Students Need More Attention: BERT-based Attention Model for Small Data with Application to Automatic Patient Message Triage

Shijing Si[*] SHIJING.SI@DUKE.EDU
Rui Wang[*] RW161@DUKE.EDU
Jedrek Wosik[†] JEDREK.WOSIK@DUKE.EDU
Hao Zhang[*] HZ210@DUKE.EDU
David Dov[*] DAVID.DOV@DUKE.EDU
Guoyin Wang[‡] GUOYINWANG.DUKE@GMAIL.COM
Ricardo Henao[*] RICARDO.HENAO@DUKE.EDU
Lawrence Carin[*] LCARIN@DUKE.EDU
[*] *Duke University, Durham, NC, USA*
[†] *Duke University School of Medicine, Durham, NC, USA*
[‡] *Amazon Alexa AI, Seattle, WA, USA*

Editor: Editor's name

Abstract

Small and imbalanced datasets commonly seen in healthcare represent a challenge when training classifiers based on deep learning models. So motivated, we propose a novel framework based on BioBERT (Bidirectional Encoder Representations from Transformers for Biomedical TextMining). Specifically, (*i*) we introduce Label Embeddings for Self-Attention in each layer of BERT, which we call LESA-BERT, and (*ii*) by distilling LESA-BERT to smaller variants, we aim to reduce overfitting and model size when working on small datasets. As an application, our framework is utilized to build a model for patient portal message triage that classifies the urgency of a message into three categories: non-urgent, medium and urgent. Experiments demonstrate that our approach can outperform several strong baseline classifiers by a significant margin of 4.3% in terms of macro F1 score. The code for this project is publicly available at `https://github.com/shijing001/text_classifiers`

1. Introduction

Online patients portals, *e.g.*, MyChart by Epic, have become increasingly prevalent tools for communication between patients and healthcare providers (Ramsey et al., 2018). These portals have the potential to boost the productivity of providers, improve patient satisfaction, and reduce communication barriers (Goldzweig et al., 2013; Sieck et al., 2017). Despite these benefits, patient-provider communication tools have produced unintended consequences such as increased, often unpaid, workload for providers (Hefner et al., 2019).

Further, due to the flood of non-urgent incoming patient messages, some, which may require a timely emergency response, can be delayed or effectively neglected.

To address these concerns, we consider the task of automated patient message classification to estimate the urgency of messages based on their content. Our dataset consists of $1,756$ messages collected from a University Hospital's portal. These messages, manually adjudicated by experienced healthcare providers, were grouped into three categories: non-urgent, medium, and urgent, as summarized in Table 1. With merely 170 urgent instances, our dataset is both small and imbalanced, posing a significant challenge in terms of properly training machine-learning-based classifiers. This challenge is common in many clinical datasets (Zhao et al., 2018; Para et al., 2019) due to the fact that manually labeling data is very laborious, time-consuming, expensive and oftentimes prohibitive. Further, certain labels are rare by nature, for instance, urgent electronic messages are far less common because patients tend to call the healthcare provider directly rather than using the portal.

Label	Count	Typical Example
Non-urgent	631	That would be awesome... thank you.
Medium	955	Dr. [name]. All seems well now. I am at home resting. My wife and I have a trip planned to Maryland this week beginning on Wednesday. We can fly, drive or stay home if I should not travel. Are there any reasons that I should not fly.
Urgent	170	I have continued having chest pain shortness of breath since waking. Please tell me what to do. I have tried in hailers am going to try nebulizers. I just feel extremely tight in my chest.

Table 1: Typical examples of patient messages to providers grouped by urgency. These are examples of the message urgency dataset used in the experiments.

Machine learning approaches have been previously applied to message classification tasks in the healthcare domain. For example, Cronin et al. (2015) utilized logistic regression and random forest algorithms to classify between patient health information needs such as symptom management and medication side effects, based on their messages. Cronin et al. (2017) studied the use of rule-based and random forest classifiers to classify patient portal messages into broad communication types, *e.g.*, appointment rescheduling, examination enquiry, *etc.* Sulieman et al. (2017) showed that Convolutional Neural Networks (CNNs) outperform traditional classifiers in portal message classification, and Tafti et al. (2019) developed an ensemble of neural networks for text classification to categorize free-text patient portal messages as either containing active symptom descriptions or logistic requests. Chen et al. (2019) leveraged traditional machine learning methods such as Support Vector Machines (SVMs) to detect hypoglycemia incidents reported in patient messages. To the best of our knowledge, most existing classifiers employ either traditional machine learning methods such as SVMs or shallow networks for the classification of patient generated messages.

Recently, the field of natural language processing (NLP) has seen significant progress in large-scale pre-trained language models, which often considerably boost the performance of classifiers on text classification tasks (Devlin et al., 2018; Radford et al., 2019). Here, we introduce the use of Bi-directional Encoder Representations from Transformers (BERT) (Devlin et al., 2018) for patient message classification. BERT produces sequence representations using a multi-layered attention mechanism, which models associations between the tokens (words) of a sentence (Vaswani et al., 2017). We consider the challenging task of properly training such complex models on a small and imbalanced dataset, which is complicated because these models typically have hundreds of millions of parameters. Previous studies addressed this challenge by pre-training BERT on a large dataset and then fine-tuning it on the (small) dataset at hand (Devlin et al., 2018). However, the performance of this procedure is usually limited by the size and difficulty of the available small dataset.

Generalizable Insights from Machine Learning in the Context of Healthcare

In this paper we address this challenge with a framework which beyond the use of a pre-trained model, has two novel components. First, we introduce a novel attention mechanism that utilizes label embeddings to better capture associations between the labels and the tokens of the input sequence. Importantly, the proposed attention mechanism can be incorporated into existing attention layers with minimum modification, allowing the use of pre-trained BERT or BioBert models (Devlin et al. (2018); Lee et al. (2020)). We term our method LESA-BERT, short for Label Embedding on Self-Attention in BERT. Second, a large deep learning model like BERT has millions of parameters and tends to overfit on small datasets. We hypothesize that a model with a reduced number of parameters may result in a smaller generalization error, thus likely improved performance. Accordingly, we employ the knowledge distillation technique to compress the LESA-BERT model, by training a smaller *student* model to reproduce the prediction ability of the *teacher*, a fine-tuned LESA-BERT model. From our framework, we devise distilled variants of LESA-BERT. We demonstrate LESA-BERT by building an automatic message triage classifier that can predict the urgency of messages based on their content. Experiments demonstrate that LESA-BERT and distilled variants result in improved performance compared to multiple baseline approaches. Specifically, LESA-BERT outperforms the baselines by a 2.5% margin in terms of macro F1 score, while the distilled LESA-BERT with 6 encoder layers (Distil-LESA-BERT-6) further outperforms LESA-BERT by 1.8%. Therefore, in total Distil-LESA-BERT-6 outperforms the baseline models by 4.3% in F1 score.

2. Related Work

BERT Widely used for natural language processing, BERT produces sentence representations through a multi-layered attention mechanism that encodes relations between the tokens that compose the sentence (Vaswani et al., 2017; Devlin et al., 2018). Due to its complex structure, which encompasses a large number of parameters, BERT is typically pre-trained on a large dataset and then fine-tuned on the target dataset. Standard pre-training approaches include unsupervised tasks such as masked language modeling and next sentence prediction, which leverage massive unlabeled, general domain corpora like English Wikipedia (2,500M words) or BooksCorpus (800M words) (Zhu et al., 2015). Subsequently, the model is fine-tuned by adding one or more additional layers, the parameters of which are optimized using the target dataset, in a technique termed transfer learning. Recently BERT-based models have been leveraged to NLP tasks in healthcare. Huang et al. (2019) developed ClinicalBERT by pre-training BERT on a large set of clinical notes, and utilized it to predict patients' 30-day hospital readmission based on both discharge summaries and

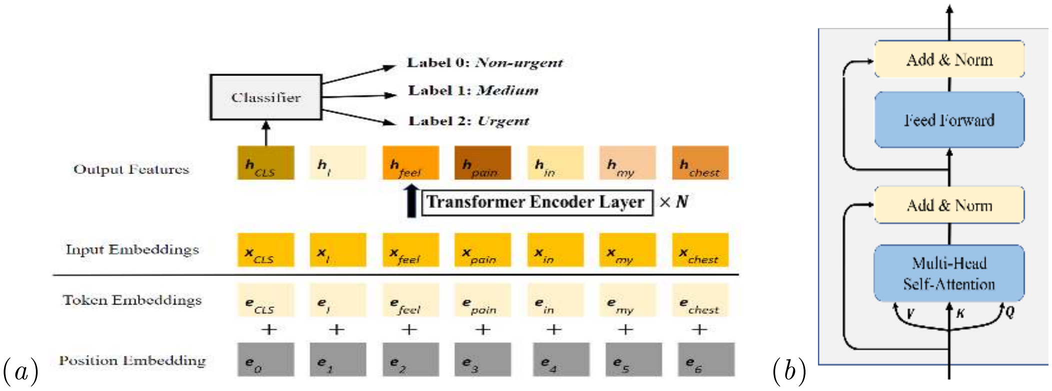


Figure 1: (a) BERT model structure for text classification on the message urgency dataset. Token and position embeddings are summed up as input embeddings, then fed through N transformer encoder layers, yielding a high-level representations (features) for classification. (b) Architecture of one transformer encoder layer.

clinical notes. Lee et al. (2020) introduced BioBERT, which is essentially BERT pre-trained on large biomedical corpora that includes PubMed abstracts and PubMed Central full-text articles. They showed that pre-training the model directly on the domain of interest leads to improved performance on various biomedical NLP tasks.

Label Embedding Label embedding is a technique that embeds class labels along with the (text) data into a joint latent space, where the model can be trained to cross-attend the inputs and labels to boost the performance of deep learning models. Label embeddings were previously leveraged for image classification (Akata et al., 2015), multi-modal learning between images and text (Kiros et al., 2014), text recognition in images (Rodriguez-Serrano and Perronnin, 2015), zero-shot learning (Li et al., 2015; Ma et al., 2016) and text classification Zhang et al. (2017). Notably, Wang et al. (2018) proposed a framework named Label Embedding Attentive Model (LEAM), which jointly embeds the words and labels in a common latent space, and improves the performance on general text classification tasks. Inspired by LEAM, we consider the joint representation of the message and its corresponding class token, and propose to incorporate label embeddings to the self-attention mechanism inside BERT encoders, which improves the attention between the class token to the other tokens in the message.

Knowledge Distillation Knowledge distillation is a technique used to train a small model usually called *student* to reproduce the predictions, thus performance, of a larger model called a *teacher* (Hinton et al., 2015). It is typically used to compress models, which reduces their storage and computational costs, thus facilitating their deployment. Model distillation has been previously applied to compress complex models such as BERT and long short-term memory (LSTM) networks (Sanh et al., 2019; Tang et al., 2019). When the training data size is small, we hypothesize that the teacher model may be prone to overfitting. In this situation, knowledge distillation is more likely to produce a student model that outperforms the larger, more complex teacher model, as we show in our experiments.

3. Methods

3.1. Background

BERT is an architecture composed of a stack of transformer encoder layers, each including two sub-layers: a *multi-head self-attention* module and a *feed forward* network. Each of these encoder layers, shown in Figure 1(b), is structured as a residual block with appropriate layer normalization and dropout (He et al., 2016; Ba et al., 2016). In Figure 1(a) we show how to fine-tune BERT-based models on the message urgency dataset to build a classifier. First, each text sequence of length L is prepended with the special token $[CLS]$. The sum of token and position embeddings, $[\boldsymbol{x}_{[CLS]}, \boldsymbol{x}_{\text{token}_1}, \ldots, \boldsymbol{x}_{\text{token}_L}]$, are represented as $[\boldsymbol{e}_{[CLS]}, \boldsymbol{e}_{\text{token}_1}, \ldots, \boldsymbol{e}_{\text{token}_L}]$ and $[\boldsymbol{e}_0, \ldots, \boldsymbol{e}_L]$ vectors, respectively. We use $[\boldsymbol{h}_{[CLS]}, \boldsymbol{h}_{\text{token}_1}, \ldots, \boldsymbol{h}_{\text{token}_L}]$ vectors to denote the output, high-level representation of classification and input tokens, each of which has the same dimensionality as the input. After applying N (usually $N = 12$) different encoder layers (of the same structure in Figure 1(b) but different weight matrices), we obtain the high-level representations (output features) of the input sequence, in which each token representation contains information of other tokens. Finally, the first row of the output features, $\boldsymbol{h}_{CLS}$, is considered as the global sequence aggregator and thus fed through a softmax classification function composed of one or more fully connected layers.

The multi-head self-attention module is an ensemble of multiple attention modules sharing the same formulation. Given a text sequence $\boldsymbol{t}$ represented as embedding matrix $\mathbf{X} \in \mathbb{R}^{(L+1)\times D}$, L represents the length of the text sequence and D the token and position embedding dimensions. Note that the first row in $\mathbf{X}$ corresponds to the $[CLS]$ token, hence the $L+1$ rows in $\mathbf{X}$. For a single-attention head, tokens of $[CLS]$ and the input sequence are first mapped into the key, query and value triplets, denoted as matrices $\mathbf{K} \in \mathbb{R}^{(L+1)\times d}$, $\mathbf{Q} \in \mathbb{R}^{(L+1)\times d}$ and $\mathbf{V} \in \mathbb{R}^{(L+1)\times d}$, respectively, via:

$$\mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V, \tag{1}$$

where $\{\mathbf{W}_K, \mathbf{W}_Q, \mathbf{W}_V\} \in \mathbb{R}^{D\times d}$ are learnable parameters for the key, query and value of self-attention. With $\mathbf{K}$, $\mathbf{Q}$ and $\mathbf{V}$, the attention mechanism can be formulated as

$$\begin{aligned} \mathbf{A} &= \frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} \in \mathbb{R}^{(L+1)\times(L+1)} \\ \mathbf{O}_i &= \text{attention}_i(\mathbf{K},\mathbf{Q}, \mathbf{V}) = \text{softmax}(\mathbf{A})\mathbf{V} \in \mathbb{R}^{(L+1)\times d}, \end{aligned} \tag{2}$$

where $i = 1, \ldots, h$, h is the number of attention heads, softmax$(\cdot)$ is the softmax function applied row-wise. $\mathbf{A}$ is the attention score matrix representing the compatibility of $\mathbf{K}$ and $\mathbf{Q}$ obtained via inner products. The multi-head self-attention is defined by concatenating and projecting the representation of each head as

$$\mathbf{O} = [\mathbf{O}_1, \cdots, \mathbf{O}_h]\mathbf{W} \in \mathbb{R}^{(L+1)\times D}, \tag{3}$$

where $[\cdot,\cdot]$ denotes column-wise concatenation and $\mathbf{W} \in \mathbb{R}^{(d\times h)\times D}$ is a learnable projection matrix.

Label	Key Words/Phases
Medium	Loss of coordination/balance, Dizziness, Near syncope, Leg swelling, Headache
Urgent	Blue lips, Chest pain, Disorientation, Paralysis, Loss of consciousness

Table 2: Key words/phases for initialization of label embeddings.

After the multi-head self-attention module, the position-wise (one token at the time) feed forward network module in Figure 1(b) composed of two fully connected layers is applied,

$$\mathrm{FFN}(\boldsymbol{x}) = \max(0, \boldsymbol{u}\mathbf{W}_1 + \boldsymbol{b}_1)\mathbf{W}_2 + \boldsymbol{b}_2, \tag{4}$$

where $\max(0,\cdot)$ is the standard ReLU activation function, $\{\mathbf{W}_1, \mathbf{W}_2, \boldsymbol{b}_1, \boldsymbol{b}_2\}$ are learnable parameters, and $\boldsymbol{u}$ is the layer normalized residual block $\boldsymbol{u} = \mathrm{LayerNorm}(\boldsymbol{x} + \boldsymbol{o})$, where $\boldsymbol{x}$ (rows of $\mathbf{X}$) and $\boldsymbol{o}$ (rows of $\mathbf{O}$) are the inputs and outputs of the multi-head self-attention module in (1)–(3), respectively, and the $\mathrm{LayerNorm}(\cdot)$ operator is implemented according to (Ba et al., 2016).

3.2. Multi-head Attention with Label Embedding

For the patient message triage task, the labeled patient text data is usually limited and hard to obtain. Specially, data for the urgent class is scarce but of the highest priority for the application. Therefore, we incorporate *label embeddings* as prior information into the self-attention modules, so that the model can more easily attend to class-representative keywords. The modified module is shown in Figure 2(a) and described below. We asked healthcare providers to select a set of keywords or short phrases associated with the classes medium and urgent. These are shown in Table 2. The label embeddings of each class is initialized as the average of the corresponding keyword embeddings. The label embedding for non-urgent class is initialized at randomly provided there are no immediately obvious keywords for this class.

Below we describe how to incorporate label embeddings into the multi-head self-attention in each encoder layer of BERT. Note that from Figure 1(a), the text sequence $\boldsymbol{t}$ with L tokens, prepended with the token $[CLS]$ as previously described, has embedding matrix $\mathbf{X} = [\boldsymbol{x}_{[CLS]}, \mathbf{X}_w] \in \mathbb{R}^{(L+1)\times D}$, where the $\boldsymbol{x}_{CLS}$ and $\mathbf{X}_w$ are the input embedding vector for $[CLS]$ and matrix for all other input tokens in the sequence, respectively. Subsequently, the query, key and value matrices in each attention head can be represented equivalently to (1) as

$$\begin{aligned}
\mathbf{Q} &= [\boldsymbol{q}_{[CLS]}; \mathbf{Q}_w] = [\boldsymbol{x}_{[CLS]}; \mathbf{X}_w]\mathbf{W}_Q \in \mathbb{R}^{(L+1)\times d}, \\
\mathbf{K} &= [\boldsymbol{k}_{[CLS]}; \mathbf{K}_w] = [\boldsymbol{x}_{[CLS]}; \mathbf{X}_w]\mathbf{W}_K \in \mathbb{R}^{(L+1)\times d}, \\
\mathbf{V} &= [\boldsymbol{v}_{[CLS]}; \mathbf{V}_w] = [\boldsymbol{x}_{[CLS]}; \mathbf{X}_w]\mathbf{W}_V \in \mathbb{R}^{(L+1)\times d},
\end{aligned} \tag{5}$$

where $[\cdot;\cdot]$ denotes row-wise concatenation, and $\{\boldsymbol{q}_{[CLS]}, \boldsymbol{k}_{[CLS]}, \boldsymbol{v}_{[CLS]}\}$ and $\{\mathbf{Q}_w, \mathbf{K}_w, \mathbf{V}_w\}$ are the query, key and value triplets of the $[CLS]$ token and the other input tokens, respectively. From (2), the original attention weights (before softmax) of the augmented sequence can be written as

$$\mathbf{A}_{\mathrm{BERT}} = \frac{1}{\sqrt{d}} \begin{bmatrix} \boldsymbol{q}_{[CLS]}^T \\ \mathbf{Q}_w \end{bmatrix} \begin{bmatrix} \boldsymbol{k}_{[CLS]} & \mathbf{K}_w^T \end{bmatrix} = \frac{1}{\sqrt{d}} \begin{bmatrix} \boldsymbol{q}_{[CLS]}^T \boldsymbol{k}_{[CLS]} & \boldsymbol{q}_{[CLS]}^T \mathbf{K}_w^T \\ \mathbf{Q}_W \boldsymbol{k}_{[CLS]} & \mathbf{Q}_W \mathbf{K}_w^T \end{bmatrix}. \tag{6}$$

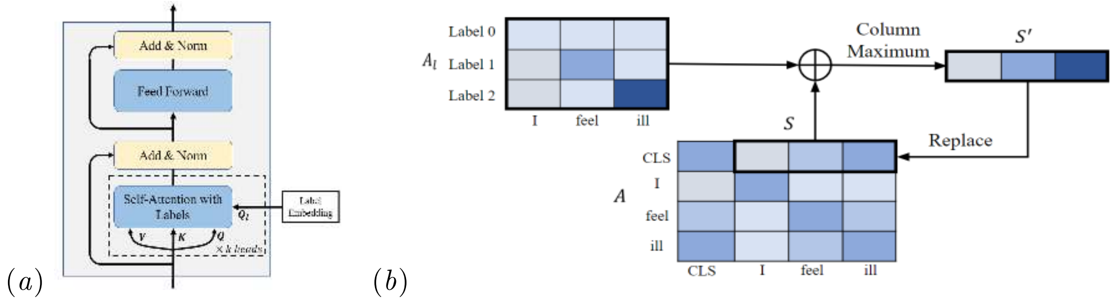


Figure 2: (a) Incorporating label embeddings to the multi-head self-attention in BERT. (b) Modifying self-attention scores with label embeddings. $\bigoplus$ indicates row concatenation.

In $\mathbf{A}_{\text{BERT}}$, which is equivalent to (2), the cross-attention between the $[CLS]$ token and all input tokens is denoted as $\mathbf{S} \triangleq \boldsymbol{q}_{[CLS]}^T \mathbf{K}_w^T \in \mathbb{R}^{1\times L}$.

In LESA-BERT, we introduce label embeddings to boost the model's attention to keywords associated with different labels. We incorporate label embeddings to self-attention in three steps. First (Step 1), we compute the cross attention between the label embeddings and the message tokens

$$\mathbf{Q}_l = \mathbf{X}_l \mathbf{W}_Q \in \mathbb{R}^{3\times d}, \tag{7}$$

$$\mathbf{A}_l = \frac{\mathbf{Q}_l \mathbf{K}_w^T}{\sqrt{d}} \in \mathbb{R}^{3\times L}, \tag{8}$$

where $\mathbf{X}_l \in \mathbb{R}^{3\times D}$ is a matrix containing the three label embeddings, *i.e.*, non-urgent, medium and urgent, which are encoded into label queries $\mathbf{Q}_l$ via the same $\mathbf{W}_Q$ as in (5). Next (Step 2), we compute a modified cross-attention row vector $\mathbf{S}'$ as

$$\mathbf{S}' = \max([\mathbf{S}; \mathbf{A}_l]) \in \mathbb{R}^{1\times L}, \tag{9}$$

where we concatenate $\mathbf{S}$ and $\mathbf{A}_l$ by row and then keep the maximum value of each column. As a result, $\mathbf{S}'$ represents the maximum attention score of a input token with both the $[CLS]$ token and the label embeddings. Finally (Step 3), we obtain the attention weights in LESA-BERT by replacing $\mathbf{S}$ by $\mathbf{S}'$ in (6), thus obtaining

$$\mathbf{A}_{\text{LESA}-\text{BERT}} = \frac{1}{\sqrt{d}} \begin{bmatrix} \boldsymbol{q}_{[CLS]}^T \boldsymbol{k}_{[CLS]} & \mathbf{S}' \\ \mathbf{Q}_W \boldsymbol{k}_{[CLS]} & \mathbf{Q}_W \mathbf{K}_w^T \end{bmatrix}. \tag{10}$$

In (10), when a token is highly relevant to one of the labels it will result in a larger attention score with $[CLS]$ in $\mathbf{S}'$, thus the $[CLS]$ embedding will be less affected by irrelevant information in the sequence, unlike (2) where only attention from the current $[CLS]$ embedding is considered. The proposed attention layer is shown in Figure 2(b). The attention score matrix $\mathbf{A}$ in (2) is replaced as $\mathbf{A}_{\text{LESA}-\text{BERT}}$ in (10). All other components are exactly the same as the original encoders in BERT as shown in (1)–(4).

Note that by incorporating label embeddings to the self-attention layer in BERT model, (9) allows the model to account for label information across all layers. We share the same label embedding for all the layers. The label embedding is adapted to different layers via each

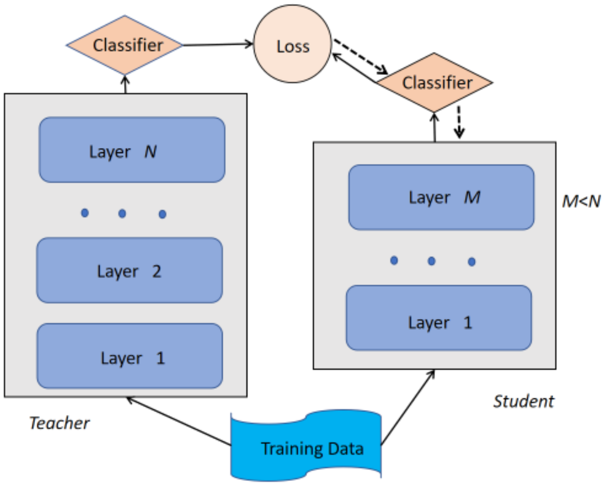


Figure 3: Example of knowledge distillation from a N-layered (teacher) to a M-layered (student) model with $M < N$. In our case, each layer is a LESA-BERT encoder. Solid black lines represent forward computations, whereas dashed lines indicate error back-propagation. The loss function is L_{ce} in (11).

layer's $\mathbf{W}_Q$s in the multi-head attention module. The label embeddings include trainable parameters, which are tuned simultaneously with other parameters in BERT. If keywords for labels are available, then label embeddings are initialized accordingly, otherwise, label embeddings could also be initialized at random (Wang et al., 2018). All other parameters can be initialized from the pre-trained Bert or BioBert model. Though LESA-BERT is motivated by our application, it can be employed for general text classification tasks.

To the best of the authors' knowledge, LESA-BERT is the first work to incorporate label embeddings to perform self-attention in BERT encoders. Note that this technique could also be applied to models that have the self-attention components such as GPT-2, and transformers (Radford et al., 2019).

3.3. Knowledge Transfer with Model Distillation

Knowledge distillation (Buciluă et al., 2006; Hinton et al., 2015) is a compression technique, in which a compact model, called the student model, is trained to reproduce the behavior of a larger teacher model, as in Figure 3. In our experiments, we use a loss function defined over the cross-entropy between the teacher and the student class probabilities, given by:

$$L_{ce} = -\mathrm{softmax}(\boldsymbol{z}^t/T_0)^T \cdot \log \mathrm{softmax}(\boldsymbol{z}^s/T_0), \tag{11}$$

where $\boldsymbol{z}^t$ and $\boldsymbol{z}^s$ are vectors of the output logits from the teacher and student classification networks, respectively, and T_0 is a parameter controlling the degree to which we focus on the class with the highest probability, usually set to 1. When the probabilities $\mathrm{softmax}(\boldsymbol{z}^t/T_0)$ from the teacher model are similar to those of the student model, the cross entropy loss in (11) is small. Therefore, the cross entropy loss measures the difference between output probabilities of teacher and student classifiers.

The knowledge distillation technique is typically used to compress large models such as BERT in order to reduce their computational and memory costs; usually to allow their deployment (Buciluă et al., 2006; Hinton et al., 2015; Sanh et al., 2019; Tang et al., 2019; Strubell et al., 2019). Here, we hypothesize that, beyond these advantages, knowledge distillation can reduce overfitting in the case of small datasets as is our case. Specifically, we expect a smaller generalization error of the distilled model due to its smaller size. According to this hypothesis, we propose a distilled variant of the LESA-BERT: a small BERT model with fewer encoder layers (student) trained to reproduce the behavior of the fine-tuned LESA-BERT (teacher), which has 12 layers. In our experiments, we indeed show the improved performance of the distilled model.

4. Experimental Results

4.1. Cohort

In this work, we utilized 1,756 web portal messages generated from 10/2014 to 08/2018 by adult patients ($>$ 18 years old) of a large academic medical center. The Electronic Health Record (EHR) system (Epic Verona, WI, USA) from Duke University Health System with associated patient portal (MyChart) was the source of all patient messages. A custom-built Application Programming Interface (API) securely made available the portal messages from the EHR enterprise data warehouse into a highly protected virtual network space offered by the medical center. Approved users were allowed access to work with the identifiable protected health information. These messages included free, unstructured plain text sent by patients to their healthcare team. Responses and messages sent from the clinician or health system to the patient were excluded from the analysis. Portal messages were manually labeled by experienced sub-specialty (cardiology) clinicians into three levels of priority: non-urgent, medium and urgent. Non-urgent labels include notes of appreciation (*e.g.*, thank you). The Medium urgency class contains messages that could be reasonably responded to in 1-3 days. Urgent messages are those requiring an immediate phone call to the patient by the clinician. Conditions suggesting acute myocardial infarction, exacerbation of heart failure respiratory distress or possible stroke were labeled as urgent and would be inappropriate for an asynchronous patient portal.

As summarized in Table 1, the data set is imbalanced. Specifically, the total number of messages, 1,756, includes 631 non-urgent, 955 medium and 170 urgent messages. Urgent messages are scarce ($\sim$ 10%), thus the culprit of the imbalance issue. Table 1 also includes a typical example message for each class. For example, an urgent message could be that a patient reports chest pain. In our experiments, the dataset is split into 80% training set ($\sim$ 1.4K) and 20% test set ($\sim$ 0.35K).

4.2. Baselines

To evaluate the performance of LESA-BERT, we compare it to strong baseline classifiers of three kinds: (*i*) traditional machine learning methods: SVM, (*ii*) shallow neural networks: text CNN and Bi-LSTM with attention layer, and (*iii*) pre-trained deep learning models: BERT and BioBERT.

SVM: We utilize an ℓ_2 regularized SVM classifier (Evgeniou et al., 2000), which optimizes the hinge loss function with a ℓ_2 penalty. SVMs are very effective for high-dimensional data (Ghaddar and Naoum-Sawaya, 2018), especially when the number of dimensions is greater than that of samples. SVMs have been widely used for text classification (Tong and Koller, 2001; Dadgar et al., 2016). In our experiments, we implement linear SVM classifiers with the Python module scikit-learn (Pedregosa et al., 2011), and train it via stochastic gradient descent (SGD) with $\ell_2 = 6 \times 10^{-4}$ penalty coefficient. The learning rate is set by Bottou (2010) and the weights are randomly initialized.

Bi-LSTM with Attention: LSTM, short for long short term memory, is one kind of neural network suited for sequential data. In NLP, bidirectional (forward and backward) LSTMs are usually used as feature extractors for sequences of tokens. On top of the Bi-LSTM layer, the attention layer is introduced to capture important words that drive the decisions of the document classification (Yang et al., 2016). The attention weights are further employed to compute the weighted sum of output vectors of Bi-LSTM as the hidden representation for each message. Then, this representation is fed into a fully connected layer that produces the logits for the three labels. The hidden dimension of Bi-LSTM is set to 60, and the max length of each message is set to 256 tokens. We implement Bi-LSTM with attention model in Pytorch (Paszke et al., 2019), and train it using Adam (Kingma and Ba, 2015) with batch size of 8. The learning rate is 0.01 and the weight matrices are initialized with method in He et al. (2015).

Text CNN: Details of text CNN model can be found in Kim (2014). We set the following parameters for our CNN model: the convolution kernel sizes: $\{1, 2, 3, 4, 5\}$, the filter numbers: $\{200, 300, 500, 500, 200\}$, dropout probability: 0.5, and maximum number of tokens in each message is 256. Text CNN also comprises maximum pooling layers, as well as the rectified linear unit (RELU) used as the non-linear activation function. The parameters of the network are learned in mini-batches of size 8 using Adam (Kingma and Ba, 2015). The learning rate is 0.01 and the weight matrices are initialized with method in He et al. (2015).

BERT/BioBERT: For BERT and BioBERT, the base uncased, *i.e.*, all words are treated as lowercase, model is employed in this research. BERT base comprises 12 layers, 768 hidden units, 12 self-attention heads, and 110M parameters. These model are trained with warm-up Adam (Gotmare et al., 2019) setting a batch size of 8 and a learning rate of 3×10^{-5}. The maximum sequence length is set at 256 and other parameters remain the same as the default BERT base configuration. The weight matrices in BERT and BioBERT are set to BERT base uncased (Devlin et al., 2018) and BioBERT v1.1 (Lee et al., 2020), respectively.

4.3. Proposed Models

We briefly cover the configuration of our models, which are implemented with Pytorch in a Python 3 environment.

LESA-BERT: LESA-BERT has the same configuration of BioBERT except for its label embeddings, where embeddings of urgent and medium labels are initialized by the average embeddings of their keywords (see Table 2) while the embeddings of non-urgent labels is randomly initialized.

Distil-LESA-BERT: We distill the fine-tuned LESA-BERT model to two variants: one with 6 encoder layers (Distil-LESA-BERT-6), and the another one with 3 (Distil-LESA-BERT-3). The learning rate for these two variants is 2×10^{-5} and they are initialized from the first 6 or 3 layers of the pre-trained BioBert model accordingly.

To disentangle the effects of distillation and label embeddings, we implement the knowledge distillation without label embeddings with 6 encoder layers, *i.e.*, Distil-BERT-6.

4.4. Feature Choices

We follow the literature to choose features of each message for the classification task (Sulieman et al., 2017). For traditional classifiers like SVM, TF-IDF features are commonly used. For shallow neural networks like TextCNN and Bi-LSTM with attention, word embeddings such as Word2vec or GloVe are utilized as input features. For BERT-based deep learning models, we just tokenize each message then the model initializes them from pre-training embeddings (Devlin et al., 2018).

For the SVM classifier, we use TF-IDF vectors (Salton and Buckley, 1988) of each message as features. TF represents the frequency of a certain token in a message and IDF represents the inverse of the number of messages in which this token has appeared. The product of TF and IDF for a token is a score that represents the importance of the token in a message. TF-IDF features have been shown to be very effective in text classification (Zhang et al., 2008).

For shallow neural network classifiers like CNN or LSTM models, instead of using TF-IDF, we adopt pre-trained GloVe word vectors (Pennington et al., 2014; Si et al., 2019) as features for each message. We use the 100 dimensional GloVe word vectors pre-trained on 6 billion tokens from Wikipedia and Gigaword data sets.

For the BERT-based models including BERT, BioBERT and LESA-BERT, we use the wordpiece tokenizer (Wu et al., 2016) and keep the models' pre-trained token embeddings as initialization, so that the input embeddings are compatible with the models' pre-trained parameters. We utilize the BERT base uncased model with 12 layer encoders and 12 attention heads and 768 dimensional embeddings, which was pre-trained on BookCorpus (Zhu et al., 2015) and English Wikipedia. BioBERT shares the same model configuration as BERT base, but was further pre-trained on biomedical corpora (PubMed abstracts and PMC full-text articles). Parameters in LESA-BERT except the label embeddings are initialized from the pretrained BioBert model.

4.5. Evaluation Metrics

For classification tasks, commonly utilized metric criteria are precision, recall, F1 score and Area Under the Curve (AUC) score. Oftentimes a classifier has a good precision with a poor recall, thus F1 score provides a good balance of precision and recall. AUC is typically employed for binary classifiers. Since the message urgency dataset comprises multiple (three) classes and is imbalanced, we find that macro average F1 score as the most suitable evaluation criteria (Parambath et al., 2014). The F1 score is given by the harmonic mean of precision and recall. Accuracy is a poor metric in this situation because it encourages the model to focus on the majority class. Similarly, micro-level metrics like

Table 3: Performance metrics of different classifiers on the patient messages data.

Model	Macro F1	Macro Precision	Macro Recall
SVM	0.748 ± 0.007	0.795 ± 0.007	0.731 ± 0.006
TextCNN	0.754 ± 0.020	0.772 ± 0.031	0.749 ± 0.031
Bi-LSTM Attention	0.761 ± 0.016	0.758 ± 0.016	0.769 ± 0.021
BERT	0.761 ± 0.021	0.762 ± 0.019	0.761 ± 0.024
BioBERT	0.764 ± 0.010	0.774 ± 0.015	0.758 ± 0.009
Distil-BERT-6	0.754 ± 0.010	0.742 ± 0.010	0.787 ± 0.007
LESA-BERT	**0.789 ± 0.011**	0.784 ± 0.010	0.797 ± 0.014
Distil-LESA-BERT-6	**0.807 ± 0.009**	0.816 ± 0.004	0.798 ± 0.024
Distil-LESA-BERT-3	**0.780 ± 0.017**	0.768 ± 0.016	0.816 ± 0.015

micro F1 score are not preferred because they are weighted by the number of cases in each label, which makes them favor the majority classes.

We also evaluate the advantages of the distillation process. We distill the fine-tuned LESA-BERT model to 6-layered and 3-layered versions, and evaluate the number of parameters and the inference time required for a full pass on the test set on a CPU with batch size of 1, which are commonly used evaluation criteria for distillation (Sanh et al., 2019).

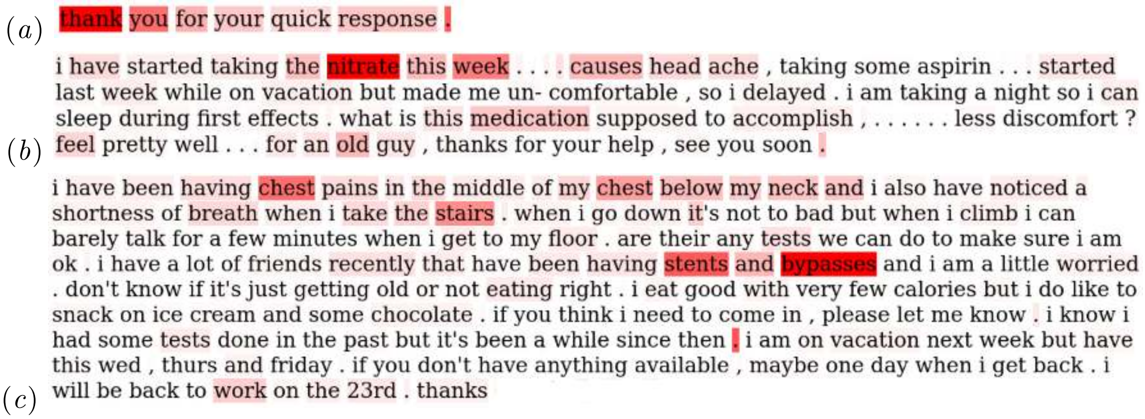
4.6. Quantitative Results

Table 3 presents quantitative results for various classifiers on the test set data. Mean and standard errors are computed based on 5 different random seeds for the SGD/Adam training algorithm. The F1 scores for shallow neural networks, *i.e.*, TextCNN and Bi-LSTM attention, are slightly better than SVM, which might be owed to the benefits of semantic features captured by GloVe word vectors. BERT-based deep neural networks such as BERT and BioBERT only marginally outperforms Bi-LSTM attention and TextCNN, which is likely caused by the small size (~ 1.4K) of the training data. The F1 score of BioBERT is 0.3% higher than that of BERT, which means that pre-training on large biomedical corpora transfers extra beneficial information to BERT model. In terms of macro average F1 score, our LESA-BERT and its distilled variants achieve higher F1 scores compared with other classifiers. The performance gains can be attributed to the addition of the label embeddings.

The model Distil-LESA-BERT-6 provides higher F1 score (0.807) than that of the full 12-layered LESA-BERT. Possibly, this implies that the full model overfits the training set, while the distilled model with 6-layers reduces the overfitting. Distil-LESA-BERT-3 model, which comprises only 3 layers provides F1 score of 0.78, slightly lower than the full LESA-BERT model. The Distil-BERT-6 model uses knowledge distillation when fine-tuning on downstream datasets without label embeddings, and its Macro F1 is 0.754, lower than the original BERT model. This fact shows the effectiveness of label embeddings on the BERT model.

Table 4: Number of parameters and inference times on the patient messages data.

Model	Number of parameters (Millions)	Inference time (Seconds)
LESA-BERT	110 (×1.0)	212.6 (×1.0)
Distil-LESA-BERT-6	66 (×0.6)	79.8 (×0.38)
Distil-LESA-BERT-3	44 (×0.4)	40.8 (×0.19)



thank you for your quick response !

i have started taking the nitrate this week . . . . causes head ache , taking some aspirin . . . started last week while on vacation but made me un- comfortable , so i delayed . i am taking a night so i can sleep during first effects . what is this medication supposed to accomplish , ... less discomfort ? feel pretty well . . . for an old guy , thanks for your help , see you soon .

i have been having chest pains in the middle of my chest below my neck and i also have noticed a shortness of breath when i take the stairs . when i go down it's not to bad but when i climb i can barely talk for a few minutes when i get to my floor . are their any tests we can do to make sure i am ok . i have a lot of friends recently that have been having stents and bypasses and i am a little worried . don't know if it's just getting old or not eating right . i eat good with very few calories but i do like to snack on ice cream and some chocolate . if you think i need to come in , please let me know . i know i had some tests done in the past but it's been a while since then . i am on vacation next week but have this wed , thurs and friday . if you don't have anything available , maybe one day when i get back . i will be back to work on the 23rd . thanks

Figure 4: Visualization of learned LESA-BERT attention scores for tokens in messages in test data. One example from each label: (a) non-urgent, (b) medium, and (c) urgent.

4.7. Computational Cost

Table 4 presents the number of parameters in LESA-BERT and its distilled variants, as well as inference times for a full pass over the test set with batch size of 1. As the number of layers in LESA-BERT decreases, the number of parameters and inference time reduces dramatically. Distilling the original 12-layered LESA-BERT to 6 layers reduces the number of parameters by 40% and accelerates the inference by 2.7 times. Distil-LESA-BERT-3 has fewer parameters and faster inference speed than Distil-LESA-BERT-6. Taking into account their F1 scores and standard errors, Distil-LESA-BERT-6 and Distil-LESA-BERT-3 match performance of the 12-layer LESA-BERT. Distil-LESA-BERT-6 provides the best F1 score compared to all other methods by F1 score. However, in the case that inference time is preferred, Distil-LESA-BERT-3 will be the top choice.

4.8. Qualitative Results

In Figure 4, we visualize how the obtained representation by LESA-BERT attend to different tokens in three example messages, one for each class. Tokens are shaded in terms of their importance for classification. Dark-colored tokens are more prominently weighted (attended) when constructing the message embedding. As seen in Figure 4, attention scores can identify relevant keywords in all three examples. We choose these three examples because they are representative in their respective categories.

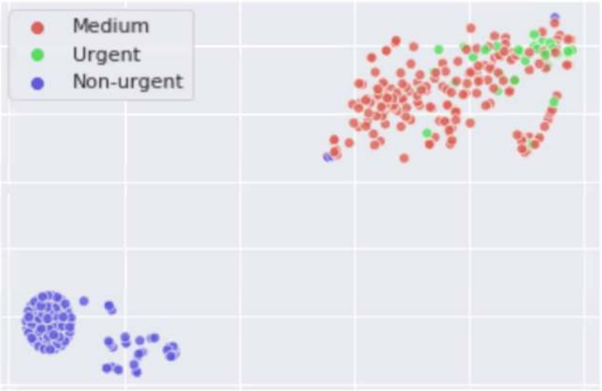


Figure 5: t-SNE representation of $[CLS]$ tokens for messages grouped by class.

Figure 5 shows the 2-dimensional t-SNE plot (Maaten and Hinton, 2008) of the latent code of each message in the test set, *i.e.*, the embedding of $[CLS]$ token in the output encoder of LESA-BERT. Each point corresponds to a single message and the colors represent the different labels. From this figure, non-urgent messages (purple dots) are well separated from messages of the other two classes. For the urgent and medium classes, their data points have significant overlap, meaning that urgent and medium messages, not surprising, have similar characteristics (features). This may be in part caused by the subjectiveness of the labeling process as the definitions of urgent and medium messages are not exactly clear.

5. Discussion

We have proposed a framework for building classifiers on small and imbalanced datasets, which is a commons scenario in healthcare. Our method builds upon a large deep learning model, BioBERT, which was pre-trained on huge general-domain and biomedical corpora. We developed LESA-BERT, which has a novel self-attention architecture that can incorporate label embeddings to boost the model's capacity for attention. We fine-tuned LESA-BERT on the target dataset by initializing its parameters from BioBERT. Subsequently, we distilled the fine-tuned 12-layered LESA-BERT to 6-layered or 3-layered variants, and found that the 6-layered Distil-LESA-BERT outperforms the 12-layered LESA-BERT. Therefore, knowledge distillation is not only a tool for model compression, but can also be used to reduce overfitting. We demonstrated the application of our framework on a real healthcare dataset –*message urgency*– and built a message triage classification model. Our methods outperformed baseline classifiers by a significant margin. Our technical solution can be easily applied to other clinical datasets.

Clinical Implications We built an automatic message triage model that can predict the priority of patient portal messages based on their content. This version of message triage could be further improved with a larger dataset to train our classifiers. The message triage system is of real impact to healthcare providers and potentially save valuable working hours by freeing them from reading patient portal messages in chronological order. Promptly accessing and realizing urgent messages can impact both patient safety as well as clinician workflow. After the message triage classifier produces the priority of messages, healthcare providers can take further actions based on their existing workflow. Further, they may want to utilize automatic responses or templates to answer non-urgent messages. Medium or time-sensitive messages may either be directed to emergency rooms, urgent care or telephone encounters.

Limitations Although our methods outperform a wide selection of baseline models, it can still produce some mistakes from a human perspective. For instance, Figure 5 shows that two non-urgent messages (purple dots) fall in the region of medium and urgent labels. In fact, we have investigated these very few messages that are close to medium or urgent messages, and one example reads "No chest pain at all". This message has very strong keywords, chest pain, which usually appears in urgent messages. Therefore, the LESA-BERT model maps this message to urgent class by ignoring the negation word "No". Many researchers have found that deep learning models are powerful but still easy to fool (Moosavi-Dezfooli et al., 2016; Heaven, 2019). As future work we plan on incorporating negation detection (Qian et al., 2016) into our classification framework.

Our current message triage system has a few limitations from the clinical perspective. Firstly, our message classifier only has three levels of urgency, which could be further refined into more granular categories. After passing patient portal messages through our urgency classifier, urgent messages will be treated with priority, but non-urgent and medium ones will probably need further automatic classifiers, such as message intention detector, *etc.* Secondly, our current work is limited to only utilizing the content of portal messages. One possible follow-up research could be to incorporate patient's demographic information or even comorbidities. For example, age can be quite important when deciding the priority of messages. Messages from patients aged over 65 should receive relatively more immediate attention than younger patients. Comorbidities could also be taken into account as patients with heart attacks, for example, usually get more intense care than patients with, for example, common respiratory infections.

Acknowledgments

This work was supported in part by Duke Institute for Health Innovation pilot funding.

References

Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1425–1438, 2015.

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

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

Jinying Chen, John Lalor, Weisong Liu, Emily Druhl, Edgard Granillo, Varsha G Vimalananda, and Hong Yu. Detecting hypoglycemia incidents reported in patients' secure messages: Using cost-sensitive learning and oversampling to reduce data imbalance. *Journal of medical Internet research*, 21(3):e11990, 2019.

Robert M Cronin, Daniel Fabbri, Joshua C Denny, and Gretchen Purcell Jackson. Automated classification of consumer health information needs in patient portal messages. In *AMIA Annual Symposium Proceedings*, volume 2015, page 1861. American Medical Informatics Association, 2015.

Robert M Cronin, Daniel Fabbri, Joshua C Denny, S Trent Rosenbloom, and Gretchen Purcell Jackson. A comparison of rule-based and machine learning approaches for classifying patient portal messages. *International journal of medical informatics*, 105:110–120, 2017.

Seyyed Mohammad Hossein Dadgar, Mohammad Shirzad Araghi, and Morteza Mastery Farahani. A novel text mining approach based on tf-idf and support vector machine for news classification. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)*, pages 112–116. IEEE, 2016.

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

Theodoros Evgeniou, Massimiliano Pontil, and Tomaso Poggio. Regularization networks and support vector machines. *Advances in computational mathematics*, 13(1):1, 2000.

Bissan Ghaddar and Joe Naoum-Sawaya. High dimensional data classification and feature selection using support vector machines. *European Journal of Operational Research*, 265 (3):993–1004, 2018.

Caroline Lubick Goldzweig, Greg Orshansky, Neil M Paige, Ali Alexander Towfigh, David A Haggstrom, Isomi Miake-Lye, Jessica M Beroes, and Paul G Shekelle. Electronic patient portals: evidence on health outcomes, satisfaction, efficiency, and attitudes: a systematic review. *Annals of internal medicine*, 159(10):677–687, 2013.

Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=r14EOsCqKX.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Douglas Heaven. Why deep-learning ais are so easy to fool. *Nature*, 574(7777):163, 2019.

Jennifer L Hefner, Sarah R MacEwan, Alison Biltz, and Cynthia J Sieck. Patient portal messaging for care coordination: a qualitative study of perspectives of experienced users with chronic conditions. *BMC family practice*, 20(1):57, 2019.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

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

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751, 2014. URL http://aclweb.org/anthology/D/D14/D14-1181.pdf.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Ryan Kiros, Ruslan Salakhutdinov, and Rich Zemel. Multimodal neural language models. In *International conference on machine learning*, pages 595–603, 2014.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240, 2020.

Xirong Li, Shuai Liao, Weiyu Lan, Xiaoyong Du, and Gang Yang. Zero-shot image tagging by hierarchical semantic embedding. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 879–882, 2015.

Yukun Ma, Erik Cambria, and Sa Gao. Label embedding for zero-shot fine-grained named entity typing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 171–180, 2016.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

Oznur Esra Para, Ebru Akcpinar Sezer, and Hayri Sever. Clinical decision support systems: From the perspective of small and imbalanced data set. *Health Informatics Vision: From Data via Information to Knowledge*, 262:344, 2019.

Shameem Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. Optimizing f-measures by cost-sensitive classification. In *Advances in Neural Information Processing Systems*, pages 2123–2131, 2014.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8024–8035, 2019.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

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.

Zhong Qian, Peifeng Li, Qiaoming Zhu, Guodong Zhou, Zhunchen Luo, and Wei Luo. Speculation and negation scope detection via convolutional neural networks. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 815–825, 2016.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Alexandra Ramsey, Erin Lanzo, Hattie Huston-Paterson, Kathy Tomaszewski, and Maria Trent. Increasing patient portal usage: preliminary outcomes from the mychart genius project. *Journal of Adolescent Health*, 62(1):29–35, 2018.

Jose Antonio Rodriguez-Serrano and Florent C Perronnin. Label-embedding for text recognition, April 14 2015. US Patent 9,008,429.

Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Shijing Si, Weiguo Zheng, Liuyang Zhou, and Mei Zhang. Sentence similarity computation in question answering robot. In *Journal of Physics: Conference Series*, volume 1237, page 022093. IOP Publishing, 2019.

Cynthia J Sieck, Jennifer L Hefner, Jeanette Schnierle, Hannah Florian, Aradhna Agarwal, Kristen Rundell, and Ann Scheck McAlearney. The rules of engagement: perspectives on secure messaging from experienced ambulatory patient portal users. *JMIR medical informatics*, 5(3):e13, 2017.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

Lina Sulieman, David Gilmore, Christi French, Robert M Cronin, Gretchen Purcell Jackson, Matthew Russell, and Daniel Fabbri. Classifying patient portal messages using convolutional neural networks. *Journal of biomedical informatics*, 74:59–70, 2017.

Ahmad P Tafti, Sunyang Fu, Aditya Khurana, George M Mastorakos, Kenneth G Poole, Stephen J Traub, James A Yiannias, and Hongfang Liu. Artificial intelligence to organize patient portal messages: a journey from an ensemble deep learning text classification to rule-based named entity recognition. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1380–1387. IEEE, 2019.

Raphael Tang, Yao Lu, Linqing Liu, Lili Mou, Olga Vechtomova, and Jimmy Lin. Distilling task-specific knowledge from bert into simple neural networks. *arXiv preprint arXiv:1903.12136*, 2019.

Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

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.

Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. Joint embedding of words and labels for text classification. In *ACL*, 2018.

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.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. Multi-task label embedding for text classification. *arXiv preprint arXiv:1710.07210*, 2017.

Wen Zhang, Taketoshi Yoshida, and Xijin Tang. Tfidf, lsi and multi-word in information retrieval and text categorization. In *2008 IEEE International Conference on Systems, Man and Cybernetics*, pages 108–113. IEEE, 2008.

Yang Zhao, Zoie Shui-Yee Wong, and Kwok Leung Tsui. A framework of rebalancing imbalanced healthcare data for rare events' classification: a case of look-alike sound-alike mix-up incident detection. *Journal of healthcare engineering*, 2018, 2018.

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.