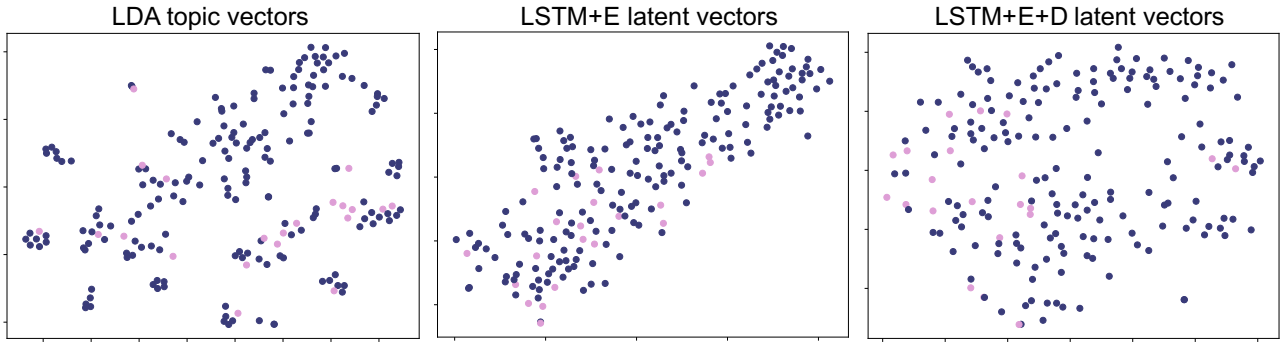

Figure 4. We visualize how well the latent representations separate documents according to mortality label. Each dot corresponds to a latent document embedding $z^{(t)}$, colored by in-hospital mortality label where pink indicates positive instances ("died") and dark blue indicates negative instances ("survived"). The joint models LSTM+E(+D) produce latent representations that better separate documents corresponding to mortality rate: the pink dots are more closely clustered together in the "LSTM+E(+D) latent vectors" plots, whereas the pink dots are more spread apart in the "LDA topic vectors" plot.

θ_D increased the AUC. However, these results might be due to using a suboptimal setting for topic modeling. We leave further experiments on finetuning $\lambda_1, \lambda_2, \lambda_3$ in Eq. (2) to balance the tradeoff between better mortality prediction performance and better latent topic modeling as future work.

Interpreting topics from single-layer encoder and decoder turns out to be extremely difficult. The Encoder seems to pick important words predictive of mortality, instead of finding cohesive topics. The structure of the Decoder is similar to probabilistic latent semantic indexing (PLSI), which computes the probability of word w in document d as follows:

$$p(w|d) = \sum_t p(w|t)p(t|d),$$

where $p(w|t)$ and $p(t|d)$ correspond to the input vector and the weights of the Decoder, respectively. We imposed sparsity on $p(w|t)$ and $p(t|d)$ in an attempt to make topics more interpretable, but we could not attain satisfactory topics. In order to make sure that the Decoder works as expected, we may need to run PLSI on the data and compare the result topics with those obtained from our models. The learned topics may not be interpretable simply because documents are noisy; each document is a concatenation of multiple clinical notes in the same time point that may be of very different types.

5.3. Topic Quality

We evaluate the quality of learned topics by analyzing the t-SNE plots of the latent document vectors in Fig. 5. We visualized the LDA topic vectors and the latent vectors of LSTM+E(+D) with colors representing patients (Fig. 5). We focused on the documents in the blue box in the left

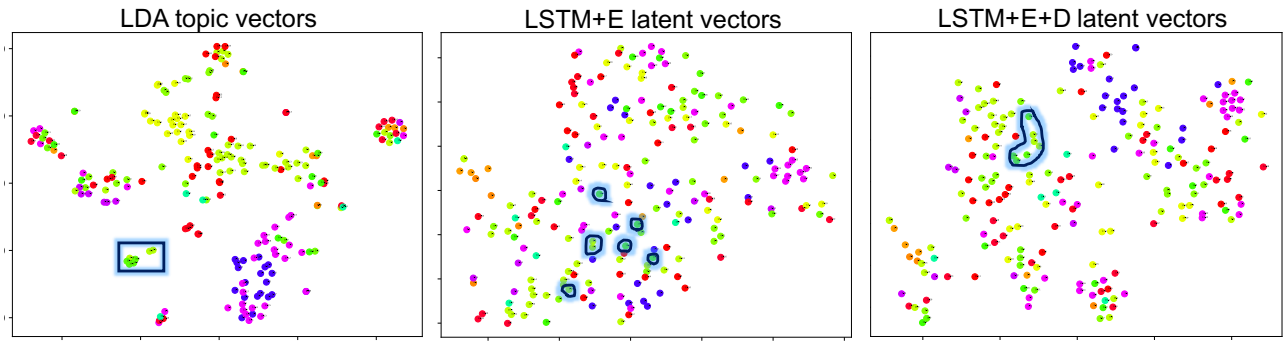

Figure 5. We visualize the topic clusterings produced by the latent vectors of each model. Each point represents to a document, and each color corresponds to a different patient. (We colored the dots by patient id because documents belonging to the same patient are more likely to have the same topics, and so it is slightly easier to visualize topic clusters.) Generally, we observed that the Autoencoder model (LSTM+E+D) produces topic clusterings that are more similar to the LDA topic clusterings than the Encoder model (LSTM+E). For example, consider the cluster of documents inside the box in the "LDA topic vectors" plot. We circle the corresponding documents in the LSTM+E(+D) plots. Notice that in the LSTM+E+D plot, the documents are also clustered closely together, but in the LSTM+E plot, these documents are more dispersed. We observed this general trend for many LDA topic clusters, and concluded that the Decoder network indeed helps the model learn better topic distributions. We did not observe any significant difference in the quality of topic clusters learned by LSTM+E+D vs. by LSTM+E+T+D, so we omit the plot for LSTM+E+T+D here.

pane and looked at whether these documents are clustered closely in the LSTM+E (middle pane) and LSTM+E+D (right pane) plots. LSTM+E+D seems to produce a more similar clustering to LDA than LSTM+E does, and thus LSTM+E+D may outperform LSTM+E with respect to the first method. Note that we did not observe any significant difference in the quality of topic clusters learned by LSTM+E+D vs. by LSTM+E+T+D, so we omitted the plot for LSTM+E+T+D.

6. Conclusion

We presented three joint models of LSTM and topic modeling that combine the benefits of both (1) LSTMs, which can capture time dependencies for mortality prediction, and (2) topic modeling, which can interpret topics predictive of mortality. Our models improved the accuracy of mortality prediction significantly from the baseline LDA-based models and were able to learn latent document representations indicative of mortality. We also proposed a method for interpreting topics from our models based on the Encoder and Decoder weights. However, the words with the highest weights for each hidden node did not provide interpretable, cohesive topics as LDA topics do. This may imply that LDA-like topics are suboptimal as feature for mortality prediction, and more indicative information is conveyed rather by certain individual words. We tried to make the Decoder work in a similar manner to PLSI and LDA by imposing constraints, but it turned out to be extremely difficult to obtain interpretable LDA-like topics from the Decoder. This noise might come from our concatenating multiple clinical notes of different types in the same time point.

Future work

To understand which words are indicative of mortality and survival, we can investigate which words trigger each hidden node in the LSTM. To do this, we calculate the derivative of the value of each hidden node in the LSTM cell in terms of each input word, while fixing the weights of the whole network after they are trained. By training in this way, our model may be able to generate a bag-of-words vector that maximizes each hidden node in the LSTM.

We observed that most of the top words for each hidden node (ranked by the Encoder or Decoder weights) are typos (e.g., "recomendations", "evenign") or rare words. To make these top words more interpretable, we suggest preprocessing the data in the following ways: (1) Remove the n most frequent and infrequent words. (2) Use a spell-checker to fix typos. Since electronic health records have many medical jargons, we want to be careful and only replace 1-character typos (e.g., correct "cooeprative" to "cooperative").

In Section 5.3, we evaluated the quality of learned topics by analyzing the t-SNE plots of the latent document vectors. For future work, we suggest a method to quantitatively evaluate the learned topics by comparing the clusters they form to a gold standard. More specifically, for each latent vector, we identify k nearest neighbors and compute the overlap with gold standard neighbors of the vector. Since no ground truth quality scores are available, we tried two gold standards: LDA topics and patient identity. In the first method, we identify the k nearest neighbors for each LDA latent vector and compute the overlap with the k nearest neighbors of the same document vector for LSTM+E and

T0	#, pt, impaired, ##date##, activity, sit, status, balance, mobility, stand
T1	#, ##date##, ##time##, am, dl, mg, meq, weight, arterial, nutrition
T2	tube, #, placement, chest, reason, line, ##date##, tip, ##time##, left
T3	#, insulin, gtt, blood, dm, hr, diabetes, patient, continue, type
T4	chest, ##date##, reason, ap, #, portable, left, ##time##, examination, report

(a) Top 10 words with the highest weights in the LDA model

T0	barium, dsfx, diagonosis, darts, ndka, swallon, palplabe, cpapwith, ileoileo, laminar
T1	evenign, tranversing, #dex, bronchusplan, ecxema, degenerate, flunisolide, sangiunous, sternotony, interstitices
T2	thorcic, medisternal, proxtmal, merry, reslts, depressant, presacral, godchild, killed, q#am
T3	lb#, wenckebach, quaderant, aticoagulated, nonproducted, emg, en, hypotherm, faa, dalaudid
T4	diagnosing, gallstones, retroardiac, parieto, lymphoproliferative, isoview, pati, femoroacetabular, periampullary

(b) Top 10 words with the highest Encoder weights θ_E in LSTM+E

T0	tacromulius, kerly, parathroid, programming, coure, placeemrnt, toilteting, reveals, intervan, pheochromocytoma
T1	filmr, dap, nugauze, extubed, intracystic, plsn, replete, implements, intraparotid, wrote
T2	sunburn, infrahilar, peppermint, extenisve, #reisman, intraconal, overwhich, isoview, evevning, ischiorectal
T3	finesteride, helmut, rhabdomyo, hepatopulmonary, displ, comparison#, franc, doudoerm, posterioly, diminutive
T4	relan, interluminal, reinserted, peforation, reticulation, ooob, allohsct, cholangiograms, hemiparesus, contniued

(c) Top 10 words with the highest Decoder weights θ_D in LSTM+E+D

T0	recomendations, left, odorous, tachynea, #pcs, doxazosin, specifice, hydromyelic, message, spiration
T1	pharmacology, pancreati, cofirm, frn, eventially, wthdrawal, rslts, wean, famiily, rpb
T2	relaxation, baseclinical, titarted, sadk, lymphomatoid, bph, mahogany, atelelectasis, frontalis, dlr
T3	carcino, eea, orlsca, hospitization, captain, suringe, wellness, obstructed, agrestat, dilatation
T4	after#, syggestive, nutritions, leni, diruetics, diaphroetic, vase, breathsounds, insominia, perirectal

(d) Top 10 words with the highest Decoder weights θ_D in LSTM+E+T+D.

Table 3. Example topics T0, . . ., T4 and their top 10 words sorted by LDA weights (a) or the trained networks weights (b-d). The topics were chosen randomly from LDA and from the joint models trained on in-hospital mortality. Limited interpretability is offered by these top words, which include many typos (e.g., "evenign", "placeemrnt") and rare words or medical jargons (e.g., "tachynea", "doxazosin").

LSTM+E+D, respectively. Document representations similar to LDA topics may be considered good in that they represent cohesive, interpretable topics, but setting the standard to LDA topics is not optimal, as our ultimate goal is to find more indicative topics than LDA topics. In the second method, we assume that each patient has consistent topics, and for each latent vector, we compute the percentage of the k nearest neighbor vectors that are from the same patient. The assumption may not hold, however, if a patient's symptoms and condition are diverse and change significantly over time.

Since we interpret topics from the Decoder weights θ_D in LSTM+E(+T)+D, we can use arbitrarily deep Encoder and Transcoder networks to increase the complexity of our model, which may allow the model to learn better hidden representations for both topic modeling and mortality prediction. However, keep in mind that a more complex network requires more training data in order for the model to saturate.

References

Cao, Ziqiang, Li, Sujian, Liu, Yang, Li, Wenjie, and Ji, Heng. A novel neural topic model and its supervised extension. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pp. 2210–2216, 2015. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9303.

Chia, C C and Syed, Z. Scalable noise mining in long-term electrocardiographic time-series to predict death following heart attacks. In *Proceedings of the 20th ACM SIGKDD international . . .*, 2014.

Chia, Chih-chun and Syed, Zeeshan. Computationally Generated Cardiac Biomarkers: Heart Rate Patterns to Predict Death Following Coronary Attacks. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pp. 735–746. Society for Industrial and Applied Mathematics, Philadelphia, PA, December 2013.

Chung, Junyoung, Ahn, Sungjin, and Bengio, Yoshua. Hierarchical multiscale recurrent neural networks. *CoRR*, abs/1609.01704, 2016. URL http://arxiv.org/abs/1609.01704.

Dieng, Adji B., Wang, Chong, Gao, Jianfeng, and Paisley, John William. Topicrnn: A recurrent neural network with long-range semantic dependency. *CoRR*, abs/1611.01702, 2016. URL http://arxiv.org/abs/1611.01702.

Ghassemi, Marzyeh, Naumann, Tristan, Doshi-Velez, Finale, Brimmer, Nicole, Joshi, Rohit, Rumshisky, Anna, and Szolovits, Peter. Unfolding Physiological State: Mortality Modelling in Intensive Care Units. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 75–84, New York, NY, USA, 2014. ACM.

Grnarova, Paulina, Schmidt, Florian, Hyland, Stephanie L, and Eickhoff, Carsten. Neural Document Embeddings for Intensive Care Patient Mortality Prediction. *arXiv*, cs.CL, 2016.

Hardisty, Eric, Boyd-Graber, Jordan, and Resnik, Philip. Modeling perspective using adaptor grammars. *EMNLP '10: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, October 2010.

Hinton, Geoffrey E and Salakhutdinov, Ruslan R. Replicated Softmax: an Undirected Topic Model. In Bengio, Y, Schuurmans, D, Lafferty, J D, Williams, C K I, and Culotta, A (eds.), *Advances in Neural Information Processing Systems*, pp. 1607–1614. Curran Associates, Inc., 2009.

Jo, Yohan and Rosé, Carolyn Penstein. Time Series Analysis of Nursing Notes for Mortality Prediction via a State Transition Topic Model. *Proceedings of the 24th ACM International Conference on Information and Knowledge Management*, 2015.

Johnson, Alistair E W, Pollard, Tom J, Shen, Lu, Lehman, Li-wei H, Feng, Mengling, Ghassemi, Mohammad, Moody, Benjamin, Szolovits, Peter, Anthony Celi, Leo, and Mark, Roger G. MIMIC-III, a freely accessible critical care database. *Scientific ...*, 3:160035, May 2016.

Lipton, Zachary C, Kale, David C, Elkan, Charles, and Wetzell, Randall. Learning to Diagnose with LSTM Recurrent Neural Networks. *arXiv.org*, November 2015.

Rakotomamonjy, Alain. Optimizing Area Under Roc Curve with SVMs. *The 1st workshop on ROC analysis in artificial intelligence*, 2004.

Srivastava, Nitish, Salakhutdinov, Ruslan R, and Hinton, Geoffrey E. Modeling Documents with Deep Boltzmann Machines. *arXiv.org*, September 2013.

Wallach, H. Topic modeling: beyond bag-of-words. *Proceedings of the 23rd international conference on ...*, 2006.

Zhai, Ke and Boyd-Graber, Jordan L. Online Latent Dirichlet Allocation with Infinite Vocabulary. *Proceedings of the 30th International Conference on Ma- chine Learning*, 2013.

Zhang, Dongxu, Luo, Tianyi, and Wang, Dong. Learning from LDA using deep neural networks. In *Natural Language Understanding and Intelligent Applications - 5th CCF Conference on Natural Language Processing and Chinese Computing, NLPCC 2016, and 24th International Conference on Computer Processing of Oriental Languages, ICCPOL 2016, Kunming, China, December 2-6, 2016, Proceedings*, pp. 657–664, 2016. doi: 10.1007/978-3-319-50496-4_59. URL http://dx.doi.org/10.1007/978-3-319-50496-4_59.