Time-Aware Transformer-based Network for Clinical Notes Series Prediction

Dongyu Zhang DZHANG5@WPI.EDU
Data Science Program
Worcester Polytechnic Institute
Worcester, MA, USA

Jidapa Thadajarassiri JTHADAJARASSIRI@WPI.EDU
Data Science Program
Worcester Polytechnic Institute
Worcester, MA, USA

Cansu Sen CSEN@WPI.EDU
Department of Computer Science
Worcester Polytechnic Institute
Worcester, MA, USA

Elke Rundensteiner RUNDENST@WPI.EDU
Department of Computer Science
Worcester Polytechnic Institute
Worcester, MA, USA

Abstract

A patient's clinical notes correspond to a sequence of free-form text documents generated by healthcare professionals over time. Rich and unique information in clinical notes is useful for clinical decision making. In this work, we propose a time-aware transformer-based hierarchical architecture, which we call **F**lexible **T**ime-aware **L**STM **Trans**former (**FTL-Trans**), for classifying a patient's health state based on her series of clinical notes. FTL-Trans addresses the problem that current transformer-based architectures cannot handle, which is the multi-level structure inherent in clinical note series where a note contains a sequence of chucks and a chuck contains further a sequence of words. At the bottom layer, FTL-Trans encodes equal-length subsequences of a patient's clinical notes ("chunks") into content embeddings using a pre-trained ClinicalBERT model. Unlike ClinicalBERT, however, FTL-Trans merges each content embedding and sequential information into a new position-enhanced chunk representation in the second layer by an augmented multi-level position embedding. Next, the time-aware layer tackles the irregularity in the spacing of notes in the note series by learning a flexible time decay function and utilizing the time decay function to incorporate both the position-enhanced chunk embedding and time information into a patient representation. This patient representation is then fed into the top layer for classification. Together, this hierarchical design of FTL-Trans successfully captures the multi-level sequential structure of the note series. Our extensive experimental evaluation conducted using multiple patient cohorts extracted from the MIMIC dataset illustrates that, while addressing the aforementioned issues, FTL-Trans consistently outperforms the state-of-the-art transformer-based architectures up to 5% in AUROC and 6% in Accuracy.

1. Introduction

Background. Clinical notes, written by healthcare workers, are rich resources of a patient's health status. Expert insights and observations about patients in these documents can be tremendously valuable for supporting decisions on clinical diagnosis. Recently, machine learning models are being developed to support such clinical decision making by exploiting this treasure trove of clinical notes (Grnarova et al., 2016; Dubois et al., 2017; Boag et al., 2018; Huang et al., 2019; Alsentzer et al., 2019). Clinical notes naturally have a *multi-level sequential structure*. Namely, they correspond to a sequence of documents created over time with each document itself consisting of a sequence of words. Further, the actual timing, relative and absolute, of these notes themselves can hold the key to insights into the patient's progression related to a disease or its treatment.

Figure 1 depicts an example for three patients, where the note contents for all three patients are the same. However, the creation time and the chronological order of the notes differ. In our example, a model that only considers the content of the notes will treat these three patients similarly since their notes contain exactly the same content. Thus, this model would predict the same outcome for these three patients and fail to deliver their actual outcome (survived for the first patient and died for the other two patients). Another model that improves in capturing sequential information but without considering the time of occurrence will be able to distinguish between the second patient and the other two patients. However, this model will still suffer in the failure of distinguishing the first patient from the third patient in this case. Thus, we postulate that both sequential and temporal knowledge must also be incorporated to design an effective model for clinical prediction tasks.

State-of-the-Art. Many NLP techniques such as bag of words (BOW) or word2vec have been applied to clinical note prediction (Boag et al., 2018). However, they produce a patient representation that does not capture the language dynamics and sequential nor the contextual information of words. A hierarchical attention model has been proposed to utilize the nested sequential structure of clinical note sequences (Sen et al., 2019). However, recently, transformer-based architectures, over and over, have shown superior performance over recurrent architectures for many NLP tasks (Vaswani et al., 2017; Devlin et al., 2018; Radford et al., 2018, 2019; Lan et al., 2019; Yang et al., 2019). Inspired by these developments, recently, ClinicalBERT (Huang et al., 2019; Alsentzer et al., 2019) has been designed for medical NLP problems. It is an application of the BERT model (Devlin et al., 2018) pre-trained on a clinical corpus from the MIMIC-III dataset (Johnson et al., 2016). BERT-based models enforce a length constraint on the input text. ClinicalBERT, thus, splits notes into equal length subsequences ("chunks"). It generates a prediction for each chunk and then aggregates these predictions together to compute the patient-level prediction. However, this approach disregards the interrelations among clinical notes and their chunks. It also loses knowledge about the multi-level sequential information inherent to the series of clinical notes.

Challenges. The utilization of clinical note sequences for generating a patient-level decision faces the following challenges:

- *Complex interrelations among clinical notes and their chunks*: BERT-based methods (Huang et al., 2019; Devlin et al., 2018) impose an input text-length limitation. In an attempt to utilize information, a natural approach is to split notes into chunks and

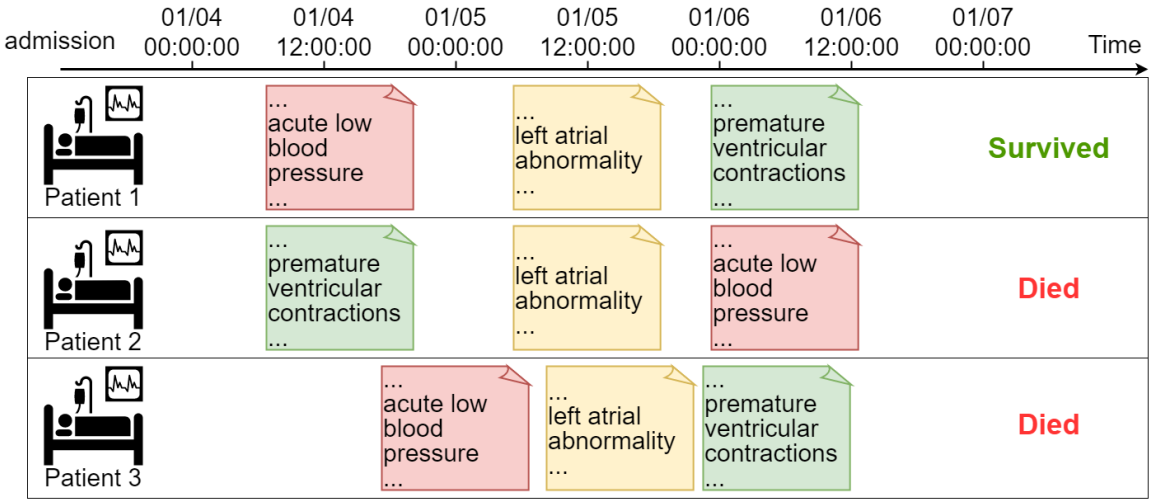

Figure 1: Examples of major changes in patients' states from minor variants of clinical note series. The three patients associate with the same content but distinct order and time of occurrence. Only the first patient survived while the others passed away.

then use these as input, as done by ClinicalBERT (Huang et al., 2019). However, in doing so, they lose the information about which chunk belongs to which note. Yet, this information can be crucial because chunks from the same note represent the patient's condition at a certain moment and together they supplement each other's information to form a complete picture of a patient at this moment. Hence, capturing the interrelations of a note with its respective chunks is necessary for making accurate clinical predictions.

- *Multi-level sequential structure*: Clinical note series by nature constitute a multi-level sequential structure. That is, they do not only correspond to a series of notes but also each note is composed of a sequence of words. Moreover, the necessity of chucking to incorporate arbitrarily long texts furthermore generates another hierarchical level into the clinical note data. This multi-level sequential structure is lost by current state-of-the-art methods that directly feed chunks into a model.

- *Unknown temporal importance*: In Figure 1, the order and the content of the clinical notes are exactly the same for the first and third patient. Yet the final outcome of these two patients differs dramatically. This implies that the temporal information plays an important role in predicting patients' outcomes. Thus an ideal model must have a time-aware design. However, the relative temporal importance of medical events to clinical outcome is unknown. Thus, a time-aware design is needed capable flexibly capturing the patterns of temporal importance.

Proposed Method. To overcome these challenges, we propose a novel hierarchical model structure, FTL-Trans[1], to learn patient representations from clinical notes. Our

1. FTL-Trans source code available at https://github.com/zdy93/FTL-Trans

model design takes the interrelations among chunks and notes into account. Further, FTL-Trans leverages both the time and multi-level sequential information inherent in clinical notes. FTL-Trans consists of four successive layers: (1) *Chunk Content Embedding Layer* encodes the text of each chunk into a content embedding utilizing a transformer-encoder layer, initialized with ClinicalBERT weights (Huang et al., 2019); (2) *Position-Enhanced Chunk Embedding Layer* merges each content embedding and sequential information of both the note and its contained chunk into a single representation. The note position information is represented by a *Global Position embedding*, while the chunk position information is encoded by a *Local Position embedding*; (3) *Time-Aware Layer* implements a novel **F**lexible **T**ime-aware **LSTM** (**FT-LSTM**) cell to incorporate temporal information into the chunk representation learned from the Position-Enhanced Chunk Embedding Layer for generating heuristic patient representations; and lastly the (4) *Classification Layer* generates a patient-level prediction using this learned patient representation.

We compare FTL-Trans's performance with state-of-the-art methods including BERT (Devlin et al., 2018) and ClinicalBERT (Huang et al., 2019) for five clinical tasks including in-hospital mortality, 30-day readmission prediction, Escherichia Coli Infection prediction, Enterococcus Sp. infection prediction, and Klebsiella pneumoniae infection prediction. These tasks are all extracted from the MIMIC dataset (Johnson et al., 2016). Our experimental evaluation illustrates that FTL-Trans constantly outperforms the state-of-the-art models up to 5% in Area Under the Receiver Operating Characteristic curve (AUROC) and up to 6% in accuracy.

Clinical Relevance. During a patient's stay in the hospital, tens or even hundreds of clinical notes are created. Each note in turn consists of hundreds of words. Healthcare workers from doctors to nurses document the patients' status and their diagnosis in their notes as they care for the patient. In addition, they also note clinical decisions after reviewing previous treatment with the help of these notes. However, since the clinical notes are too long to carefully read, potentially valuable information in the clinical notes may go unnoticed by clinicians - potentially risking health or even life. Also, the sequential order and timing of clinical notes are both known to potentially be critical indicators on the outcome of the health status of a patient. In this work, we propose a deep learning model that exploits clinical notes for making patient-level predictions about critical medical conditions, including hospital-acquired infections, in-hospital mortality, and re-admission. While our approach is not the first to utilize clinical notes for prediction tasks, we design a unique model architecture for exploiting the multi-level sequential structure and temporal information characteristic of clinical notes. These tend to frequently be ignored by previous works. Healthcare professionals could potentially gleam additional complementary information about a patient' health conditions beyond their own perspective and diagnosis. Alerts generated by our model may help medical staff intervene earlier in a patient's treatment due to identifying a possible concern more swiftly. In particular, the value and importance both of multi-level sequential information and temporal information of clinical notes are being revealed by our work. Future clinical research which focuses on reviewing the treatment process can make use of our model to analyze the impact of order and timing of medical intervention on a patient's clinical outcome.

Generalizable Insights about Machine Learning in the Context of Healthcare

In general, the following insights can be obtained from this work.

- The structure of clinical notes sequences naturally corresponds to a multi-level hierarchy, rendering it challenging to develop a model that effectively utilizes these rich resources for predicting a patient's outcomes. FTL-Trans can capture the multi-level sequential structure of clinical notes sequences by its hierarchical model design and the novel global and local position embeddings. Our evaluation results show that FTL-Trans consistently outperforms other state-of-the-art transformer-based architectures which do not consider the multi-level sequential structure.
- Clinical notes are in practice not regularly-spaced across time. In fact, we find that the temporal information of clinical notes plays an important role in predicting patients' outcomes. We introduce FT-LSTM, a novel time-aware LSTM cell structure, into our proposed FTL-Trans deep learning architecture. This component successfully handles temporal information of clinical notes and this way effectively improves the predictions.

2. Related Work

Clinical Note Classification Many works have utilized clinical notes for classification tasks. Lehman et al. (2012) applied the Hierarchical Dirichlet Processes (HDP), a topic learning technique, to extract the topic distribution of clinical concepts from clinical notes to predict medical outcomes. Grnarova et al. (2016) uses a convolutional neural network (CNN) to represent a clinical note for the mortality prediction. Dubois et al. (2017) propose two different approaches to summarize clinical notes into a patient representation. One of them is to use GloVe (Pennington et al., 2014) to learn word embeddings and then aggregate these word embeddings into patient-level embedding. The other approach is to use a Recurrent Neural Network (RNN) with an embedding layer to construct the patient representation. Boag et al. (2018) investigate the bag of words (BOW), word2vec, and the combination of word2vec and Long Short-Term Memory (LSTM) to build representations for clinical notes. Then they compare their performance on multiple clinical tasks. The result suggests that different representations have different strengths.

However, these aforementioned works (Boag et al., 2018; Dubois et al., 2017) are based on BOW, word2vec, and Glove, which are context-independent models, that is, they cannot capture word position information. Sen et al. (2019) propose a hierarchical RNN with hierarchical attention for classification of documents series, which incorporates the position information of words by RNN. But it can only handle a fixed number of words from each note, while the rest of the words are ignored.

More recently, BERT model (Devlin et al., 2018) has achieved significant success in many NLP tasks by pre-training a deep bidirectional representation on unlabeled text, jointly conditioned on both left and right contexts. BERT architecture takes the context and the order of words into account. Owing to this success, variants of the BERT model has been proposed for the clinical domain. In particular, ClinicalBERT (Huang et al., 2019), an application of the BERT model to the clinical domain, is pre-trained on clinical notes from the MIMIC dataset (Johnson et al., 2016). Since trasformer-based models impose a length

constraint on the input text, ClinicalBERT splits clinical notes into equal-length chunks and makes a prediction for each chunk. The prediction for the patient is then an aggregation of predicted values from each chunk. This approach does not consider the creation time of each clinical note. Further, it also ignores the multi-level sequential information in the sequence of clinical notes.

Time-Aware Models Traditional recurrent neural networks, such as RNN and LSTM, often make the assumption that the time gaps between the elements of a sequence are uniformly distributed. This assumption does not always hold in real-world data. Hence, some methods incorporate time information into the model to address this time irregularity issue. RETAIN model (Choi et al., 2016) examines EHR data in reverse time order to assign higher attention to recent clinical visits. T-LSTM, a variant of LSTM, takes the elapsed time between events into account (Baytas et al., 2017). The memory cell in T-LSTM is adjusted in a way that longer the elapsed time, smaller the effect of previous memory to the current output. In Chen et al. (2017), a time-aware attention mechanism is proposed. Time differences between events are used to decay the weight of previous events before being fed into the contextual module. Su et al. (2018) propose a universal time-decay function to mimic the complex contextual patterns in dialogues.

Bai et al. (2018) propose Timeline, an interpretable deep learning model. Timeline learns time decay factors for a disease so the long term impact of chronic events and the short-term effect of acute events can be captured separately. Kumar et al. (2019) propose a temporal attention layer to project the user embeddings in the sequential user-items interactions. The temporal attention layer converts elapsed time into a time-context vector and has an element-wise product with the previous embedding. ATTAIN (Zhang et al., 2019) is a time-aware disease progression model that utilizes the attention mechanism to generate decay weights from the time interval between previous events and the current event. It then uses these weights to discount previous memory cells. These models assume that the influence of previous events decays over time, which may not be true in some cases. Some previous works (Chen et al., 2017; Baytas et al., 2017) use a fixed function of time intervals to capture the change of temporal importance. Yet this does not always reflect the actual change of temporal importance. However, the change of temporal influence over time is task-dependent. For different clinical tasks, the influence of clinical event changes over time may exhibit distinct trends.

3. Cohort

Our experiment data is extracted from the MIMIC-III (Medical Information Mart for Intensive Care III) database (Johnson et al., 2016). MIMIC-III is comprised of de-identified health data associated with over 40,000 patients who stayed in intensive care units of the Beth Israel Deaconess Medical Center in Boston, MA, between 2001 and 2012. MIMIC-III is a commonly used dataset in clinical machine learning studies (Baytas et al., 2017; Huang et al., 2019; Alsentzer et al., 2019; Sen et al., 2019). We extract clinical notes for a variety of patient cohorts from the *NoteEvents* table for our evaluation tasks. The creation times of the clinical notes we use in our model are the *charttime*s from the *NoteEvents* table. In *NoteEvents* table, there are 2,083,180 notes from 15 categories. These categories are Case Management, Consult, ECG, Echo, Discharge summary, General Nursing, Nursing/other,

Nutrition, Pharmacy, Physician, Radiology, Rehab Services, Respiratory, Social Work. We use all types of notes unless otherwise stated in the cohort-specific descriptions. We extract five cohorts from the MIMIC datasets, details of which are provided below.

- **In-hospital Mortality Prediction.** To form our in-hospital mortality cohort, we use the *hospital_expire_flag* from the *Admissions* table. If this flag is set to 1, this indicates that the patient has passed away in the critical care unit. There are 5,854 admissions where *hospital_expire_flag*=1. We use the clinical notes of a patient from their admission until one day before the patient's death. This is to make sure no direct mention of a patient's outcome is included in the data. Therefore, the patients who have only stayed one day are filtered out, because all of their notes are from the date of death or discharge. Remaining 5,287 patients form our mortality-positive cohort. Then we subsample a equal-size negative cohort of 5,287 patients among the ones where *hospital_expire_flag*=0. We remove the clinical notes from the day of discharge, as well as discharge summaries, for this negative cohort, as these notes tend to include the patient's clinical outcome.

- **30-day Readmission Prediction.** In the *Admissions* table, re-admitted patients without scheduled appointments within 30 days of a prior discharge date are marked with a readmission flag. All other admissions are considered negative. We follow the data extraction procedures in Huang et al. (2019) to filter out the in-hospital death and newborn admissions. The remaining 2,960 cases form our readmission-positive cohort. Then we subsample a negative cohort of 2,960 cases among the readmission-negative set.

- **Escherichia Coli (E. coli) Infection Prediction.** To form the Escherichia Coli Infection prediction cohort, we use the table *MicrobiologyEvents* for locating the tests associated with the organism *80002 - Escherichia Coli*. Patients with at least one positive result for this microorganism are labeled positive. Patients with no record of this test are labeled negative. We use the clinical notes of a patient from their admission until one day before the time of the microbiology test. This is to make sure no direct mention of a patient's test result is included in the data. Patients who receive this test result within the first day of their admission are filtered out because all of their notes before the test are from the same date of the test. Of 3,082 E. coli-positive patients, 1,894 patients are remained to form the E. coli positive cohort. We randomly subsample 1,894 admissions among the negative patients to form the negative cohort. For the negative cohort, clinical notes up until the half-way of patient's hospital stay are used, following common practice (Wiens et al., 2012; Sen et al., 2017). For the same reason mentioned in the mortality cohort descriptions, we do not keep discharge summaries in this cohort.

- **Enterococcus Species (Enterococcus Sp.) Infection Prediction.** For the Enterococcus Sp. infection cohort, we use the table *Microbiologyevents* for locating the test associated with the organism *80053 - Enterococcus Sp.*. Patients with at least one positive result for this microorganism are labeled positive. Patients with no record of this test are labeled negative. We use the same procedure as that for E. coli to filter out the clinical notes that we cannot use. Of 2,884 Enterococcus Sp. positive patients,

2,301 have enough note events. These patients form the Enterococcus Sp. positive cohort. We randomly subsample 2,301 admissions among the negative patients to form the negative cohort. We use half-way of the negative admissions data and as what we do in the E. coli infection cohort. Also, discharge summaries are not used here.

- **Klebsiella pneumoniae (K. pneumoniae) Infection Prediction.** To form the Klebsiella pneumoniae infection cohort, we use the table *microbiologyevents* for locating the tests associated with the organism *80004 - Klebsiella pneumoniae*. Patients with at least one positive result for this microorganism are labeled positive. Patients with no record of this test are labeled negative. Following the same procedure as for E. coli and Enterococcus, we filter out the clinical notes that cannot be used. Of 1,575 K. pneumoniae positive patient admissions, 1,046 have enough note events. These patient admissions form the K. pneumoniae positive cohort. We randomly subsample 1,046 admissions among the negative patient admissions to form the negative cohort. We use half-way of the negative admissions data. We also drop the discharge summaries.

3.1. Data Preprocessing

Following the same procedure as for ClinicalBERT (Huang et al., 2019), we first remove punctuation and lowercase text in clinical notes. Private information such as the names of medical staff and patients' are de-identified due to privacy concerns. We then remove all de-identified information in notes. For the notes which do not have *charttime* value, we use 23:59:59 in their corresponding *chartdate* as their *charttime*. Then, we use WordPiece embedding (Wu et al., 2016) to tokenize each note and split it into equal-size chunks. We use 128 as the chunk size (i.e., 128 tokens in each chunk). Statistics about all five datasets are presented in Table 1.

Table 1: Cohort Statistics

Statistics		Dataset				
		Mortality	Readmission	E. coli	Enterococcus Sp.	K. pneumoniae
# Notes / Patient	Mean	36.10	32.10	27.60	26.61	29.27
	Median	16	15	13	13	13
# Words / Note	Mean	227.82	291.43	214.35	207.33	210.41
	Median	146	164	140	141	140
Total # Notes		381718	190004	104531	122473	61164
Total # Patients		10574	5920	3788	4602	2092

4. Methodology

4.1. Problem Definition

For a patient cohort consisting of K patients, the sequence of clinical notes associated with the k-th patient can be represented as $\mathcal{N}^{(k)} = \{N_1^{(k)}, N_2^{(k)}, ..., N_{m^{(k)}}^{(k)}\}$ along with their corresponding creation time $\mathcal{T}^{(k)} = \{t_1^{(k)}, t_2^{(k)}, .., t_{m^{(k)}}^{(k)}\}$ where $m^{(k)}$ denotes k-th patient's

total number of clinical notes. For $k, l \in [1, K]$, $m^{(k)}$ does not necessarily equal $m^{(l)}$. Each note $N_i^{(k)}$ contains a sequence of tokens $\{w_1^{(k)}, w_2^{(k)}, ..., w_{n_i}^{(k)}\}$ where the total number of tokens $n_i^{(k)}$ can vary in each note. We aim to build a model that uses the set of patient's clinical notes sequence $\mathcal{N} = \{\mathcal{N}^{(1)}, \mathcal{N}^{(2)}, ..., \mathcal{N}^{(K)}\}$, the corresponding creation time sequence $\mathcal{T} = \{\mathcal{T}^{(1)}, \mathcal{T}^{(2)}, ..., \mathcal{T}^{(K)}\}$, and the true labels of the patients' clinical outcome $\mathcal{Y} = \{y^{(1)}, y^{(2)}, ..., y^{(K)}\}$, $y^{(k)} \in \{0, 1\}$ (e.g., 1 indicates an in-hospital death and 0 indicates survival) to predict the probability of a patient having the positive clinical outcome (e.g., in-hospital death). For patient k, the goal is to learn a mapping:

$$< \mathcal{N}^{(k)}, \mathcal{T}^{(k)} > \longrightarrow P(label = 1 | \mathcal{N}^{(k)}, \mathcal{T}^{(k)}) \tag{1}$$

where $P(label = 1 | \mathcal{N}^{(k)}, \mathcal{T}^{(k)}) \in [0, 1]$. As we discussed in the beginning of this paragraph, $\mathcal{N}^{(k)}$ is an **ordered** sequence of notes, and each note in it is also an **ordered** sequence of tokens. This multi-level sequential information in $\mathcal{N}^{(k)}$ along with the time information $\mathcal{T}^{(k)}$ is taken into account by our model. For simplicity in the following sections, we describe our method for a single patient and drop the superscript (k) hereafter.

Table 2: Basic Notation

Notation for A Single Patient	
Notation	Explanation
N_i	The patient's i-th clinical note
m	Total number of note for the patient
o_i	Total number of chunks in note N_i
$N_i C_j$	The j-th chunk in note N_i
p	Total number of token in each chunk
$w_l^{N_i C_j}$	The l-th token in chunk $N_i C_j$
$E_j^{N_i}$	The chunk content embedding of chunk $N_i C_j$
G_i	The global position embedding of each chunk in note N_i
L_j	The local position embedding of j-th chunk in each note
$R_j^{N_i}$	The position-enhanced chunk embedding of chunk $N_i C_j$
$\Delta \overrightarrow{t}_j^{N_i}$	The time interval between the current chunk $N_i C_j$ and the next chunk, which could be either $N_i C_{j+1}$ or $N_{i+1} C_1$
$\Delta \overleftarrow{t}_j^{N_i}$	The time interval between the current chunk $N_i C_j$ and the previous chunk, which could be either $N_i C_{j-1}$ or $N_{i-1} C_{o_{i-1}}$
y	The label of the patient
$\hat{y}$	The prediction of y
where $i = 1, ..., m$, $j = 1, ..., o_i$ and $l = 1, ..., p$	

4.2. FTL-Trans Overview

We propose a hierarchical model structure to capture the temporal and multi-level sequential information within clinical notes. Our proposed model extracts a single, patient-level

representation from temporal patient notes which could be used to predict clinical outcomes, such as mortality or readmission prediction. Figure 2 shows the architecture of the proposed model framework, which we call **FTL-Trans**. FTL-Trans is composed of four layers. The first layer is the Chunk Content Embedding Layer (Section 4.3), which reads the tokens within each equal-length subsequence of a patient's clinical notes ("chunk") and encodes the linguistic information into a vector. The second layer is the Position-Enhanced Chunk Embedding Layer (Section 4.4), which is designed for combining the chunk content embeddings with multi-level positional information of notes and chunks. This way, our model is able to account for the sequential information of notes and chunks. The third layer is the Time-Aware Layer (Section 4.5), which incorporates both the position-enhanced chunk embedding and time information to learn a patient-level representation of the sequence of clinical notes for downstream tasks. The last layer is the Classification Layer (Section 4.6) for making clinical predictions based on the learned patient-level representations. The following sections describe the details of each layer.

4.3. Chunk Content Embedding Layer

Chunk Content Embedding Layer is designed to encode the text within each chunk. To model the textual content of each chunk, we use a ClinicalBERT layer (Huang et al., 2019), which is a pre-trained BERT model using a medical corpus.

Since the BERT architecture has a maximum length requirement for input sequence, each of the patient's notes N_i is first split into sequences of o_i chunks $\{N_iC_1, N_iC_2, ..., N_iC_{o_i}\}$. Each chunk is composed of p tokens $N_iC_j = \{w_1^{N_iC_j}, w_2^{N_iC_j}, ..., w_p^{N_iC_j}\}$. Following the common practice in BERT-based architectures, the first token of a chunk is the special token '[CLS]'.

Tokens within each chunk are then fed into a transformer-encoder layer. We initialize this layer with the pre-trained ClinicalBERT model (Huang et al., 2019). The output of the ClinicalBERT layer is the chunk content embedding as follows:

$$E_j^{N_i} = ClinicalBERT(N_iC_j) \tag{2}$$

In ClinicalBERT, this chunk content embedding $E_j^{N_i}$ is directly used for generating prediction scores for each chunk:

$$P(label = 1|E_j^{N_i}) = \sigma(WE_j^{N_i}) \tag{3}$$

Where σ is the sigmoid function, and W is a parameter matrix. However, FTL-Trans instead learns to: 1) Differentiate between chunks from different notes with the help of Position-Enhanced Chunk Embedding Layer, and 2) Learns temporal importance with the help of Time-Aware Layer.

4.4. Position-Enhanced Chunk Embedding Layer

The Position-Enhanced Chunk Embedding Layer is used to merge each chunk content embedding and sequential information of both note and chunk into a single representation.

The position-enhanced chunk embedding is constructed from three sources: the chunk content embedding $E_j^{N_i}$, the position embedding of the notes G_i, which we name *Global*

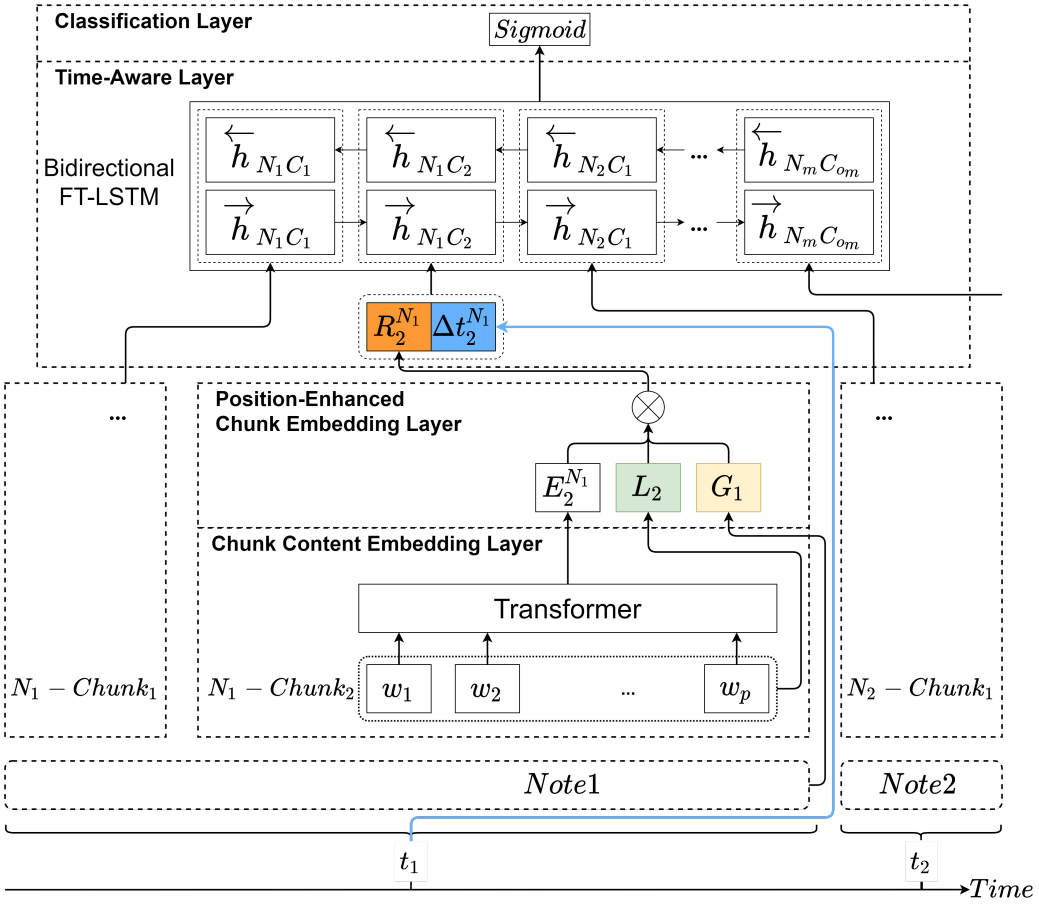

Figure 2: FTL-Trans is composed of four layers. The Chunk Content Embedding Layer learns an embedding that contains the semantic textual information in each chunk. Then the Position-Enhanced Chunk Embedding Layer merges this linguistic representation with position information of the note and chunk into a final representation of chunk. Next, the Time-Aware Layer utilizes both the chunk representation and the time information to learn a single representation for the sequence of clinical notes. The learned representation is fed into the Classification Layer for patient-level prediction.

Position embedding and the position embedding of chunks L_j, which is called *Local Position embedding*. G_i indicates the position of the note N_i in the sequence of notes $\mathcal{N}$. L_j indicates the position of the chunk N_iC_j in the note N_i. Both L_j and G_i are vectors with learned sets of parameters. We first concatenate $E_j^{N_i}$ with G_i and L_j then feed it into a single layer perceptron, followed by a layer normalization (Ba et al., 2016) and dropout (Srivastava et al., 2014) to get the position-enhanced chunk embedding $R_j^{N_i}$ as follows:

$$R_j^{N_i} = \text{Dropout}(\text{LayerNorm}(W_{ec} \cdot [E_j^{N_i}, G_i, L_j] + b_{ec})) \tag{4}$$

where W_{ec} and b_{ec} are trainable parameters shared across chunks.

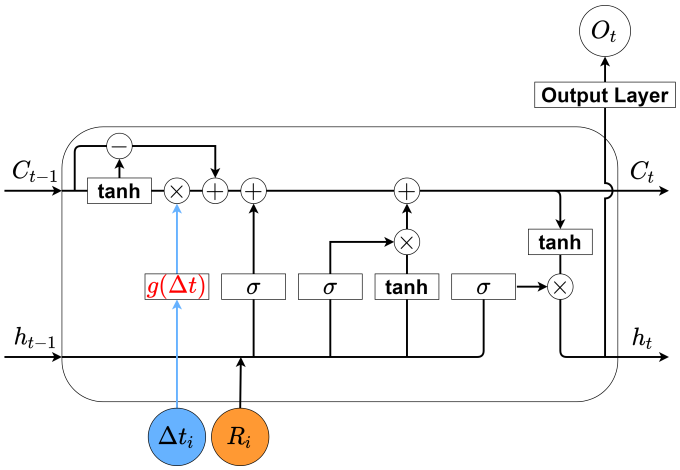

Figure 3: Flexible Time-aware Long Short Term Memory (FT-LSTM) takes the chunk representations R_i and the elapsed time at the current time step Δt_i as input. The elapsed time between consecutive events are irregular. In the FT-LSTM cell, the previous memory C_{t-1} is decomposed into long and short term memory and the short term memory is discounted by a flexible decay function $g(\Delta t)$.

4.5. Time-Aware Layer

The time-aware layer is designed for capturing the temporal information in the clinical note sequences to reflect the change of temporal importance of clinical events over time. One of the time-aware models is the time-aware LSTM (T-LSTM), which is proposed in Baytas et al. (2017). In this work, we propose the Flexible T-LSTM (FT-LSTM), which is an extension of the T-LSTM model. In FT-LSTM, the previous memory is decomposed into long-term and short-term components. Then the short-term memory will be discounted by a time decay factor computed by the time between successive elements. Finally, the discounted short-term memory and long-term memory will be combined to get new memory. T-LSTM uses a non-increasing function of the elapsed time which transforms the time interval into a weight assigned to short-term memory content using $g(\Delta t) = \frac{1}{\Delta t}$ or $g(\Delta t) = \frac{1}{\log(e+\Delta t)}$. These functions do not have any trainable parameters. Hence their assumption is that the temporal influence will always decay in a fixed mode. However, this assumption may not be true in some cases, especially within the clinical domain. Instead of using a non-increasing function, we propose a flexible and universal decay function, as shown in Figure 3, which is inspired by Su et al. (2018):

$$g(\Delta t) = \frac{q_1}{a \cdot \Delta t^b} + q_2(c \cdot \Delta t + d) + \frac{q_3}{1 + (\frac{\Delta t}{f})^g} \tag{5}$$

where q_i are the weights of the three sub-functions $\frac{1}{a \cdot \Delta t^b}$, $c \cdot \Delta t + d$, and $\frac{1}{1+(\frac{\Delta t}{f})^g}$, denoting the three possible shapes of decay function: convex, linear and concave respectively. All parameters q_i, a, b, c, d, f, g are trainable. During the training procedure, a flexible decay function will be learned by combining the three sub-functions with adjustable weights.

We use the bidirectional FT-LSTM to capture the temporal influence in two directions. The bidirectional FT-LSTM is composed of forward cells $\overrightarrow{h}$ and backward cells $\overleftarrow{h}$. We train this FT-LSTM by inputting the position-enhanced chunk embedding $R_j^{N_i}$ and the time interval $\Delta t_j^{N_i}$. The time intervals are comprised of the $\Delta \overrightarrow{t}_j^{N_i}$ and $\Delta \overleftarrow{t}_j^{N_i}$. $\Delta \overrightarrow{t}_j^{N_i}$ denotes the time interval between the current chunk N_iC_j and the next chunk, which could be either N_iC_{j+1} or $N_{i+1}C_1$. $\Delta \overleftarrow{t}_j^{N_i}$ denotes the time interval between the current chunk N_iC_j and the previous chunk, which could be either N_iC_{j-1} or $N_{i-1}C_{o_{i-1}}$. We compute these two time intervals as follows:

$$\Delta \overrightarrow{t}_j^{N_i} = \begin{cases} 0 & j \neq o_i \text{ or } i = m \\ t_{i+1} - t_i & \text{otherwise} \end{cases} \tag{6}$$

$$\Delta \overleftarrow{t}_j^{N_i} = \begin{cases} 0 & j \neq 1 \text{ or } i = 1 \\ t_i - t_{i-1} & \text{otherwise} \end{cases} \tag{7}$$

where o_i is the number of chunks in N_i.

The elapsed time $\Delta \overrightarrow{t}_j^{N_i}$ are fed into $\overrightarrow{h}$ cell, and the elapsed time $\Delta \overleftarrow{t}_j^{N_i}$ are fed into $\overleftarrow{h}$ cell. As shown in Figure 3, FT-LSTM first decomposes the memory from last cell into short-term memory $C_{t-1}^S = \tanh(W_d C_{t-1} + b_d)$ and long-term memory $C_{t-1}^L = C_{t-1} - C_{t-1}^S$, where W_d, b_d are the parameters for decomposition, which is learned during the model training procedure. Using the decay function $g(\Delta t)$, FT-LSTM discounts C_{t-1}^S to get the discounted short-term memory $\hat{C}_{t-1}^S = C_{t-1}^S * g(\Delta t)$. Then $\hat{C}_{t-1}^S$ and C_{t-1}^L are combined to get the adjusted memory $C_{t-1}^* = C_{t-1}^L + \hat{C}_{t-1}^S$. After the adjusted memory C_{t-1}^* is obtained, the following computations for the forget gate, input gate, output gate, candidate memory, current memory, and current hidden state in the FT-LSTM cell are the same as in the standard LSTM.

4.6. Classification Layer

The final Classification Layer is for making patient-level predictions. It takes the last hidden state of FT-LSTM $\tilde{h}$, the concatenation of last forward cell $\overrightarrow{h}$ and backward cell $\overleftarrow{h}$, as the input. $\tilde{h}$ is fed into a dropout layer and a single layer perceptron, followed by a sigmoid function. The output is a prediction score within $[0, 1]$, as shown below:

$$\hat{y} = \sigma(W_c \text{Dropout}(\tilde{h}) + b_c) \tag{8}$$

where W_c and b_c are trainable parameters.

We use the cross-entropy loss to train our model:

$$L = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \tag{9}$$

where $\hat{y}$ is the prediction and y is the true label.

5. Experiments

5.1. Compared Methods

To evaluate the performance of FTL-Trans, we work with five prediction tasks using cohorts extracted from MIMIC-III datasets. Our tasks are in-hospital mortality prediction, 30-day readmission prediction, Escherichia Coli infection prediction, Enterococcus Sp. infection prediction, and Klebsiella Pneumoniae infection prediction. The details of cohort extraction can be found in Section 3.

We compare our proposed FTL-Trans model with the following state-of-the-art alternative approaches:

- **BERT:** The original BERT model (Devlin et al., 2018) that is pre-trained on BookCorpus (Zhu et al., 2015) and English Wikipedia. The sequence of notes is first split into n chunks $\{C_1, C_2, ...C_n\}$. The BERT model predicts a probability P_i for each chunk, denoting the predicted positive probability score for chunk C_i. We then compute a patient-level prediction score $P_{\text{pt-mean}} = \frac{\sum P_i}{n}$.

- **ClinicalBERT Simple Mean (ClBERT-sm):** An application of BERT is pre-trained on MIMIC-III dataset (Huang et al., 2019). ClinicalBERT model predicts the positive probability for each chunk, and we report the simple mean of chunk prediction scores $P_{\text{pt-mean}}$, computed in the same way as we outlined for the BERT model.

- **ClinicalBERT Adjusted Mean (ClBERT-am):** Huang et al. (2019) designs a method to compute the patient-level output probability based on chunk-level probabilities, which is $P_{\text{pt-mean}} = \frac{P_{max}+P_{mean}n/c}{1+n/c}$. Here, n is the number of chunks split from the patient's notes. C is a scaling factor to control the influence of n. P_{max} and P_{mean} are the max and mean values of predicted probability scores across the n chunks, respectively.

These three models utilize a flattened representation of chunks, thus, they ignore the sequential and temporal information in the sequence of clinical notes.

Next, we design three variations of our proposed method, all of which take the multi-level sequential information into account yet in alternative ways.

- **LSTM + Transformer (L-Trans):** A variation of FTL-Trans model in which we replace the bidirectional FT-LSTM with a traditional bidirectional LSTM. This model does not capture the time information of notes.

- **Patient-level Transformer + Transformer (P-Trans):** A variation of FTL-Trans model in which we replace the bidirectional FT-LSTM with a shallow BERT model with a single transformer encoder block layer. There is no mechanism to capture the time information in this model.

- **T-LSTM + Transformer (TL-Trans):** A variation of FTL-Trans model in which we replace the bidirectional FT-LSTM with bidirectional T-LSTM (Baytas et al., 2017) in the time-aware layer. The time decay function in this model is $g(\Delta t) = \frac{1}{\log(e+\Delta t)}$, which is a fixed function without any trainable parameters.

We speculate that FTL-Trans should outperform all of these variations as it has the most flexible temporal importance mechanism (i.e., FT-LSTM layer).

5.2. Results

5.2.1. Performance Comparison

We use the following popular metrics, namely, Area Under the Receiver Operating Characteristic curve (AUROC), Accuracy, and Area Under Precision-Recall curve (AUPR), for the evaluation study. We split each cohort into train, validation, and test sets, with a ratio of $8:1:1$. We provide the details of the implementation of our models in the Appendix.

Table 3 reports the results for the five prediction tasks. FTL-Trans outperforms alternative methods in almost all of the tasks. For AUROC and Accuracy, FTL-Trans has the best performance across all five tasks. The difference between BERT and both ClBERT-sm and ClBERT-am suggests that models applied to the clinical domain perform best when they are pre-trained on a medical corpus. The difference between ClBERT-sm and ClBERT-am is not significant across the five cohorts. It indicates that the design of the aggregation method in flat models may not play a key role.

The mortality and readmission cohorts have more notes and a larger number of patients than other cohorts. The performance of the flat models in these two cohorts is worse than the performance of the hierarchical models. This illustrates the necessity of using the multi-level sequential information inherent in clinical notes. We also note that the advantage of hierarchical models is likely not going to be significant in smaller cohorts. The shortage of data might cause the contribution from sequential information to become less important.

Our model, FTL-Trans, shows advantages over flat models most of the time, especially in the AUROC and accuracy metrics. This demonstrates that the combination of exploiting both sequential information and temporal information can make a steady contribution to the model's performance. By comparing the performance of L-Trans, TL-Trans, and FTL-Trans, we see the usage of temporal information for clinical prediction, and, more importantly, the design of the time decay function have a strong impact on the outcome of the prediction task. The changing trend of temporal importance is task-dependent. A flexible time decay function with trainable parameters in FT-LSTM can better utilize the temporal information compared to the fixed time decay function in T-LSTM. On the contrary, a fixed decay function may worsen the performance of the model.

5.2.2. Effectiveness of Global and Local Position Embeddings

We also study the performance improvement triggered by the inclusion of the global and local position embeddings, respectively. For this, we create four variations of the FTL-Trans model where we allow a varying leves of position-wise information to propagate within the network. Our baseline for these experiments does not incorporate any positional information from chunks nor from notes. *Global position embedding model* utilizes positional information from only notes, whereas *Local position embedding model* utilizes positional information from only chunks. The *Multi-level Position Embedding model* corresponds to the FTL-Trans model, where we employ both global and local position embeddings.

Table 4 reports the experimental results for these four model variations. The results confirm that the use of global and local position embeddings does indeed help to improve the

Table 3: Performance Comparison of baselines and the proposed model

Method Type	Model	**Mortality**		
		AUROC	Accuracy	AUPR
Flat Models	BERT	88.02 ± 1.77	79.88 ± 2.21	87.69 ± 1.78
	ClBERT-sm	90.52 ± 1.27	82.26 ± 0.98	90.63 ± 1.55
	ClBERT-am	90.27 ± 1.16	82.25 ± 1.08	90.26 ± 1.54
Hierarchical models without time info	L-Trans	94.39 ± 0.65	87.68 ± 1.10	94.39 ± 0.92
	P-Trans	94.13 ± 0.85	86.58 ± 1.73	94.08 ± 0.49
Hierarchical models with time info	TL-Trans	93.78 ± 0.96	86.06 ± 0.58	93.78 ± 1.08
	FTL-Trans	**95.00 ± 0.86**	**88.17 ± 1.05**	**95.02 ± 0.82**
Method Type	Model	**Readmission**		
		AUROC	Accuracy	AUPR
Flat Models	BERT	65.97 ± 2.07	56.87 ± 2.97	63.32 ± 2.60
	ClBERT-sm	72.87 ± 2.15	65.03 ± 2.51	71.64 ± 2.59
	ClBERT-am	72.85 ± 2.10	65.79 ± 1.84	71.67 ± 2.51
Hierarchical models without time info	L-Trans	76.00 ± 2.16	68.36 ± 1.16	74.15 ± 4.01
	P-Trans	75.23 ± 1.47	67.43 ± 0.51	**74.36 ± 3.60**
Hierarchical models with time info	TL-Trans	73.31 ± 1.36	67.74 ± 2.27	71.36 ± 2.98
	FTL-Trans	**76.74 ± 2.40**	**70.61 ± 2.19**	74.30 ± 3.61
Method Type	Model	**Escherichia Coli**		
		AUROC	Accuracy	AUPR
Flat Models	BERT	70.07 ± 1.15	62.79 ± 2.37	70.56 ± 0.97
	ClBERT-sm	71.09 ± 1.79	64.37 ± 2.46	70.55 ± 1.39
	ClBERT-am	71.50 ± 1.68	64.20 ± 2.01	71.29 ± 1.29
Hierarchical models without time info	L-Trans	72.26 ± 0.97	67.15 ± 1.81	66.84 ± 2.05
	P-Trans	71.33 ± 1.37	64.64 ± 1.88	71.41 ± 0.98
Hierarchical models with time info	TL-Trans	72.00 ± 3.10	64.42 ± 3.85	69.38 ± 2.23
	FTL-Trans	**74.88 ± 2.99**	**68.02 ± 3.20**	**72.41 ± 1.81**
Method Type	Model	**Enterococcus Sp.**		
		AUROC	Accuracy	AUPR
Flat Models	BERT	72.47 ± 2.60	66.05 ± 2.49	70.63 ± 1.19
	ClBERT-sm	74.63 ± 2.19	67.86 ± 2.75	71.72 ± 2.67
	ClBERT-am	74.44 ± 1.71	67.46 ± 2.40	71.36 ± 2.36
Hierarchical models without time info	L-Trans	73.45 ± 2.33	65.83 ± 1.14	68.94 ± 3.71
	P-Trans	74.09 ± 1.64	65.62 ± 1.42	71.37 ± 2.26
Hierarchical models with time info	TL-Trans	73.10 ± 2.94	66.05 ± 0.95	69.39 ± 4.51
	FTL-Trans	**76.47 ± 2.25**	**69.35 ± 1.99**	**73.43 ± 3.14**
Method Type	Model	**K. pneumoniae**		
		AUROC	Accuracy	AUPR
Flat Models	BERT	67.85 ± 2.12	60.87 ± 1.74	66.98 ± 2.48
	ClBERT-sm	69.05 ± 3.95	62.86 ± 4.74	67.65 ± 3.33
	ClBERT-am	68.23 ± 3.61	60.95 ± 4.75	67.27 ± 2.99
Hierarchical models without time info	L-Trans	71.49 ± 0.66	64.84 ± 1.69	68.13 ± 2.63
	P-Trans	70.49 ± 5.10	63.49 ± 5.51	**71.90 ± 4.93**
Hierarchical models with time info	TL-Trans	68.95 ± 2.47	63.33 ± 1.83	65.72 ± 4.48
	FTL-Trans	**73.20 ± 1.80**	**66.19 ± 2.15**	69.71 ± 2.43

Table 4: Effect of Multi-level Position Embeddings in FTL-Trans

Model	Mortality		
	AUROC	Accuracy	AUPR
No Position Embedding	94.49 ± 0.65	86.93 ± 0.84	94.48 ± 0.76
Global Position Embedding	94.35 ± 0.82	86.64 ± 1.02	94.37 ± 0.75
Local Position Embedding	94.23 ± 1.01	86.03 ± 1.73	94.21 ± 1.15
Multi-level Position Embedding	**95.00 ± 0.86**	**88.17 ± 1.05**	**95.02 ± 0.82**

Model	Readmission		
	AUROC	Accuracy	AUPR
No Position Embedding	74.53 ± 1.60	68.41 ± 1.64	71.94 ± 2.83
Global Position Embedding	76.62 ± 2.42	70.95 ± 2.53	74.09 ± 3.62
Local Position Embedding	76.63 ± 2.40	**71.00 ± 2.26**	74.06 ± 3.63
Multi-level Position Embedding	**76.74 ± 2.40**	70.61 ± 2.19	**74.30 ± 3.61**

Model	Escherichia Coli		
	AUROC	Accuracy	AUPR
No Position Embedding	74.53 ± 1.60	**68.43 ± 3.56**	70.86 ± 2.74
Global Position Embedding	75.02 ± 2.88	67.68 ± 2.98	**73.42 ± 2.23**
Local Position Embedding	**75.07 ± 2.89**	68.08 ± 2.59	73.41 ± 2.15
Multi-level Position Embedding	74.88 ± 2.99	68.08 ± 3.20	72.41 ± 1.81

Model	Enterococcus Sp.		
	AUROC	Accuracy	AUPR
No Position Embedding	75.61 ± 1.84	67.36 ± 1.83	71.43 ± 3.71
Global Position Embedding	76.40 ± 2.02	68.91 ± 1.37	73.21 ± 3.73
Local Position Embedding	76.29 ± 2.01	68.95 ± 1.74	73.07 ± 3.66
Multi-level Position Embedding	**76.47 ± 2.25**	**69.35 ± 1.99**	**73.43 ± 3.14**

Model	K. pneumoniae		
	AUROC	Accuracy	AUPR
No Position Embedding	71.86 ± 3.13	65.32 ± 2.18	67.83 ± 2.98
Global Position Embedding	73.15 ± 2.23	66.51 ± 2.10	69.16 ± 1.89
Local Position Embedding	73.20 ± 2.26	**66.75 ± 2.65**	68.93 ± 1.93
Multi-level Position Embedding	**73.20 ± 1.80**	66.19 ± 2.15	**69.71 ± 2.43**

model's performance. We also notice that in the mortality cohort, the multi-level position embedding has a significant advantage over other models. However, in cohorts other than the mortality, sometimes the model with only global or only local position embeddings performs better than the model with multi-level position embedding in a few of the metrics.

The most important difference between the mortality cohort and other cohorts is that the mortality cohort has much more data than others. The mortality cohort has 10,574 patients and 381,718 notes, while the second-largest data set, the readmission cohort, contains

5,920 patients and 190,004 notes. This indicates that the shortage of data might cause the advantage of multi-level position embedding to become less significant.

Another interesting finding is that the ratio of the average number of words per note to the average number of notes per patient in the mortality dataset is the lowest across the five cohorts. We know that a note with more words will be split into more chunks. We speculate that this may have strengthening the influence of global and local position embeddings on the overall model performance for the mortality cohort.

6. Discussion

A patient's clinical notes correspond to a sequence of documents generated during the patient's stay in a hospital. The abundant information in clinical notes can be leveraged for supporting clinical prediction. However, the multi-level sequential and temporal information within clinical notes, along with the interrelationships of these notes, were largely ignored or at least not fully utilized by previous works.

In this work, we instead propose a novel hierarchical structure based on the transformer-encoder FTL-Trans model which takes the interrelationships among clinical notes and the multi-level sequential information into account. Moreover, a novel flexible time-aware layer in FTL-Trans is incorporated that is capable of learning the relevancy of timing among notes in a series that may betackles irregularly spaced. FTL-Trans utilizes a trainable time decay function in the time-aware layer to assign a decay rate to the previous event. This mimics the temporal influence of the previous event on clinical outcomes.

We evaluate our approach on five clinical prediction tasks, namely, in-hospital mortality prediction, 30-days readmission prediction, and three infection prediction tasks. Our evaluation results demonstrate that our model outperforms strong baselines. We conclude that utilizing the multi-level sequential information and the interrelationship among clinical notes consistently and in some cases significantly improve the prediction performance. Also, the flexible design of the time decay function has been shown to be beneficial for reaching better performance.

6.1. Limitations

One potential limitation of our approach is that some of our evaluation cohorts include a limited number of patients and clinical notes. For example, K. Pneumoniae cohort contains 2,092 patients and a total of 61,164 notes. These smaller cohort sizes are typical for some medical outcomes and rare conditions. Yet, our evaluation suggests that our proposed method tends to achieve a more significant performance improvement whenever more data is made available for training. Thus we anticipate that with the availability of additional data additional experiments may confirm a further improvement of our model compared to state-of-the-art solutions.

Another limitation currently are our data sources. While we have worked with colleagues at UMASS Medical Center, medical data can be challenging to get access to due to the privacy concerns for patients. The clinical notes in MIMIC-III are all from a single hospital, the Beth Israel Deaconess Medical Center in Boston, MA. Using data from multiple healthcare institutions may lead to better performance and model training. Finally, in the MIMIC-III dataset, all the Protected Health Information (PHI) was removed and

replaced by "PHI" symbols. Some other datasets replace PHI with synthetic, but realistic and consistent identification. The text distributions in these two types of data could have significant differences. This all indicates that FTL-Trans cannot be directly applied to the dataset with synthetic or real identification without pre-training and fine-tuning.

6.2. Future Work

In the future, the hierarchical structure in our proposed FTL-Trans model could be expanded even further. In our current work, for simplicity, we use a single bidirectional FT-LSTM in the Time-aware Layer to feed chunks across all notes. In future work, we could add another layer of the FT-LSTM and then first feed the chunks from the same note into the first layer to generate the note representation. Then each note representation could be fed into the second layer of FT-LSTM to generate the patient representation. Expanding our model could reach better prediction performance.

In this work, we focused on studying whether our model design indeed incorporates temporal and sequential information well. Hence, we implement experiments on balanced data to avoid some of the other possible challenges. However, for some rare diseases, there may not be many cases available and thus usable for model training. Another future direction to consider could be to use unbalanced data for model training to assess the effectiveness of our proposed technology for such practical data sets. Few-shot learning and prediction for those rare diseases could also be explored.

Acknowledgments

We would like to thank the Data Science Research Group at Worcester Polytechnic Institute, the anonymous reviewers, and meta-reviewers for their feedback on the paper.

This work is supported in part by the NSF Division Information and Intelligent Systems (award 1852498 and 1815866).

References

Emily Alsentzer, John R Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. Publicly available clinical bert embeddings. *arXiv preprint arXiv:1904.03323*, 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Tian Bai, Shanshan Zhang, Brian L Egleston, and Slobodan Vucetic. Interpretable representation learning for healthcare via capturing disease progression through time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 43–51, 2018.

Inci M Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K Jain, and Jiayu Zhou. Patient subtyping via time-aware lstm networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 65–74. ACM, 2017.

Willie Boag, Dustin Doss, Tristan Naumann, and Peter Szolovits. What's in a note? unpacking predictive value in clinical note representations. *AMIA Summits on Translational Science Proceedings*, 2018:26, 2018.

Po-Chun Chen, Ta-Chung Chi, Shang-Yu Su, and Yun-Nung Chen. Dynamic time-aware attention to speaker roles and contexts for spoken language understanding. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 554–560. IEEE, 2017.

Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, pages 3504–3512, 2016.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Sebastien Dubois, Nathanael Romano, David C Kale, Nigam Shah, and Kenneth Jung. Learning effective representations from clinical notes. *stat*, 1050:15, 2017.

Paulina Grnarova, Florian Schmidt, Stephanie L Hyland, and Carsten Eickhoff. Neural document embeddings for intensive care patient mortality prediction. *arXiv preprint arXiv:1612.00467*, 2016.

Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. Clinicalbert: Modeling clinical notes and predicting hospital readmission. *arXiv preprint arXiv:1904.05342*, 2019.

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

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1269–1278, 2019.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Li-wei Lehman, Mohammed Saeed, William Long, Joon Lee, and Roger Mark. Risk stratification of icu patients using topic models inferred from unstructured progress notes. In *AMIA annual symposium proceedings*, volume 2012, page 505. American Medical Informatics Association, 2012.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

Cansu Sen, Thomas Hartvigsen, Elke Rundensteiner, and Kajal Claypool. Crest-risk prediction for clostridium difficile infection using multimodal data mining. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–63. Springer, 2017.

Cansu Sen, Thomas Hartvigsen, Xiangnan Kong, and Elke Rundensteiner. Patient-level classification on clinical note sequences guided by attributed hierarchical attention. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 930–939. IEEE, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

Shang-Yu Su, Pei-Chieh Yuan, and Yun-Nung Chen. How time matters: Learning time-decay attention for contextual spoken language understanding in dialogues. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2133–2142, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Jenna Wiens, Eric Horvitz, and John V Guttag. Patient risk stratification for hospital-associated c. diff as a time-series classification task. In *Advances in Neural Information Processing Systems*, pages 467–475, 2012.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764, 2019.

Yuan Zhang, Xi Yang, Julie Ivy, and Min Chi. Attain: attention-based time-aware lstm networks for disease progression modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 10–16, 2019.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.

Appendix A.

We ran our experiments on a Tesla V100 GPU. To fit the data in the GPU memory (16GB), we use the last 64 chunks of each patient to train and evaluate the model. Earlier chunks are discarded to keep an equal number of chunks per patient. We use the BertAdam optimizer Wolf et al. (2019) with an initial learning rate of 2×10^{-5}, a warm-up proportion of 0.1. We implement our models in Pytorch. The pre-trained BERT model that we used is from the "pytorch-transformers" library (renamed as "Transformers" since V2.0.0), which is available at https://github.com/huggingface/transformers. The pre-trained ClinicalBERT model and code are loaded from https://github.com/kexinhuang12345/clinicalBERT. Each model is trained for 3 epochs. For every task, We report average evaluation results of each model from 5 random initialization. Training time varies depending on the cohort size. The average training time of FTL-Trans on the mortality cohort is 1.56 hours/epoch.

For the flexible time decay function $g(\Delta(t))$, we conducted a hyper-parameter search for parameters in it. Optimum initial values found are $a = 1$, $b = 1$, $c = 2.9$, $d = 0.02$, $f = 4.5$, $g = 2.5$, $q1 = q2 = q3 = 0.33$. We also have a constraint on $g(\Delta(t))$ to make sure $g(\Delta(t))$ is within $[0, 1]$. For the other parameters in FT-LSTM, we initialized them using a normal distribution with mean = 0 and standard deviation = 0.02.