LCD Benchmark: Long Clinical Document Benchmark on Mortality Prediction

WonJin Yoon, PhD [1,2], Shan Chen, MS [1,2,3,4], Yanjun Gao, PhD[5], Dmitriy Dligach, PhD[6], Danielle S. Bitterman, MD[1,2,3,4], Majid Afshar, MD MSCR[5], Timothy Miller, PhD[1,2,*]

1. Computational Health Informatics Program, Boston Children's Hospital, MA, USA
2. Department of Pediatrics, Harvard Medical School, MA, USA
3. Artificial Intelligence in Medicine (AIM) Program, Mass General Brigham, Harvard Medical School, Boston, MA, USA
4. Department of Radiation Oncology, Brigham and Women's Hospital/Dana-Farber Cancer Institute, Boston, MA, USA
5. Department of Medicine, University of Wisconsin - Madison, Madison, WI, USA
6. Loyola University Chicago. Department of Computer Science. Chicago, IL, USA

* Corresponding Author. Timothy.Miller@childrens.harvard.edu

Abstract

Natural Language Processing (NLP) is a study of automated processing of text data. Application of NLP in the clinical domain is important due to the rich unstructured information implanted in clinical documents, which often remains inaccessible in structured data. Empowered by the recent advance of language models (LMs), there is a growing interest in their application within the clinical domain. When applying NLP methods to a certain domain, the role of benchmark datasets are crucial as benchmark datasets not only guide the selection of best-performing models but also enable assessing of the reliability of the generated outputs. Despite the recent availability of LMs capable of longer context, benchmark datasets targeting long clinical document classification tasks are absent. To address this issue, we propose LCD benchmark, a benchmark for the task of predicting 30-day out-of-hospital mortality using discharge notes of MIMIC-IV and statewide death data. Our notes have a median word count of 1687 and an interquartile range of 1308 to 2169. We evaluated this benchmark dataset using baseline models, from bag-of-words and CNN to Hierarchical Transformer and an open-source instruction-tuned large language model. Additionally, we provide a comprehensive analysis of the model outputs, including manual review and visualization of model weights, to offer insights into their predictive capabilities and limitations. We expect LCD benchmarks to become a resource for the development of advanced supervised models, prompting methods, or the foundation models themselves, tailored for clinical text.
The benchmark dataset is available at
https://github.com/Machine-Learning-for-Medical-Language/long-clinical-doc

INTRODUCTION

Clinical notes describe and communicate events or interactions about a patient, written by healthcare providers[1]. These notes are rich sources of information important for clinical decision-making, often containing details that may not be readily available in a structured format. Clinical Natural Language Processing (NLP) can extract information from unstructured data[2]. This capability can be used to build an end-to-end model for specific tasks or to supplement structured data that can be used for further use in ML/AI models[3–7]. With the recent emergence of transformer-based Language Models (LMs), research on clinical NLP has achieved remarkable improvements. However, due to the architectural characteristics of transformer models, most available LMs have constraints on the maximum length of the input sequence that a model can process at once. In the clinical NLP domain, this can be a major technical hurdle for translational applications as the clinical notes can be longer than what most transformer models can process. For example, BERT and RoBERTa models can handle up to 512 tokens at one time, but the discharge summaries in MIMIC-IV have 1,600 words on average, which in token is about six times longer than the 512 token limit. These constraints raise the need for the development of the models capable of processing longer documents as well as long document benchmark datasets to test the ability of developed models.

A handful of methods have been introduced to utilize LMs in processing long documents. One simple method is to split input texts into smaller parts and merge the results after the predictions. For example, Devlin et al.[8] truncated input with a given window size and processed the remaining on subsequent processing steps for the Question Answering task. Alsentzer et al.[9] breakdown input documents to sentence-level inputs for Clinical NLP tasks. Chen et al.[10] sectionized long clinical documents and only kept informative sections for downstream NLP tasks. These methods are straightforward and easy to implement, but do not allow the LM to consider the entire context of original inputs. Another method is to increase the maximum length of the model itself. LongFormer[11] and Clinical-LongFormer[12] have expanded the maximum processing window by replacing full self-attention flow with a combination of local attention and task-oriented global attention. Another approach that falls into this branch is the large LMs trained on longer context windows. Pre-training or fine-tuning of these models requires significant computational resources that precludes access for many researchers and healthcare institutions. Existing work has therefore focused on existing models using prompt-based zero- or few-shot learning[13–15].

In this paper, we describe work in developing a benchmark for clinical long document processing models, based on the out-of-hospital mortality prediction task. The source of the dataset is MIMIC-IV v2.2 corpus, specifically discharge notes for patients who were admitted to the ICU and discharged to locations other than hospice facilities. Along with the benchmark dataset, we explore multiple machine learning models for the task, including traditional Support Vector Machine using Bag-of-Words, Convolutional Neural Networks, a hierarchical transformer encoder[16], and zero-shot large language models. In the results section, we select three baseline models, the best-performing CNN model, hierarchical transformer, and large language models (Mixtral-8x7B-instruct-v0.1) and analyze the outputs. Based on expert physician review, we

discovered that the dataset is challenging and at the same time the models can find meaningful signals. We additionally leverage the architecture of the hierarchical transformer model to analyze its behavior. In these analyses, we visualize and quantify the extent to which they jointly consider information from different sections of the discharge summary.

Contribution of this paper is two-fold. In clinical NLP perspective, we anticipate that the proposed dataset will serve as a solid foundation for model development and, moreover, as a forum for evaluating Large LMs on long clinical document classification tasks[1]. Second, the utilization of predictive models for 30-day mortality at the time of discharge is anticipated to facilitate timely end-of-life discussions with patients and their families. Such conversations are crucial for enhancing the quality of life for patients nearing the end of life, by ensuring that care decisions align with their values and preferences[17–20].

MATERIALS AND METHODS

Medical Information Mart for Intensive Care IV (MIMIC-IV)

The Medical Information Mart for Intensive Care (MIMIC) is a series of publicly available electronic health record (EHR) databases collected from Beth Israel Deaconess Medical Center (BIDMC)[21]. MIMIC databases contain multi-modal data such as text data, structured data (including laboratory data, admission records, and demographic data), and radiograph images for some versions. All the records and text data are de-identified.

MIMIC-IV[22] is the latest release that encompasses admissions between 2009 and 2019, focused on structured data and text data of ICU patients. We used MIMIC-IV v2.2 data[2] with discharge summaries and multiple structured records, including out-of-hospital mortality records from Massachusetts State Registry of Vital Records and Statistics[22].

Preprocessing

Preprocessing of our benchmark dataset is composed of three steps. First, following the criteria of Harutyunyan et al.[23], we collected admission records with an ICU stay. In the second step, we merged date of death data using the admission records identifier (*hadm_id*). In-hospital deaths were collected directly in the EHR of BIDMC or an affiliated institution, while out-of-hospital deaths were collected following a matching algorithm[22] with the Massachusetts State Registry of Vital Records and Statistics to incorporate the deaths in Massachusetts into MIMIC-IV. Dates of death were censored at one-year from the patient's last hospital discharge and null dates of death represent patients discharged and alive at one year after discharge. The third step filtered out records with task-specific restrictions. For our proposed 30-day

[1] The benchmark dataset is available at https://github.com/Machine-Learning-for-Medical-Language/long-clinical-doc and https://www.codabench.org/competitions/2064/
[2] Published: Jan. 6, 2023. https://physionet.org/content/mimiciv/2.2/

out-of-hospital mortality prediction dataset, we excluded admissions with in-hospital deaths and admissions where a patient had a discharge disposition of "***hospice***" in structured data because these patients are expected to die shortly after discharge. The training, validation, and testing datasets were partitioned according to patient ID to guarantee that all admissions from the same patient are allocated to the same dataset subset. For the note data, we only utilized discharge notes and not radiology reports. Full details are available in Appendix A, and python implementation of the exact algorithm is available on GitHub.

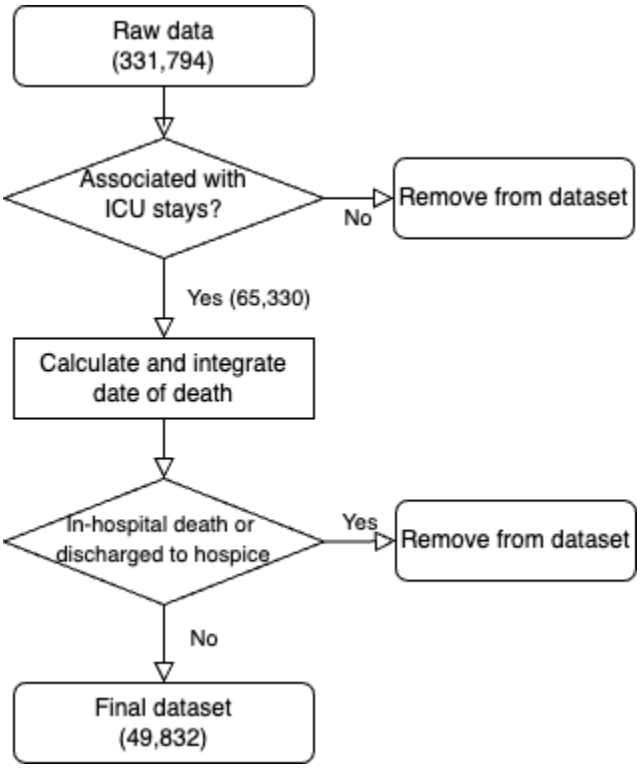

Figure 1 Diagram of the data preprocessing steps. The number of the notes is denoted in the parenthesis.

Table 1 Statistics of the datapoints during pre-processing steps. Admission-level data composes a minimum unit of data (often referred as example) and each admission has at most one discharge note. Some patients may have multiple admissions, resulting in several records. These patients might be represented in both the 'positive' and 'negative' notes categories (asterisked cells).

	# of admissions	# of notes	# of patients
Raw data	431,231	331,794	180,733
Data associated with ICU stays	65,330	65,330	50,253
Final dataset	49,832	49,832	39,705
-> Positive notes	1,830	1,830	1,772 *
-> Negative notes	48,002	48,002	38,408 *

Figure 1 shows the processing flow and Table 1 shows the number of datapoints after processing steps. As shown on the last two rows, our dataset is highly imbalanced; the negative-label notes, which means the patient survived, are about 26 times more abundant than positive-label notes. Of note, the number of admissions in the raw dataset exceeds the number of discharge notes because 99,437 admissions do not have discharge notes.

Figure 2 displays a histogram of the number of tokens in discharge notes. Each note is tokenized with *microsoft/xtremedistil-l6-h256-uncased* tokenizer and the Hugging Face Transformers library. As this model employs a word-piece tokenizer, a single word can be broken down into subwords and tokenized into multiple tokens depending on the frequency of the word. The median value for the token length were: 3978 (Interquartile range (IQR) 3085 - 5091) for train; 3991 (IQR 3080 - 5103) for development; and 3952 (IQR 3072 - 5072) for test set.

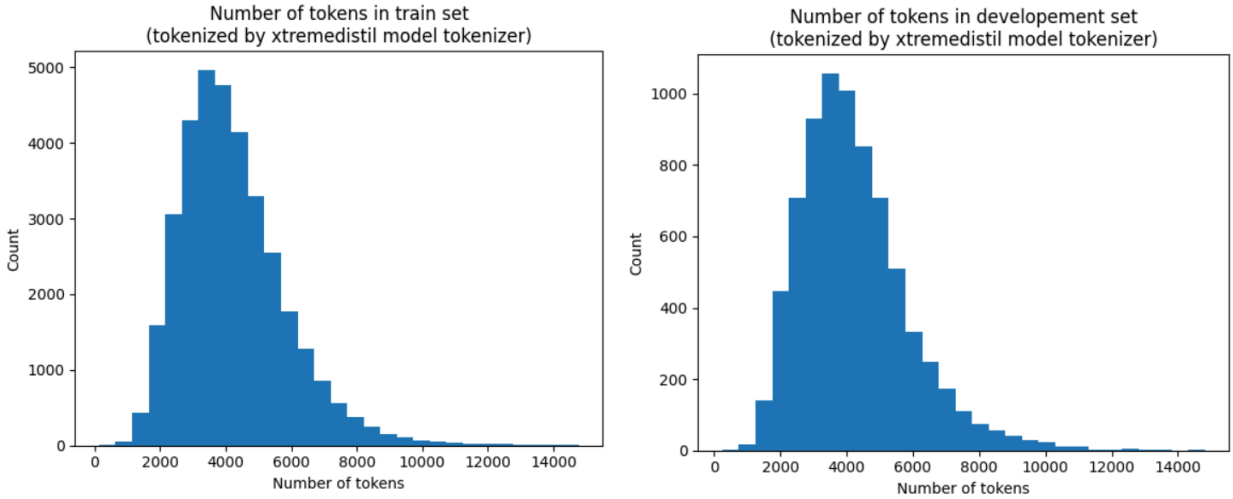

Figure 2 Histogram of the number of tokens in datapoints. Each note in datapoints is tokenized using microsoft/xtremedistil-l6-h256-uncased tokenizer and Huggingface Transformers library. Datapoints in sub-datasets are sorted into 30 bins. Longtail samples that have more than 15,000 tokens are excluded when plotting these graphs.

Baseline model

Bag-of-words (BoW) model: BoW model is a widely used baseline model for NLP where a given text sequence is represented with the frequency of words or word chunks in the sequence. BoW models learn vocabulary occurrence information but do not utilize the information of the order of word chunks in an input. Hence, they have a very limited ability to use syntactic information. The size of the word chunk, which could be one or a few words

depending on the window size, can add the ability to represent local syntactic information, but it can also make vocabularies sparse and very large. Despite these limitations, BoW is a strong baseline for document classification tasks with limited training dataset[24].
Bag-of-words models were implemented using scikit-learn[25] CountVectorizer module with monogram and bigram and *SGDClassifier* with default settings (hinge loss, max_iter=1000, tol=1e-3).

Convolutional Neural Networks

Kim et al.[26] proposed Convolutional Neural Network (CNN) as a feature extractor for the sentence classification task. Our CNN model followed the structure of Kim et al. with modification on embedding layer and hyperparameter settings: Kim et al. used word vectors pre-trained with continuous bag-of-words architecture namely *word2vec* (Mikolov et al., 2013), whereas our model used a randomly initialized embedding layer of 100 dimensions.

Pretrained transformer models

The transformer is a model architecture that relies on the self-attention mechanism, which is effective at capturing global dependencies within an input sequence. As the structure does not involve recurrence, transformer models can be efficiently trained using parallel computation units, such as GPUs, and this enables the training of much larger LMs. BERT and GPT models are some of the early proposed transformer LMs. These models are pre-trained on large-scale corpora and further finetuned to task-specific datasets for supervised learning. Empowered by the pretrained LMs, models tackling clinical NLP tasks have shown remarkable progress.

The self-attention mechanism of early transformer models is implemented by fully connecting each unit of sequence. This requires memory and computational costs that are quadratic with respect to the length of the input sequence, making it impossible to use transformer models for longer sequences.

Longformers: To mitigate this computational limitation for processing long documents, a handful of methods such as blend of local window and global attention approach and sliding window attention [11,12,27,28] have been proposed. Longformer[11] and Clinical-Longformer[12] are examples of such methods. Clinical-Longformer model was initialized from the pre-trained weights provided by the original authors[3] and fine-tuned on our dataset.

Hierarchical Transformers: Su et al.[16] introduced a hierarchical transformer, a stacked two transformer encoders of word-level encoder layer group and chunk-level encoder layer group. Hierarchical transformer splits input sequences into smaller chunks and first encoded chunks with word-level encoder to output chunk representations. The latter part of the structure, chunk-level encoder works as a feature extractor given the chunk representations of the former part and predicts classes for an input document (Figure 3). Hierarchical transformer models

[3] https://huggingface.co/yikuan8/Clinical-Longformer

were experimented with two settings, *xtremedistil* model and *PubMedBERT* model as initial weights for word-level encoder. The chunk-level encoder of the hierarchical transformer model was randomly initialized. Chunk size of hierarchical transformers were tested with two settings, 256 tokens and 512 tokens. In this paper we refer the letter setting as "*Bigchunk*" setting.

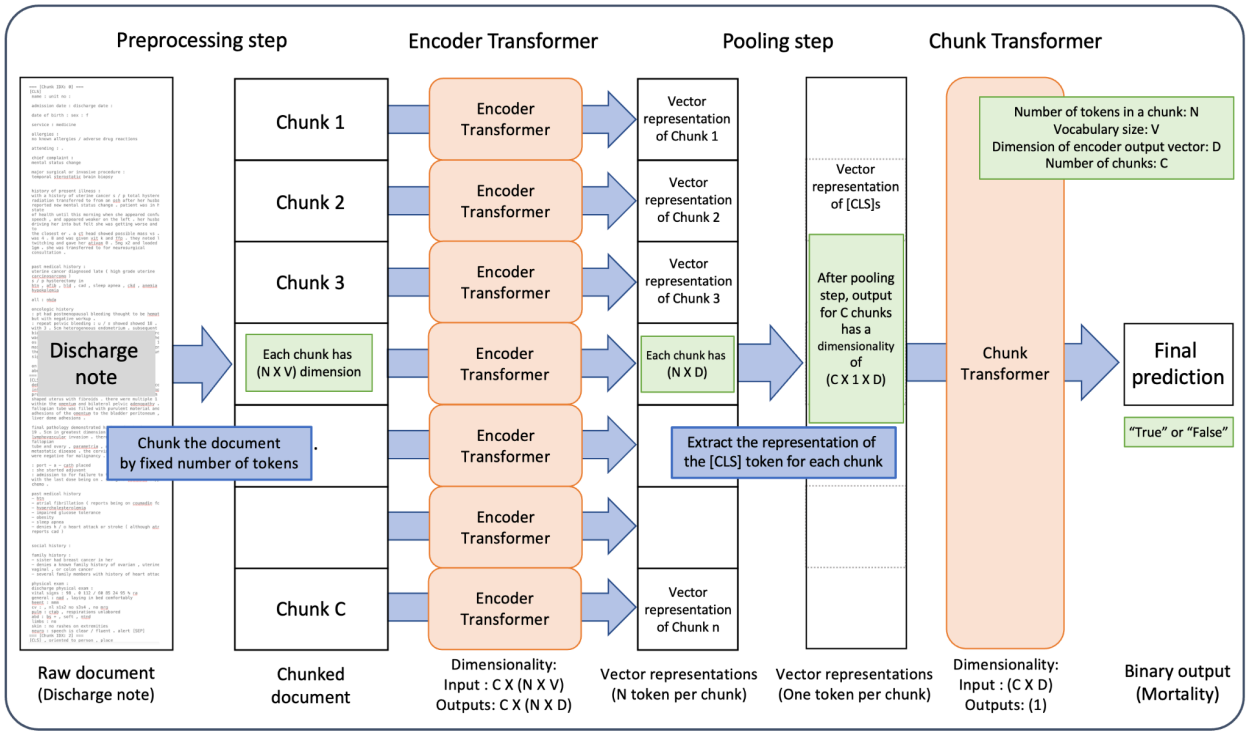

Figure 3 Structure of hierarchical transformer model. White boxes represent data and orange boxes represent transformer architecture. Green boxes represent dimensionalities of vectors for the step. All Encoder Transformers share weights. The figure shows [CLS] extraction as an example of the "pooling" methods.

Large LMs: We explored the ability of zero-shot mortality prediction using large LMs, Mixtral (8x7B)[4], and GPT4-32k (GPT-4). For the GPT4-32k, we used the HIPAA-compliant version that is provided through Mass General Brigham Azure version 0613. We experimented with Llama 2 models but were unable to find prompts that elicited results better than random guessing. For zero-shot experiments, we used the Hugging Face library to load and inference the Mixtral model and Azure API for the GPT-4 model. These models were selected as they are able to handle context with 32k tokens. For Mixtral, a 4-bit quantized [29] version was used. Figure 4 shows our *prompting template* for large LM models (Mixtral and GPT-4). The model is asked to choose the answer between *0:alive*, *1:death* and we used a regular expression that looks for the first incidence of "0" or "1" to extract answers from them.

[4] Mixtral-8x7B-Instruct-v0.1

```
<s>[INST] <<SYS>>
Below is a clinical document, please remember the following clinical context
and answer how likely is the given patient's out hospital mortality in 30 days?
<</SYS>>
Here is the clinical document:
<text>
$DISCHARGE_NOTE
</text> [INST] How likely is the given patient\'s out hospital mortality in 30
days? Please only use to answer with one word: 0:alive, 1:death [/INST]
```

Figure 4 Our prompt template for large LM experiments. $DISCHARGE_NOTE should be replaced by the real discharge notes.

Experiment details

Our primary metric is F-1 score for the positive labels and we used Receiver Operating Characteristic/Area Under the Curve (ROC AUC) as supplementary metric. Note that we used hinge loss for BoW models, which does not produce probability estimates for the calculation of ROC score. For BoW, CNN and Hierarchical Transformers, we experimented with 5 or 10 runs with identical settings except for the random seeds and averaged the performance to minimize the effect of random initialization of the model.

CNN, Hierarchical transformers, and Clinical-Longformer models were trained and tested on the CNLPT library[30] (available on GitHub[5]). Hyperparameter search for the BoW model was only performed on the n-gram window of the vectorizer, and we selected best performing settings based on experiments on the development dataset.
The models were evaluated against the dev set during the training time, and the best performing checkpoints were selected based on the average of Accuracy and the F-1 score.

CNN model and hierarchical transformer models have flexibility in selecting the maximum sequence length (*max_seq_length*), as unlike most language models, these models can expand the window without pre-training again from scratch. We selected *max_seq_length* to be 8192 tokens, which can cover 97% of the notes in the train and development set without truncation (based on *xtremedistil* model tokenizer).
Since the open-sourced Clinical-Longformer only supports maximum sequence length of 4096, we tested both right-truncation and left-truncation settings, i.e. truncating the ending part and the beginning part of the input sequence respectively.

Visualization of model attention

One of the benefits of the hierarchical transformer model is that it can provide a window into interpretability by highlighting the saliency of each input segment into the model prediction. This

[5] https://github.com/Machine-Learning-for-Medical-Language/cnlp_transformers/tree/dev-v0.7.0

becomes possible because the model splits the input into several chunks, each chunk is encoded through an encoder layer, and each encoded chunk representation works as an input unit of the chunk attention layers. By analyzing attention values and the vector norm[31] of each chunk, we can infer the model's prioritization of information across various chunks.

Early works attempting to understand the decision making process of transformer models have focused on explaining linguistic phenomena with attention weights [32–35]. However, multiple works argued that the attention weight based analysis is noisy and sometimes not explainable [31,36,37].

Kobayashi et al.[31] proposed vector norm based analysis, noting that the output vector of each attention layer is a weighted sum of vectors. Following the expression of Kobayashi et al., we denote vector representation of input unit, which is a chunk as we look into chunk-level encoder, at j-th position as x_j, and attention weight for j-th input to i-th output unit is denoted as $\alpha_{i,j}$. Then, the output vector (y_i) can be expressed as Equation (1) where a function $f(x)$ is a simplified notation of value transformation given input unit vector x.

$$y_i = \sum_{j=1}^{n} \alpha_{i,j} f(x_j) \quad (1)$$

As the equation explains, the output is affected by not only attention weights, $\alpha_{i,j}$, but also transformed input vector, $f(x)$. Norm-based analysis measures the norm of the weighted vector ($||\alpha f(x)||$) to figure out which input segments are highlighted for a given input sequence. Unlike machine translation tasks where this analysis is first presented, looking into input unit alignment (i.e. finding an input unit that resonates with another word) does not teach us meaningful insights. Rather, we focused on norm of output vector of attention layer, y_i , or $\left|\left|\sum_j a_{i,j} f(x_j)\right|\right|$, which will directly show the degree of importance of each input unit in the model's decision.

To investigate the importance of aggregating information across a discharge summary, we use the vector norm method to analyze section importance for this task. We do this by aggregating the two highest vector norm chunks for each instance in the test dataset. Since all inputs have different length, the content in a chunk with a given index can have a different meaning across each sample. Hence, instead of using chunk locations alone, we use section names from chunks for the analysis. The section names were extracted using a rule-based approach. If there are multiple sections in the chunks, for example n sections in the first chunk and k sections in the seconde chunk, all combinations of sections for an instance are included in the analysis with weights set as $\frac{1}{n\,k}$ to make the sum of weights for an instance as 1. This is to adjust the effect of short sections having a higher chance of being represented. For example, if one of the two important chunks contains section headers “Brief Hospital Course” and “Admission Diagnosis” and the second chunk contains the section header “Discharge Disposition”, the section pairs (“Brief Hospital Course”, “Discharge Disposition”) and

("Admission Diagnosis", "Discharge Disposition") each receive 0.5 counts for that instance. The partial counts are summed across the test dataset to see the population level analysis. Note that some longer sections such as **"*brief hospital course*"** can appear in multiple chunks.

Qualitative analysis

For the post-experiment exploratory analysis, we conduct two-step investigations. The first step is dictionary-based detection (i.e. exact match of synonyms list) of mentions about palliative and comfort care measures[6] and Do Not Resuscitate and Do Not Intubate (DNR/DNI) status. These mentions can be a strong signal for poor prognosis and can be a first filter for data investigation. The second step is to manually review the discharge notes for the left-over samples that do not have such terms. For the manual review, we provide notes, model predictions, true labels, and three questionnaires. Regarding model predictions, predicted binary labels and order of chunk highlights are provided. Labels are set to be hidden by default, and need to click unhide to see the labels. Three questionnaires were "Does this patient label seem valid?", "Was chunk information useful?", and "Was this case difficult to predict?"

For comparative analysis, we compare outputs of three models, CNN, Hierarchical Transformer, and Mixtral and manually inspect samples of the benchmark dataset. For this analysis, we focus on open-sourced models for this section as we have more control over the prediction process and the results of these models are more likely to be reproducible.

RESULTS

Table 2 shows our experimental results for the machine learning models.
BoW and CNN models showed strong performance against the transformer models: BoW showed 28.1% F1, CNN showed 28.9% F1. Among transformer-based models, hierarchical transformers showed the best performance, which is near the BoW or CNN models. *Bigchunk* model of Hierarchical Transformer models, which refers to chunk size of 512 tokens setting as opposed to normal 256 tokens, showed the best performance of 27.8% F1.
Clinical-Longformer showed lower performance when compared with BoW, CNN and hierarchical transformers models regardless of whether the text was truncated from the bottom (right truncated) or top (left truncated) of the document. Mixtral-8x7B-instruct-v0.1 model with zero-shot methods showed performance of 22.3% F1, which is 6% lower than the best performing supervised fine-tuning approach. Our results with GPT4 showed the best performance of 32.4% F1.

Table 3 shows the results of the aggregated chunk highlight pairs analysis. The table shows the section combinations and their frequencies, which shows the summation of section pair weights, normalized by the highest weight. Sections like "Brief Hospital Course" and "Pertinent Results" frequently are in the two most-attended sections.

---

[6] Comfort care term list: "hospice", "comfort measures", "comfort care", "palliative care"

Table 2 Performance of models in out-of-hospital mortality prediction task. Scores are evaluated based on positive labels. Bag-of-words models and large LMs (Mixtral, GPT-4) do not generate probability estimates that are necessary for calculating the ROC AUC score.

Model	# Params	Max Token	Method	P	R	F1	ROC AUC
BoW		-	Sparse vector	0.4230	0.2160	0.2810	NA
CNN	3 M	8,192	Training from scratch	0.4431	0.2188	0.2899	0.8468
Hierarchical Transformers (xdistill)	31 M	8,192	Fine-tuning	0.2519	0.2567	0.2526	0.8023
Hierarchical Transformers (xdistill - bigchunk)	31 M	8,192	Fine-tuning	0.3341	0.2401	0.2788	0.838
Hierarchical Transformers (PubMedBERT)	135 M	8,192	Fine-tuning	0.4496	0.1800	0.2566	0.8602
Hierarchical Transformers (PubMedBERT - bigchunk)	135 M	8,192	Fine-tuning	0.4478	0.1698	0.2418	0.8653
Clinical Longformer (Right truncate)	149 M	4,096	Fine-tuning	0.1923	0.134	0.158	0.7362
Clinical Longformer (Left truncate)	149 M	4,096	Fine-tuning	0.1237	0.0919	0.1054	0.6828
Mixtral (8x7b)	45 B	8,192	Zero-shot	0.1586	0.3755	0.2230	NA
GPT-4 (window - 32k)	Unknown	- (32k)	Zero-shot	0.3842	0.2797	0.3237	NA

BOW = bag-of-words; CNN = convolutional neural network; Params = parameters; P = precision, R = recall, ROC AUC= receiver operating characteristic/area under the curve; SFT = supervised fine-tuned

Table 3 Population level section pairs of the most highlighted sections when using hierarchical transformers.

Section 1	Section 2	Adjusted frequency
brief hospital course	history of present illness	1
brief hospital course	admission date : discharge date	0.882317
brief hospital course	major surgical or invasive procedure	0.882317
brief hospital course	allergies	0.882317

brief hospital course	followup instructions	0.871126
brief hospital course	chief complaint	0.864856
brief hospital course	name : unit no	0.819268
history of present illness	pertinent results	0.74036
brief hospital course	past medical history	0.714455
history of present illness	followup instructions	0.711694
major surgical or invasive procedure	pertinent results	0.691837
allergies	pertinent results	0.691236
admission date : discharge date	pertinent results	0.691236
chief complaint	pertinent results	0.674251
admission date : discharge date	followup instructions	0.657774
allergies	followup instructions	0.657774
followup instructions	major surgical or invasive procedure	0.657774
name : unit no	pertinent results	0.643258
chief complaint	followup instructions	0.636643
followup instructions	name : unit no	0.609727

Comparative analysis on model predictions

Figure 5 shows a Venn diagram of the true positive and false positive samples from three models: CNN, Hierarchical Transformer, and Mixtral.The diagrams illustrate that the predictions from the two supervised models have different characteristics when compared to those from zero-shot Mixtral, a large LM. This is unsurprising, as these supervised models are strongly influenced by the dataset models are trained on, whereas the large LMs have presumably never seen the dataset.

Sixteen samples were reviewed by a board-certified critical care physician and clinical informatics expert (MA) to understand the face validity of the label and difficulty of the task. The samples were selected across the various categories: 3 were common true positives, 7 were

common false positives where 3 among them were samples without date of death records (For complete information, please see Appendix C. Overall, the physician commented that predicting a specific time window such as 30 or 60 days was difficult. This finding agreed with multiple prior studies showing that prognostication is clinically challenging in patients with serious illness, and even experienced physicians tend to overestimate survival [38–41]. Incorrect prognostication can hinder end-of-life discussions, lead to more aggressive and potentially over-treatment, and lead to interventions that are not in line with patients' goals-of-care. In the outpatient oncology setting, machine learning-guided prognostication has been found to improve advanced care planning documentation and serious illness conversations, which could improve end-of-life care. In the inpatient intensive care setting, models such as those developed here could be used to identify patients who may have lower probability of survival to improve end-of-life planning and care.

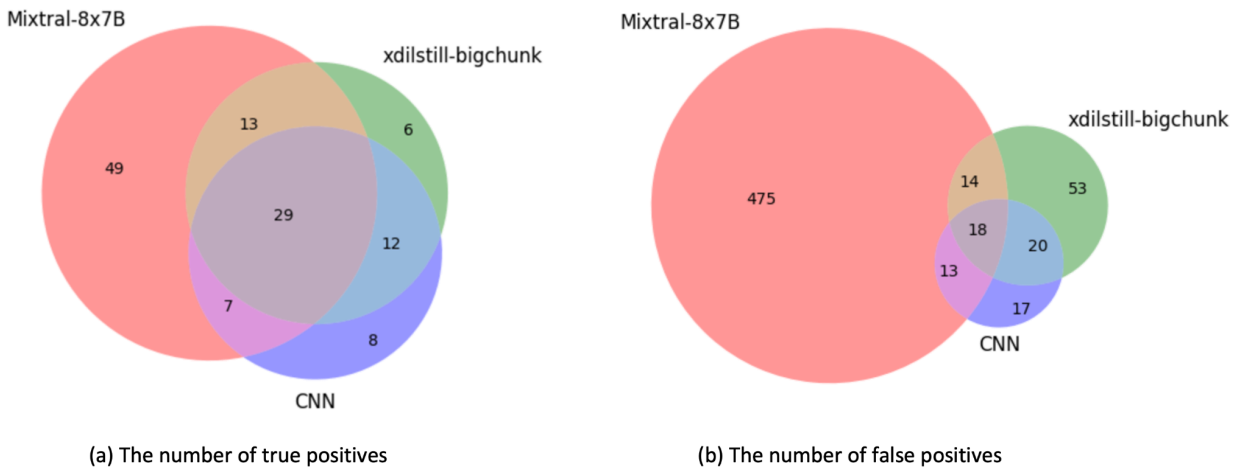

Figure 5. Venn diagram of three model predictions. Numbers in the diagram denote the number of instances in each category. Left diagram (a) shows true positives and the right diagram (b) shows false positive cases.

Common predictions: The intersection of all three models in true positive and false positive, suggests that those instances can be "easy" cases and "difficult" cases (including sudden death or label error) respectively. We examine these cases more closely, both to understand the behavior of our models, and to validate the quality of the benchmark dataset.

All three models have true positive predictions on 29 instances, which can be considered as easy-to-predict examples. Among 29 instances, 26 are identified as having comfort care mentions in the note and another partition of 26 are identified as having DNR/DNI mentions. All of the 29 instances have at least one of those two keyword sets. Note that patients identifiable as discharged to hospice through structured data were excluded from the dataset during pre-processing steps. Some of the 26 patients had discussion for discharge to *hospice facility* but were not actually discharged to there according to the structured data (cf. they were discharged to home with hospice care or alternative facilities like SKILLED NURSING FACILITY or CHRONIC/LONG TERM ACUTE CARE).

The remaining three samples were manually examined. From the structured data, they passed away in 7, 10, and 22 days. Physician commented that the labels for these three patient cases had face validity. Based on our analysis, we did not find any anomalies in the labels of all 29 instances.

For false positive predictions, three models have 18 instances in common. Since all machine-learning models predicted these negative instances as positives, instances in this category can be treated as difficult instances. These false positive predictions can be interpreted in multiple ways: the patient's condition is severely bad but the patient survived, or the prediction is correct but the label is erroneous (please see *Limitation* section for the further discussion). Our dictionary-based detection found comfort care terms from 13 notes and we manually reviewed the rest of 5 notes where it cannot find the term. Three cases survived less than 1 year and among them, two passed away after 61 and 106 days. One of the other two patients survived about 1 year and 9 months. The last patient had a comfort care mention that our detection mechanism could not catch due to euphemistic language (please see a paragraph entitled *Various mentions about comfort care* for details). This patient did not have a date of death record but our reviewing physician commented that this patient has a high possibility of death in a short period (MIMIC-IV censors death dates at one year after last discharge, so the patient may have survived over one year, or may have been lost to follow-up and died in another state). In summary, some of our data instances raise challenging points to the models, which we believe are important for the discriminative ability of a benchmark dataset. The model predictions were also reasonable and the errors are likely to happen even for a well trained model or domain experts.

Distinct predictions: Hierarchical transformer (xdistill-bigchunk) had 6 distinct true positive predictions that other models failed to predict correctly (Green area in Figure 5 - (a)). Among 6 distinct true positives, 3 mentioned comfort care and 3 did not. The former means that the other models did not predict these instances as negative cases, which can be interpreted as models recognizing other signals of survival from the text even though they were not correct predictions. This is also true for the manual analysis, the physician predicted 2 out of 3 cases to survive more than 30 days.

Attention of Hierarchical Transformer Model

We looked into the vector norm values of the hierarchical transformer to see which chunks, input units of the chunk attention layers, are highlighted during the prediction. During prediction of one of the notes without comfort care mentions, the model had highlights on the 5th chunk that has ***# icu course*** part of ***brief hospital course :*** and the last chunk, which has discharge information where a part of ***discharge medications:, discharge disposition :, discharge diagnosis :, discharge condition :, and discharge instructions :,*** and ***discharge instructions :*** sections are written (Figure 6). The brief hospital course provides informative background about the clinical findings pertaining to a patient's brain injury, while the discharge information provides complementary, non-overlapping information indicating the level of severity of the injury and mental status at the time of discharge.

Following is a part of 5th chunk:

```
# icu course
on admission , patient was monitored on cveeg with no seizures
captured . some left temporal epileptiform discharges were seen in a
semirhythmic pattern ( pleds ) , but they were not frequent or
concerning for seizure . she was continued on keppra 1500mg bid with
no seizures seen . she had a cth , which was suspicious for large
left mca stroke . mri was obtained which was concerning for
hypoglycemia related damage vs hypoxic ischemic encephalopathy with
cortical necrosis vs post - ictal changes . cta did not show vessel
abnormalities . repeat mri was performed on , and showed stable
changed . etiology of her exam was felt to be a combination of
hypoglycemia and hypoxia .

she remained intubated and off sedation for her entire stay . during
her icu stay , she began to have more spontaneous movement of her
lower extremities , and would intermittently open her eyes , and
maintained her brainstem reflexes on minimal ventilator settings .
she did not regard , track , or follow any commands . an mri was
repeated on , which showed persistent cortical slow diffusion within
left greater than right cerebral hemispheres with parietal /
temporal predominance , and new gyriform contrast enhancement ,
including a new discrete t2 hyperintense and enhancing focus in the
medial left temporal lobe .
<Omitted>
```

In this chunk, we note the patient has evidence of hypoxic brain injury and remained in a non-cognitive state that required dependence on breathing and feeding life support. .

Following is a part of the last chunk:

```
 discharge medications :
 1 . acetaminophen 650 mg
…
<Omitted>

 discharge diagnosis :
 hypoglycemic encephalopathy
 hypoxic ischemic brain injury
 urinary tract infection


 discharge condition :
 mental status : confused - always .
 level of consciousness : lethargic but arousable .
 activity status : bedbound .
```

```
 discharge instructions :
 dear ms . ,
 you were hospitalized after severely low blood sugars and
brain
 injury caused by insulin overdose . you were started on
 medication to prevent seizures . you will need to go to a
nursing
 facility to help you take care of yourself .

 it was a pleasure taking care of you ,
 your neurologists
```

In this chunk, we could again confirm that the patient had hypoxic ischemic brain injury and low blood sugars, while gaining new information about her mental status and clinical condition at the time she was discharged. While it is reasonably clear from the latter section that the patient's condition has a poor prognosis, the earlier section contains detailed information of their problems that could give the model more fine-grained information that could modulate the model's estimation of their condition's severity and neurologic function.

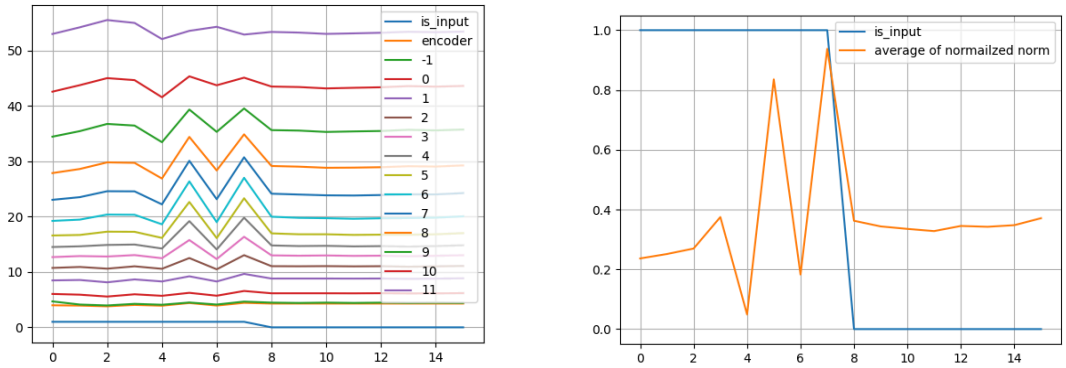

Figure 6. Model highlights during the prediction of an example instance. The position of the chunk is represented on the X-axis, and the vector norm of the layers is shown on the Y-axis. The left graph shows all layers and the right graph shows the average of all layers. Value of "is_input" denotes whether the chunk is composed of actual inputs versus padding tokens. In this graph, chunks in the first to 7th position have real input values but from the 8th chunk, chunks are filled with padding tokens.

DISCUSSION

Limitations

We first discuss the difference in settings between the baseline models, and the foreseeable impact of those differences. In the latter part of this section, we discuss the limitation of our benchmark dataset regarding the source of information and our methodology in analysis.

Different settings in baseline models: Models inherently have different settings due to the nature of their architectures. One of the notable setting differences is the variance in maximum token length for input instance across the models.

Max token length can highly impact models regardless of the structure [42], yet it can be more impactful for large LMs due to its nature of using the prompting strategy. Prompts for large LMs include system prompts, questions, and the input sequences. Users are also required to leave space for output response tokens. Moreover, large LM performance may be improved with few-shot settings, in which the prompts include one or a few training instances in the prompt, making it more important to discover the impact of a shortened input sequence.

Post-hoc experiments on Mixtral showed that when the maximum token length is limited to 2048, the performance dropped by 11 percent in absolute difference, which is about half of the performance of the full-length model.

	Performance			Predictions		
Length	Prec	Rec	F1	Positive	Negative	Pos/Neg
2048	0.08	0.17	0.11	560	7008	7.40%
8196	0.16	0.38	0.22	618	6950	8.17%

Table 4 Performance of Mixtral model by the length of input text.

For the zero-shot setting with large LMs, the performance of the models relies on how the prompt is formulated. Sometimes the model cannot produce answers that comply with the suggested answer format. For example, our prompt requires the model to answer only between 0:alive or 1:death but sometimes answers were started with “Based on the information provided,” not matching the requested format. 165 predictions from Mixtral 8*7B included the above mentioned phrase. To alleviate this problem, Gao et al.[43] proposed an alternative prompt method for zero-shot evaluation named *harness*. However, this approach can only be applied to models that support output of probability, meaning that most cloud-based models like GPT-4 cannot be evaluated using this method.

Source of data of death labels: MIMIC-IV v2.2 dataset utilized the Massachusetts Registry of Vital Records (cf. Death Certificate is public record in the state of Massachusetts) to enrich the date of death record. According to the MIMIC-IV paper, the state registry was selected instead of the Social Security Death Master File due to data quality concerns[44]. However using state registry cannot fully resolve the data concerns as patients who moved out of the state cannot be traced with this method. For example, among 18 instances of common false positive cases (i.e. union of three models used in the Comparative analysis section), three patients do not have date of death (DoD) records. We requested the physician to review these instances and found out that all of these patients are severely ill and less likely to survive long enough after discharge, meaning that these three labels may be erroneous.

Despite this intrinsic limitation, we believe the state registry is still one of the most viable options when creating a database.

Various mentions about comfort care: During our post-experiment analysis, we learned that palliative care and comfort care terms are largely varied and cannot be efficiently extracted using dictionary-based measures. Decisions on comfort care, especially in discharge notes, are highly transformed in a euphemistic manner, making it challenging to detect them. For example, there were mentions such as "*aimed at keeping you as comfortable as possible*" or more indirect mentions, "*we all met as a group to discuss the kind of care that will give you the greatest number of happy days.*"
In the first example, the "comfortable" is the key mention to identify it as comfort care. However, filtering using the single term "comfortable" is not viable as it can raise false positive cases for mentions like, "*He is tachypneic but otherwise comfortable appearing.*" A possible direction for further study involves developing robust models capable of recognizing and normalizing euphemistic terms with clinical significance.

Our rule-based extraction of mentions about comfort care found that 220 out of 7568 patients had discussion about comfort care that is not extractable from structured data. We did not exclude these patients, because this may make supervised models overfit to these terms as they can be a strong signal for the mortality task.

CONCLUSION

In this paper, we present a benchmark for evaluating long clinical document processing, entitled LCD benchmark. We tested our benchmark dataset using baseline methods, ranging from Bag-of-words to zero-shot prediction with large LMs. As a result of these methods along with further analysis, we showed that the LCD benchmark presents challenges and the potential for improvement in current neural network-based approaches. During our experiments with large LMs, we further explored the importance of their capability to process longer sequences. Additionally, we raised questions regarding the formulation of prompts. Our benchmark dataset is publicly available for the researchers who gained access to the MIMIC-IV datasets.

DATA AVAILABILITY STATEMENT

The data underlying this article are available in a github repository, at https://github.com/Machine-Learning-for-Medical-Language/long-clinical-doc .
The datasets were derived from sources: https://physionet.org/content/mimiciv/2.2/ and https://physionet.org/content/mimic-iv-note/2.2/

FUNDING

Research reported in this publication was supported by the National Library Of Medicine of the National Institutes of Health under Award Number R01LM012973, and by the National Institute Of Mental Health of the National Institutes of Health under Award Number R01MH126977.

AUTHOR NOTE

For the large language models we used, we do not have control of their training materials. Our experiments, including those with large LMs, were conducted in HIPAA Protected environments, which blocks third parties from using our data for reviewing or training purposes.

REFERENCES:

1. Rosenbloom, S. T. *et al.* Data from clinical notes: a perspective on the tension between structure and flexible documentation. *J. Am. Med. Inform. Assoc. JAMIA* **18**, 181–186 (2011).
2. Savova, G. K. *et al.* Use of Natural Language Processing to Extract Clinical Cancer Phenotypes from Electronic Medical Records. *Cancer Res.* **79**, 5463–5470 (2019).
3. Demner-Fushman, D., Chapman, W. W. & McDonald, C. J. What can natural language processing do for clinical decision support? *J. Biomed. Inform.* **42**, 760–772 (2009).
4. Yim, W., Yetisgen, M., Harris, W. P. & Kwan, S. W. Natural Language Processing in Oncology: A Review. *JAMA Oncol.* **2**, 797–804 (2016).
5. Wu, S. *et al.* Deep learning in clinical natural language processing: a methodical review. *J. Am. Med. Inform. Assoc.* **27**, 457–470 (2020).

6. Rasmy, L. *et al.* Representation of EHR data for predictive modeling: a comparison between UMLS and other terminologies. *J. Am. Med. Inform. Assoc.* **27**, 1593–1599 (2020).
7. Si, Y. *et al.* Deep representation learning of patient data from Electronic Health Records (EHR): A systematic review. *J. Biomed. Inform.* **115**, 103671 (2021).
8. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (eds. Burstein, J., Doran, C. & Solorio, T.) 4171–4186 (Association for Computational Linguistics, Minneapolis, Minnesota, 2019). doi:10.18653/v1/N19-1423.
9. Alsentzer, E. *et al.* Publicly Available Clinical BERT Embeddings. in *Proceedings of the 2nd Clinical Natural Language Processing Workshop* (eds. Rumshisky, A., Roberts, K., Bethard, S. & Naumann, T.) 72–78 (Association for Computational Linguistics, Minneapolis, Minnesota, USA, 2019). doi:10.18653/v1/W19-1909.
10. Chen, S. *et al.* Natural Language Processing to Automatically Extract the Presence and Severity of Esophagitis in Notes of Patients Undergoing Radiotherapy. *JCO Clin. Cancer Inform.* e2300048 (2023) doi:10.1200/CCI.23.00048.
11. Beltagy, I., Peters, M. E. & Cohan, A. Longformer: The Long-Document Transformer. *arXiv.org* https://arxiv.org/abs/2004.05150v2 (2020).
12. Li, Y., Wehbe, R. M., Ahmad, F. S., Wang, H. & Luo, Y. Clinical-Longformer and Clinical-BigBird: Transformers for long clinical sequences. Preprint at https://doi.org/10.48550/arXiv.2201.11838 (2022).
13. Singhal, K. *et al.* Large language models encode clinical knowledge. *Nature* **620**, 172–180 (2023).
14. Kojima, T., Gu, S. S., Reid, M., Matsuo, Y. & Iwasawa, Y. Large Language Models are Zero-Shot Reasoners. Preprint at http://arxiv.org/abs/2205.11916 (2023).

15. Agrawal, M., Hegselmann, S., Lang, H., Kim, Y. & Sontag, D. Large Language Models are Few-Shot Clinical Information Extractors. Preprint at http://arxiv.org/abs/2205.12689 (2022).

16. Su, X., Miller, T., Ding, X., Afshar, M. & Dligach, D. Classifying Long Clinical Documents with Pre-trained Transformers. Preprint at https://doi.org/10.48550/arXiv.2105.06752 (2021).

17. Wright, A. A. *et al.* Associations Between End-of-Life Discussions, Patient Mental Health, Medical Care Near Death, and Caregiver Bereavement Adjustment. *JAMA* **300**, 1665–1673 (2008).

18. Temel, J. S. *et al.* Early Palliative Care for Patients with Metastatic Non–Small-Cell Lung Cancer. *N. Engl. J. Med.* **363**, 733–742 (2010).

19. Sullivan, D. R. *et al.* Association of Early Palliative Care Use With Survival and Place of Death Among Patients With Advanced Lung Cancer Receiving Care in the Veterans Health Administration. *JAMA Oncol.* **5**, 1702–1709 (2019).

20. Kelley, A. S. & Morrison, R. S. Palliative Care for the Seriously Ill. *N. Engl. J. Med.* **373**, 747–755 (2015).

21. Johnson, A. E. W. *et al.* MIMIC-III, a freely accessible critical care database. *Sci. Data* **3**, 160035 (2016).

22. Johnson, A. E. W. *et al.* MIMIC-IV, a freely accessible electronic health record dataset. *Sci. Data* **10**, 1 (2023).

23. Harutyunyan, H., Khachatrian, H., Kale, D. C., Ver Steeg, G. & Galstyan, A. Multitask learning and benchmarking with clinical time series data. *Sci. Data* **6**, 96 (2019).

24. Wang, S. & Manning, C. Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (eds. Li, H., Lin, C.-Y., Osborne, M., Lee, G. G. & Park, J. C.) 90–94 (Association for Computational Linguistics, Jeju Island, Korea, 2012).

25. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**,

2825–2830 (2011).

26. Kim, Y. Convolutional Neural Networks for Sentence Classification. in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (eds. Moschitti, A., Pang, B. & Daelemans, W.) 1746–1751 (Association for Computational Linguistics, Doha, Qatar, 2014). doi:10.3115/v1/D14-1181.
27. Child, R., Gray, S., Radford, A. & Sutskever, I. Generating Long Sequences with Sparse Transformers. Preprint at https://doi.org/10.48550/arXiv.1904.10509 (2019).
28. Jiang, A. Q. *et al.* Mistral 7B. Preprint at http://arxiv.org/abs/2310.06825 (2023).
29. Dettmers, T., Pagnoni, A., Holtzman, A. & Zettlemoyer, L. QLoRA: Efficient Finetuning of Quantized LLMs. Preprint at http://arxiv.org/abs/2305.14314 (2023).
30. Clinical NLP Transformers (cnlp_transformers).
31. Kobayashi, G., Kuribayashi, T., Yokoi, S. & Inui, K. Attention is Not Only a Weight: Analyzing Transformers with Vector Norms. in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (eds. Webber, B., Cohn, T., He, Y. & Liu, Y.) 7057–7075 (Association for Computational Linguistics, Online, 2020). doi:10.18653/v1/2020.emnlp-main.574.
32. Clark, K., Khandelwal, U., Levy, O. & Manning, C. D. What Does BERT Look at? An Analysis of BERT's Attention. in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (eds. Linzen, T., Chrupa\la, G., Belinkov, Y. & Hupkes, D.) 276–286 (Association for Computational Linguistics, Florence, Italy, 2019). doi:10.18653/v1/W19-4828.
33. Kovaleva, O., Romanov, A., Rogers, A. & Rumshisky, A. Revealing the Dark Secrets of BERT. in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (eds. Inui, K., Jiang, J., Ng, V. & Wan, X.) 4365–4374 (Association for Computational Linguistics, Hong Kong, China, 2019). doi:10.18653/v1/D19-1445.

34. Reif, E. *et al.* Visualizing and Measuring the Geometry of BERT. in *Advances in Neural Information Processing Systems* vol. 32 (Curran Associates, Inc., 2019).

35. Lin, Y., Tan, Y. C. & Frank, R. Open Sesame: Getting inside BERT's Linguistic Knowledge. in *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (eds. Linzen, T., Chrupa\la, G., Belinkov, Y. & Hupkes, D.) 241–253 (Association for Computational Linguistics, Florence, Italy, 2019). doi:10.18653/v1/W19-4825.

36. Jain, S. & Wallace, B. C. Attention is not Explanation. Preprint at http://arxiv.org/abs/1902.10186 (2019).

37. Serrano, S. & Smith, N. A. Is Attention Interpretable? in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics* (eds. Korhonen, A., Traum, D. & Màrquez, L.) 2931–2951 (Association for Computational Linguistics, Florence, Italy, 2019). doi:10.18653/v1/P19-1282.

38. Detering, K. M., Hancock, A. D., Reade, M. C. & Silvester, W. The impact of advance care planning on end of life care in elderly patients: randomised controlled trial. *BMJ* **340**, c1345 (2010).

39. Cheon, S. *et al.* The accuracy of clinicians' predictions of survival in advanced cancer: a review. *Ann. Palliat. Med.* **5**, 22–29 (2016).

40. Gripp, S. *et al.* Survival Prediction in Terminally Ill Cancer Patients by Clinical Estimates, Laboratory Tests, and Self-Rated Anxiety and Depression. *J. Clin. Oncol.* **25**, 3313–3320 (2007).

41. Glare, P. *et al.* A systematic review of physicians' survival predictions in terminally ill cancer patients. *BMJ* **327**, 195–198 (2003).

42. Liu, N. F. *et al.* Lost in the Middle: How Language Models Use Long Contexts. Preprint at http://arxiv.org/abs/2307.03172 (2023).

43. Gao, L. *et al.* A framework for few-shot language model evaluation. (2023) doi:10.5281/zenodo.10256836.

44. Levin, M. A., Lin, H.-M., Prabhakar, G., McCormick, P. J. & Egorova, N. N. Alive or dead: Validity of the Social Security Administration Death Master File after 2011. *Health Serv. Res.* **54**, 24–33 (2019).