BMC Bioinformatics

RESEARCH

Open Access

Multi-objective data enhancement for deep learning-based ultrasound analysis

Chengkai Piao[1], Mengyue Lv[2], Shujie Wang[2], Rongyan Zhou[2], Yuchen Wang[1], Jinmao Wei[1*] and Jian Liu[1*]

*Correspondence:
weijm@nankai.edu.cn;
jianliu@nankai.edu.cn

[1] College of Computer Science, Nankai University, Tianjin, China
[2] Department of Ultrasound, Cangzhou Municipal Haixing Hospital, Cangzhou, China

Abstract

Recently, Deep Learning based automatic generation of treatment recommendation has been attracting much attention. However, medical datasets are usually small, which may lead to over-fitting and inferior performances of deep learning models. In this paper, we propose multi-objective data enhancement method to indirectly scale up the medical data to avoid over-fitting and generate high quantity treatment recommendations. Specifically, we define a main and several auxiliary tasks on the same dataset and train a specific model for each of these tasks to learn different aspects of knowledge in limited data scale. Meanwhile, a Soft Parameter Sharing method is exploited to share learned knowledge among models. By sharing the knowledge learned by auxiliary tasks to the main task, the proposed method can take different semantic distributions into account during the training process of the main task. We collected an ultrasound dataset of thyroid nodules that contains Findings, Impressions and Treatment Recommendations labeled by professional doctors. We conducted various experiments on the dataset to validate the proposed method and justified its better performance than existing methods.

Keywords: Multi-objective, Parameter sharing, Ultrasound analysis, Deep learning, Thyroid nodules

Introduction

Ultrasound Analysis, as one of the most commonly used examination methods, has been recognized as a powerful screening and diagnostic tool for physicians and radiologists [1, 2]. Recently, Deep Learning (DL) based Automatic Ultrasound Analysis (AUA) [3] methods, such as Disease Screening (DS) [4–6], Lesion Detection (LD) [7–9], Automatic Diagnosis (AG) [10–12], etc, have attracted attention from academics and practitioners [13]. As a powerful auxiliary tool to analyze the conditions of patients, AUA methods can help doctors to reduce their workloads.

In applications, the demands of medical workers for AUA also include Automatic Treatment Recommendation (ATR), a less studied field that automatically generates treatment recommendations without the intervention of doctors. Researchers proposed many DL models to address ATR tasks, which can be used to replace the doctor's work to some extent [14–19]. These models can be categorized into internal and external

methods. Internal methods adopted elaborately designed backbone structures to extract informative features. By virtue of powerful structures and skillful training tricks, such as deep structures, residual connections, dropout layers, etc, internal methods could extract representative features and soon achieved the dominant position [20–23]. External methods adopted a shared training policy that pre-trains the model on public datasets firstly, and then fine-tune its parameters on the ultrasound dataset. With the help of public knowledge, external methods can ensure the model to be sufficiently trained [24–27].

However, there are two limitations in existing methods, insufficient training data and mismatched knowledge fields [28]. On the one hand, the main way to improve the performances of internal methods is to design more complex structures and use larger training sets. But it is difficult to acquire labeled large-scale medical datasets due to the high labeling costs, which will constrain the performances of internal models. On the other hand, ultrasound data are full of professional knowledge and may not have the same semantic distribution with pre-training datasets. Furthermore, how to build a highly related pre-training dataset for the ATR task is also a problem.

To address the aforementioned limitations, we propose the Multi-Objective Data Enhancement (MODE) method, which is capable of expanding the limited dataset without extra labeling costs and external datasets. In addition, we present an ultrasound dataset of thyroid nodules, which contains not only Findings and Impressions, the results that average ultrasound reports should have, but also Treatment Recommendations and Severity Scores labeled by clinicians, to validate the feasibility of MODE.

Specifically, we define a main task and two auxiliary tasks on the ultrasound dataset, each task has its own training objective and model. The main task is the ATR task that generates Treatment Recommendations according to Findings and Impressions, we construct a Transformer model to handle this task. The First auxiliary task is a Summary task that generates Impressions for given Findings, we construct a Long-Short term Memory (LSTM) model to handle this task. The second auxiliary task is a Regression task that computes Severity Scores according to Findings and Impressions, we construct a Convolution Neural Network (CNN) model to handle this task. Meanwhile, we present the Soft Parameter Sharing (SPS) method specially for sharing the learned knowledge from the two completely different types of auxiliary tasks to the main task. In the training process, we first train the two auxiliary tasks to learn different aspects of the ultrasound data, and then train the main model with the learned auxiliary knowledge as prior information.

With the presented ultrasound dataset as an example, Fig. 1 illustrates the comparison between MODE and existing methods. In Fig. 1a, an internal method directly trains the model to generate the Treatment Recommendations, which may suffer from the limitation of insufficient training samples. In Fig. 1b, an external method pre-trains the model on public datasets to avoid over-fitting, but the learned common knowledge may not be compatible with the ultrasound dataset. In Fig. 1c, we define multiple tasks on the same dataset and share knowledge among models. For the first limitation, the multiple training objectives can force the models to learn different aspects of knowledge from the same dataset, and to indirectly increase the training samples. For the second limitation, the parameter sharing method can be used to share learned knowledge of auxiliary tasks

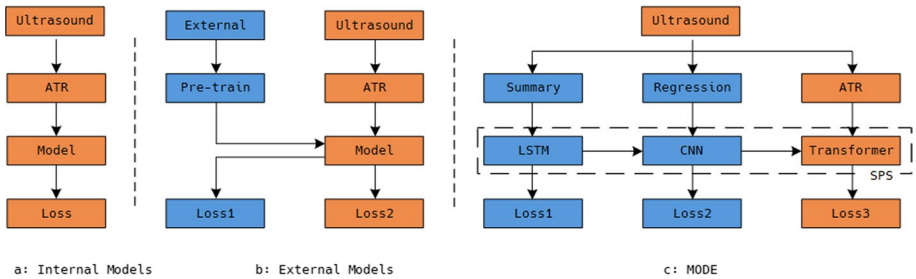

Fig. 1 The comparisons between MODE and existing methods, in which orange blocks denote the main tasks and blue blocks are auxiliary tasks

to the main task, to function like external datasets. Consequently, the proposed framework can fully optimize their parameters within less training samples and little external knowledge.

The main contributions of this paper are listed below.

- A Multi-objective data enhancement method which can expanding the ultrasound dataset without extra labeling costs and external datasets.
- A Soft Parameter Sharing method used to share the learned knowledge among models.
- An ultrasound dataset of thyroid nodules, which contains Findings, Impressions, Treatment Recommendations and Severity Scores.

The rest of this paper is organized as follows. The second section briefly introduces the related work of this paper. In the third section, we introduce and analyze the details of MODE. In the fourth part, the method is applied to ultrasonic dataset, and the experimental results are analyzed in detail. Finally, we summarize the work of this paper in the fifth part.

Related work

DL models have deeply influenced some areas of medical informatics, especially NLP-based tasks [29]. Researchers proposed many DL models to handle different kinds of medical tasks, such as Automatic Diagnosis, Disease Screening, Lesion Detection (LD), etc. ATR is a less studied area that needs both elaborately designed structures and enough training samples. According to the demand of input–output formats and model structures, two kinds of methods, viz. internal and external methods, can be used to solve ATR problems.

Complying with the former category, some researchers held the opinion that extracting abundant and high-quality semantic features are key factors [30], thus they have worked on representation models for a long time and proposed many elaborately designed structures and training tricks [31]. Specifically, [32] proposed an LSTM based model to identify text order of medical data. Borjali et al. [33] proposed a DL-NLP model for efficient and accurate hip dislocation medical adverse events detection. Liu et al. [34] proposed to use hierarchical CNN and LSTM models to handle negations and numerical values that exist in medical text. Prabhakar et al. [35] proposed to use quad-channel features to enhance the performance of LSTM.

In the latter domain, knowledge distribution is regarded as the engine room of DL models, and it will be more solid with a larger scale dataset. Following this idea, researchers used extra datasets to pre-train the model and to expand its knowledge field. Rebane et al. [36] used large scale diagnosis and drug records as the external dataset to instruct the process of medical knowledge extraction. Qin et al. [37] proposed an orthogonal wrapper to enhance the differences between datasets and thus to extract distinguishable and informative features. To take full advantage of external knowledge, a set of super-large scale Internet data was pre-trained firstly to learn word embeddings, and then a fine-tuning stage is adopted to take advantage of these representation vectors to fit specific data [38, 39].

Generally, deep and complex structures have stronger feature-selecting abilities. But in medical related tasks, insufficient training samples may be an obstacle for internal models to understanding the professional knowledge, and lead to over-fitting. Using external dataset to pre-train the model is an intuitive method to address the over-fitting problem since it can provide abundant training samples. However, it may lead to mismatched semantic distributions if we use non-related common datasets in the pre-training processes.

We present a data enhancing method, MODE, to scale up the ultrasound dataset without extra labeling costs. We define a main task and two auxiliary tasks on the ultrasound dataset, each task has its own training objective and model. According to the distinctive training objectives, auxiliary models can learn different aspects of semantic distribution. For example, Findings are descriptions of examination results, Impressions are corresponding conclusions. The Summary task is then defined to help the model learn the relations between lesions and clinical symptoms. The Regression task is trained to select the most related factor to identify the disease. With the help of the SPS method, the learned knowledge of auxiliary models corresponding to different aspects of ultrasound analysis can be concentrated to the main task, to indirectly scale up the dataset and provide solid semantic distribution. Consequently, MODE is capable of enlarging the scale of professional datasets with the same domain instances.

Multi-objective data enhancement

We introduce the structure of MODE as well as the main and auxiliary tasks in "The structure of MODE"–"Using transformer to handle the ATR task" sections. We propose the SPS method in "Soft parameter sharing" section, which is used to share the learned knowledge of auxiliary tasks to the main task. Finally we present the training process of MODE in "Training process" section.

The structure of MODE

The main idea of MODE is to reuse limited dataset by defining multiple tasks. Since all training objectives are defined on the ultrasound dataset, the MODE does not use any external data to train the models. Specifically, we define a main and two auxiliary tasks on the proposed ultrasound dataset, each of which has its own training objective. The main task, ATR, is a generation task that transforms Finding and Impression into Treatment Recommendation. The first auxiliary task is a Summary task that transforms Finding into Impression, the second auxiliary task is a Regression task that computes the Severity Score

according to Finding and Impression. By taking the learned knowledge of auxiliary tasks into account, the main task can learn different aspect of knowledge and achieve better performance.

We train a specific DL model for each task respectively and propose the SPS method to share knowledge among models. Figure 2 illustrates the topology of MODE, in which orange and blue blocks, as well as corresponding lines, represent the main and auxiliary tasks. As shown in Fig. 2a, we use an encoder–decoder LSTM to handle the Summary task since it can satisfy the input–output data format that converts Findings into Impressions. As shown in Fig. 2b, we use a CNN model to handle Severity task since it can satisfy the input–output data format transformation that converts Findings and Impressions into Severity Scores. As shown in Fig. 2c, we use an encoder–decoder Transformer to handle ATR task since it can satisfy the input–output data format transformation that converts Findings and Impressions into Treatment Recommendations.

Using long-short term memory to handle the summary task

Since LSTM is a powerful model to detect the sequential information, we use a Bi-direction LSTM model to handle the Summary task. Equation (1) illustrates the computing process of the Summary task.

$$\begin{aligned} \boldsymbol{Z} &= [LSTM(\boldsymbol{F}), LSTM(inv(\boldsymbol{F}))]_c, \\ \boldsymbol{I} &= FFN(LSTM(\boldsymbol{z})). \end{aligned} \tag{1}$$

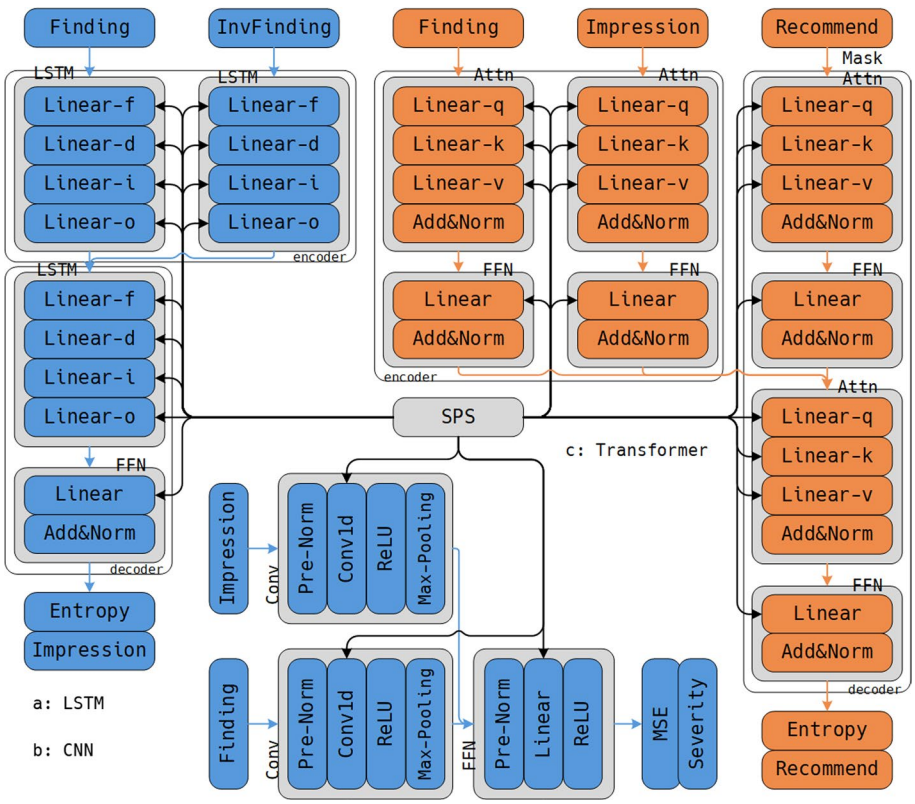

Fig. 2 The structure of MODE

where $\boldsymbol{F} \in \mathbb{R}^{|F| \times n}$ is the Finding text, $inv(\boldsymbol{F})$ denotes the inverse order of $\boldsymbol{F}$, $\boldsymbol{Z}$ is the intermediate variable, $\boldsymbol{I} \in \mathbb{R}^{|I| \times n}$ is the Impression text, "$[*]_c$" denotes the concatenate operation in the channel dimension, $|F|$ and $|I|$ denote text lengths.

An LSTM module adopts recurrent steps to iteratively load and save historical information to update its hidden states. Equation (2) illustrates the updating process of LSTM.

$$\boldsymbol{c}^t, \boldsymbol{h}^t = LSTM(\boldsymbol{c}^{t-1}, \boldsymbol{h}^{t-1}, \boldsymbol{X}_t). \tag{2}$$

where $\boldsymbol{h}$ and $\boldsymbol{c} \in \mathbb{R}^n$ are hidden and cell states, superscripts denote the recurrent index, $\boldsymbol{X}_t \in \mathbb{R}^n$ is the target word of the t th recurrent step.

In the recurrent process, LSTM trains a group of gates to control what information of $\boldsymbol{X}_t$ should be add into $\boldsymbol{c}^t$ and $\boldsymbol{h}^t$. Equation (3) illustrates the definitions of gates and the states updating process.

$$\begin{aligned}
\boldsymbol{g}_f^t &= \boldsymbol{X}_t \times \boldsymbol{W}_f + \boldsymbol{h}^{t-1} \times \boldsymbol{U}_f + \boldsymbol{b}_f, \\
\boldsymbol{g}_d^t &= \boldsymbol{X}_t \times \boldsymbol{W}_d + \boldsymbol{h}^{t-1} \times \boldsymbol{U}_d + \boldsymbol{b}_d, \\
\boldsymbol{g}_i^t &= \boldsymbol{X}_t \times \boldsymbol{W}_i + \boldsymbol{h}^{t-1} \times \boldsymbol{U}_i + \boldsymbol{b}_i, \\
\boldsymbol{g}_o^t &= \boldsymbol{X}_t \times \boldsymbol{W}_o + \boldsymbol{h}^{t-1} \times \boldsymbol{U}_o + \boldsymbol{b}_o, \\
\boldsymbol{c}^t &= \boldsymbol{c}^{t-1} \cdot \boldsymbol{g}_f^t + \boldsymbol{g}_d^t \cdot \boldsymbol{g}_i^t, \\
\boldsymbol{h}^t &= \boldsymbol{g}_o^t \cdot tanh(\boldsymbol{h}^t).
\end{aligned} \tag{3}$$

where $\boldsymbol{g}_f^t, \boldsymbol{g}_d^t, \boldsymbol{g}_i^t, \boldsymbol{g}_o^t \in \mathbb{R}^n$ are LSTM gates, $\boldsymbol{W}_*, \boldsymbol{U}_* \in \mathbb{R}^{n,n}$ are trainable parameters, $\boldsymbol{b}_* \in \mathbb{R}^n$ are bias, $\times$ denotes matrix multiplication, "$\cdot$" denotes Hadamard Product.

Although an LSTM model has such a complex computing process, each of its gates can be viewed as the combination of two linear modules. Equation (4) provides an equivalent implementation of an LSTM gate in Eq. (3).

$$\boldsymbol{g}^t = Linear_W(\boldsymbol{x}^t) + Linear_U(\boldsymbol{x}^t). \tag{4}$$

In "Soft parameter sharing" section, we propose the SPS method to share the learned knowledge of the LSTM model to the main task.

Using convolution neural network to handle the severity task

CNN is a powerful model which is good at transforming sequential data into a single value, which can be used to handle the Severity task. We use two CNN modules to extract important features in Findings and Impressions respectively. Equation (5) illustrates the computing process of CNN.

$$s = FFN([Conv1d(\boldsymbol{F}), Conv1d(\boldsymbol{I})]_t). \tag{5}$$

where s is the Severity Score of the ultrasound report, "$[*]_t$" denotes the concatenation operation in the temporal dimension.

Obviously, a CNN model mainly contains convolution and linear modules. In "Soft parameter sharing" section, we propose the SPS method to share the learned knowledge of the CNN model to the main task.

Using transformer to handle the ATR task

We use a Transformer model to generate Treatment Recommendation from Finding and Impression since it is good at searching relations among sequences. Equation (6) illustrates the computing process of Transformer.

$$\begin{aligned} \boldsymbol{Z} &= [FFN(SelfAttn(\boldsymbol{F})), FFN(SelfAttn(\boldsymbol{I}))]_t, \\ \boldsymbol{T} &= FFN(Attn(FFN(MaskAttn(\boldsymbol{T})), \boldsymbol{Z}, \boldsymbol{Z})). \end{aligned} \tag{6}$$

where $SelfAttn(\boldsymbol{F}) = Attn(\boldsymbol{F}, \boldsymbol{F}, \boldsymbol{F})$ is a special version of Attention module, $\boldsymbol{Z}$ is intermediate variable, $MaskAttn()$ has the same structure of $SelfAttn$ except it uses a mask attention matrix to prevent the module from seeing future words, $\boldsymbol{T}$ is Treatment Recommendation. Equation (7) illiterates the definition of $Attn$.

$$Attn(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^{\mathbb{T}}}{\sqrt{n}}) \times \boldsymbol{V}. \tag{7}$$

Similar to LSTM and CNN, a Transformer model is mainly controlled by three linear modules. Equation (8) illustrates the relation between linear module and the Attention.

$$\begin{aligned} \boldsymbol{Q} &= Linear_Q(\boldsymbol{X}), \\ \boldsymbol{K} &= Linear_K(\boldsymbol{X}), \\ \boldsymbol{V} &= Linear_V(\boldsymbol{X}). \end{aligned} \tag{8}$$

where $\boldsymbol{X}$ is input data.

In "Soft parameter sharing" section, we propose the SPS method to share the learned knowledge of the Transformer model to the main task.

Soft parameter sharing

To enable the main task to take different aspects of the ultrasound dataset in to account, we need a cross-model parameter sharing method to share the learned knowledge of auxiliary tasks to the main task. As we discussed earlier, although models have different structures, their fundamental bricks are CNN and Linear blocks. Considering that CNN and Linear blocks incorporate fixed size matrices as their parameters, it is plausible to indirectly share knowledge across models by letting all modules use the same parameter matrix. However, it is difficult to use a fixed size matrix as their parameters since the models have different parameter shapes, a CNN model needs 3-dimension matrices as its kernels, but a Linear model usually uses 2 dimension parameter matrices. To address this problem, we propose the Soft Parameter Sharing (SPS) method, which is capable of transforming the global shared parameter matrix into different parameter matrices.

SPS is a CNN based algorithm that utilizes the unsymmetrical character of convolution operations to tailor matrix into specific shapes. First, we define a parameter matrix, called template, as the global parameters. Then, we assign a SPS kernel to each CNN, or Linear, module. Last, we compute the shared parameters through Eq. (9), and use this parameter matrix to replace the original parameters.

$$SPS(\boldsymbol{M}) = \sigma(\boldsymbol{E} \otimes \boldsymbol{M} + \boldsymbol{b}). \tag{9}$$

where $\boldsymbol{E}$ is the global template, $\boldsymbol{M}$ is the SPS kernel used to control the output size, $\boldsymbol{b}$ is bias.

As shown in Fig. 3, the shape of shared parameter matrix can thus be revised by simply adjusting the SPS kernel. For example, given the source matrix $\boldsymbol{E} \in \mathbb{R}^{r \times c \times p}$ and a target matrix $\boldsymbol{W} \in \mathbb{R}^{r_0 \times c_0 \times p_0}$, we simply need to set the shape of $\boldsymbol{M}$ with $[r - r_0 + 1, c - c_0 + 1, p_0, p]$.

With the SPS method equipped, the CNN modules in Eq. (5) can be replaced by Eq. (10), the linear modules in Eqs. (4) and (8) can be replaced by Eq. (11).

$$Conv1D(\boldsymbol{X}) = \boldsymbol{X} \otimes \boldsymbol{W}_c + \boldsymbol{b}_c, \boldsymbol{W}_c = SPS(\boldsymbol{M}_c). \tag{10}$$

$$Linear(\boldsymbol{X}) = \boldsymbol{X} \times \boldsymbol{W}_l + \boldsymbol{b}_l, \boldsymbol{W}_l = SPS(\boldsymbol{M}_l). \tag{11}$$

where $\boldsymbol{X}$ denotes the input data, $\boldsymbol{W}_c \in \mathbb{R}^{n \times n \times k}$ is the kernel of convolution module, k denotes the kernel size, $\boldsymbol{W}_l \in \mathbb{R}^{n \times n}$ is the parameter matrix of linear module. $\boldsymbol{b}_*$ are bias, $\otimes$ denotes convolution operation.

The SPS method only replace the trainable parameters of CNN and Linear modules with the global template, their structures and training processes are unchanged. Therefore, the SPS method can share knowledge among models while maintain the structural advantage of the original model.

Training process

In the training process, we set a specific training order to ensure the main task can take different aspects of the dataset knowledge into account. Specifically, we first train the auxiliary tasks to store the learned knowledge into the global template, and then we use this informative template as the initial parameter matrix of the ATR task to share knowledge. Figure 4 illustrates the training order of MODE.

At the very first, the template is initialized with random values. Then, the template will learn different aspects of knowledge by driving the model to train auxiliary tasks. Last, the ATR task can inherit the learned knowledge of auxiliary tasks by using the same template as the initial parameters.

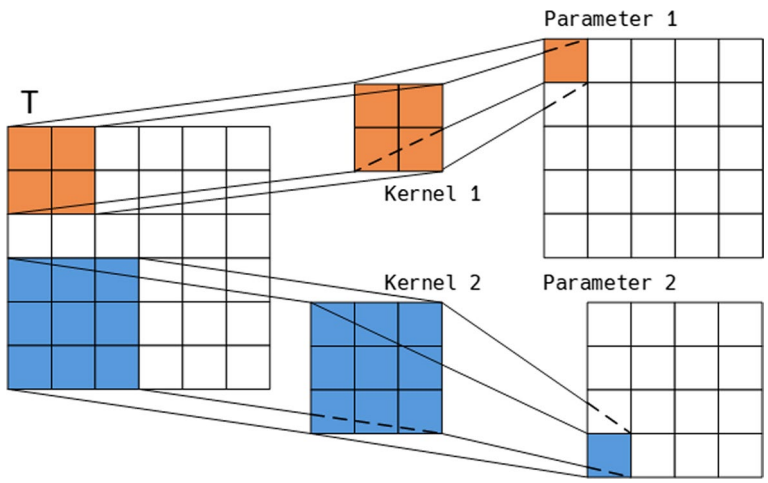

Fig. 3 The progress of soft parameter sharing

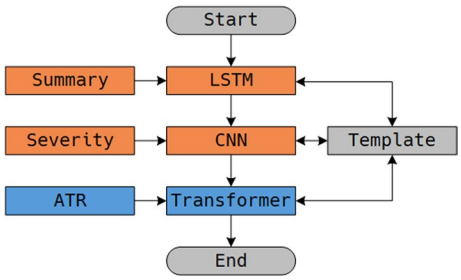

Fig. 4 The training process of MODE

For each task, we use the SPS model to compute the shared parameter matrices and to replace the original parameters of the model. Meanwhile, we use forward–backward steps to compute the gradients and update the template. With the Summary task as an example, Algorithm 1 illustrates the training process of a task.

Algorithm 1 The training process of a task

Input: Model: $model$, Template $\boldsymbol{E}$, SPS kernels $kernels$, Bias $bias$, Finding $\boldsymbol{X}$, Impression $\boldsymbol{Y}$, ϵ. // ϵ is a hyper-parameter used to judge the convergence.

Output: Updated Template $\hat{\boldsymbol{E}}$, Predicted Finding $\hat{\boldsymbol{Y}}$.

1: **for** parameter matrix $\boldsymbol{P}$, SPS kernel $\boldsymbol{M}$, bias $\boldsymbol{b}$) in ($model$, $kernels$, $bias$) **do**
2: Replace $\boldsymbol{P}$ with $\sigma(\boldsymbol{E} \otimes \boldsymbol{M} + \boldsymbol{b})$. // compute parameter matrix from the template.
3: **end for**
4: $\Delta \boldsymbol{E} := \inf$
5: **while** $\Delta \boldsymbol{E} > \epsilon$ **do**
6: $\hat{\boldsymbol{Y}} = model(\boldsymbol{X})$ // forward process.
7: $\boldsymbol{loss} = entropy(\hat{\boldsymbol{Y}}, \boldsymbol{Y})$ // compute loss.
8: $\hat{\boldsymbol{E}} := \boldsymbol{E} - gradient(\boldsymbol{loss})$ // backward process.
9: $\Delta \boldsymbol{E} := |\boldsymbol{E} - \hat{\boldsymbol{E}}|_2$
10: $\boldsymbol{E} := \hat{\boldsymbol{E}}$.
11: **end while**
12: **return** $\hat{\boldsymbol{E}}$, $\hat{\boldsymbol{Y}}$.

Experiment

In this section our goal is to showcase benefits of modeling multiple objectives in the same medical dataset, for which we define three tasks. We begin with two auxiliary tasks of Impression-generating and Severity-computing to learn different knowledge aspects from the dataset. Then, we define a main task of ATR. Finally, we train a specific model for each task and share the parameters of auxiliary tasks to the main task through SPS for which capability to share parameters would allow us to indirectly add more training samples to the limited dataset.

Dataset

We collect a group of 513 ultrasound reports from clinical patients, each report has two sequences of texts, Findings and Impressions. In addition, each report is labeled with a sequence of Treatment Recommendation and a severity score by a group of professional doctors, at least deputy chief physicians. We extract the sentences and corresponding labels from the unprocessed data. The only pre-processing operation of these sentences is to tokenize them into character level. The dataset is partitioned randomly into training set (70%), development set (10%) and test set (20%). The statistic of the dataset is listed in Table 1.

We define three tasks on the Ultrasound dataset. Specifically, Impression-generating is a summary task that generates Impression text for a given Finding. This task can alleviate the clinic doctors' workload during the ultrasound inspections. Severity-computing is a regression, or classification, task that computes the severity of the disease. To emphasis MODE is compatible with multiple loss functions, the Severity-computing is viewed as a regression task. This task can help the doctor to identify the patient's situation. ATR is a Question-Answering task that generates the Treatment Recommendation for an ultrasound report. This task can partly replace the doctor's work and make contributions to Auto Inspections of Artificial Intelligence Medical.

Competitor methods

Previous works have proposed various methods but not all of them can be applied to our tasks. We chose 4 most related sequence-to-sequence models for related tasks and implemented them as competitor methods.

- We used Medical LSTM [19] as the baseline of Internal LSTM. It is an alternative to the conventional sentiment analysis approaches in analysing large volumes of data in a potential flow.
- We used Memory-driven Transformer (MDT) [22] as the baseline of internal Transformer. It used a relational memory module and a memory-driven conditional layer normalization to record key information of the generation process of Transformer.
- We used Symptoms Frequency Position Attention (BiLSTM-SFPA) [20] as the baseline of external LSTM. It used adaptive weight assignment techniques and positional context to address APT task. Meanwhile, it used word2vec and the Chinese Ci-Lin as external knowledge.
- We used Bidirectional Transformer [26] as the baseline of external Transformer. It used Bidirectional-Transformer based architecture to generate encoded representations from external datasets firstly, and then use the learned knowledge to handle specific tasks.

Table 1 Statistics of the ultrasound dataset

Item	Type	Avg. len	Token	Label
Finding	Sequence	60	353	–
Impression	Sequence	31	263	–
Treatment	Sequence	53	277	–
Severity	Value	–	–	6

Hyper-parameters

We used random initialized vectors as word embeddings. All weights were randomly initialized by the Xavier Uniform Initializer. Dropout [40] rate, batch size and other hyper-parameters were set according to datasets and the memory capability. All texts in the same batch were padded to the same length. All models were optimized using the Adam optimizer [41]. Particularly, to select useful features, we used FP-net to extract features in an anti-gradient direction. Furthermore, Eq. (12) gives an epoch related learning rate updating policy with an initial learning rate of 0.01.

$$lr_{i+1} = 0.8 * lr_i * 0.01^{\frac{i+0.01}{epoch_{max}+0.01}}. \tag{12}$$

where lr denotes the learning rate, i is the current epoch index, $epoch_{max}$ is the max epoch, 40 in our experiments.

The experiments were conducted on a NVIDIA 3090 GPU with 24 GB memory and an Intel 10900x CPU with 64 GB memory.

Training settings

Table 2 shows the MLLs with various training settings on the development set, where **Time** refers to training time per epoch, **Param** represents the number of trainable parameters of the model. In the **Preference** column, Summary and Severity denote using corresponding auxiliary tasks. Independent and Collaborate denote whether the auxiliary tasks are trained separately or integrated in the main task. In the former method, each task is trained one-by-one and the training order is Summary-Severity-ATR. In the latter method, auxiliary tasks and the main task are trained simultaneously, the training order in each epoch is the same as Independent's and their losses are added up to perform backward steps. Flags denote adding start and end flags, $< start >$ and $< end >$, on a sequence. Since there are multiple training objectives, we use Minus Log Loss (MLL), $MLL(loss) = -log(loss)$, to represent the performance. A better model will produce larger MLL score.

As shown in the table, the MLL of ATR task drops to 1.49 without the help of Summary task or the Severity task, demonstrating the necessity of auxiliary knowledge. With the help of two auxiliary tasks, both independent and collaborative training policy can improve the performance of the ATR task. Although the performances of two methods

Table 2 Training settings of ATR task

Preference	Time (ms)	Dev. MLL	Param (M)
None	557	1.49	3.78
Summary	1380	1.51	3.97
Severity	602	1.54	3.93
Both	1390	1.57	4.12
Independent	557	1.56	3.78
Collaborate	1390	1.57	4.12
Non-flags	1390	1.57	4.12
Flags	1390	1.60	4.12

are similar, they have different advantages in applications. Specifically, the independent training policy utilizes auxiliary tasks as pure external knowledge. It is convenient to transfer this method to other tasks since adding models does not affect its structure. Collaborative training policy treats auxiliary tasks as not only external knowledge but additional training samples. Scaling up the dataset can fully train the model and decrease the risk of over-fitting. Consequently, the former method can be deployed in cross-domain areas while the latter one is more likely to perform well in disease-specific environments. Furthermore, without using start and end flags, the performance drops from 1.6 to 1.57, showing the effectiveness of having these additional nodes.

Ablation study

Influence of model size

Figure 5 illustrates the MLLs with different hidden sizes and auxiliary tasks on the test set. "None" denotes the ATR task was trained independently, "Impression" and "Severity" denote corresponding tasks were used to share their knowledge to the ATR task, "Both" denotes ATR task was trained with the help of both auxiliary models. When the hidden size increased from 128 to 1024, all methods reported increasing performance trends, which is consistent with the fact that a larger model usually has better representative ability. The performances generally increased before reaching a peak value, and then reported a decrease trend, although larger hidden sizes are adopted, which is consistent with the fact that too large models won't benefit the model. In comparison, few significant differences are observed over the peak performance with the help of auxiliary tasks. On the one hand, this shows reusing training samples can help the model to avoid over-fitting. On the other hand, this can be explained by the intuition that information exchange between auxiliary tasks and the main task can help the model to extract representative features and learn solid semantic distribution.

Influence of layers

Figure 6 illustrates the MLLs with different layers on the test set. The trend of results are similar with Fig. 5. When the number of layers increased from 1 to 10, all methods

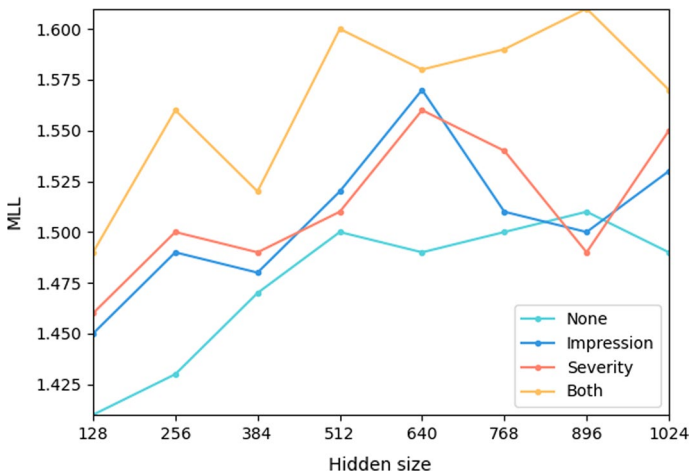

Fig. 5 Comparisons of different hidden size

reported increasing performance trends. This shows the model capability is proportional to its fitting ability. The performances of all models decreased after their peak value, which implies that stacking more layers does not extract more useful features, although the number of parameters and running time increase accordingly. Consequently, the model scale should match the dataset, using a large model to fit small datasets might not be a feasible solution. In addition, Multi-object MODE reported superior performances than its original versions. This shows that multi-objective structure can indirectly scale up the dataset and extract more useful features.

Influence of template size

Figure 7 illustrates the MLLs with different template settings on the test set. The template size is controlled by three hyper-parameters, r, c and k. The former two parameters affect the hidden size and the last parameter affects the template size. We fix r and c as 261 and adjust k from 1 to 10. In multi-objective training mode, the trend of MLL increases with k. This shows the template capability of MODE is coherent with NN models, and larger Template will produce higher performance. To validate whether multi-objective or the template contributes to the improvement, we trained the main task itself, without auxiliary tasks, with the Template. As expected, in single-task mode, MODE reports a relatively inferior performance than its multi-objective version. Consequently, multi-objective can indeed help the model to learn extensive knowledge and solid semantic distribution from auxiliary tasks, and improve the performance of the main task.

Case study

To further investigate the effectiveness of our method, we selected some cases and visualized their representation matrices. Figure 8 shows the data visualization results of an ultrasound report.

Figure 8a and b are the visualizations of Finding and Impression texts in the Severity task, Fig. 8c and d are the visualizations of predicted recommendations in the ATR

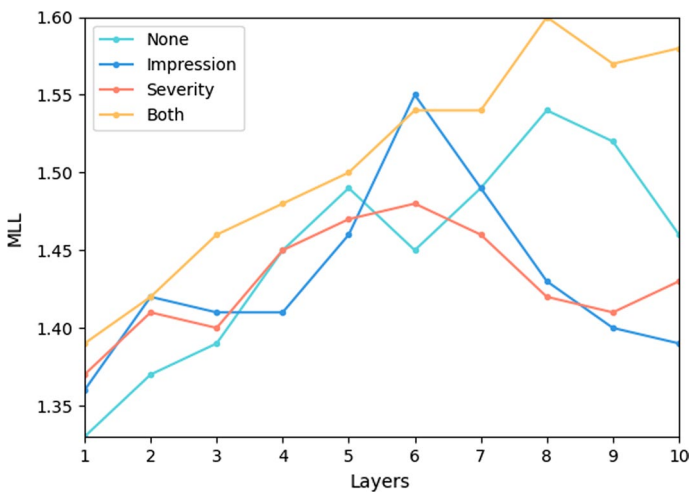

Fig. 6 Comparisons of different layers

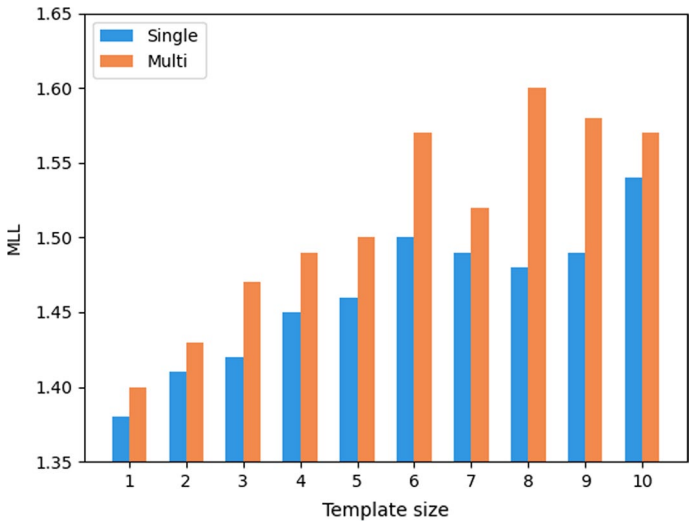

Fig. 7 Comparisons of different template settings

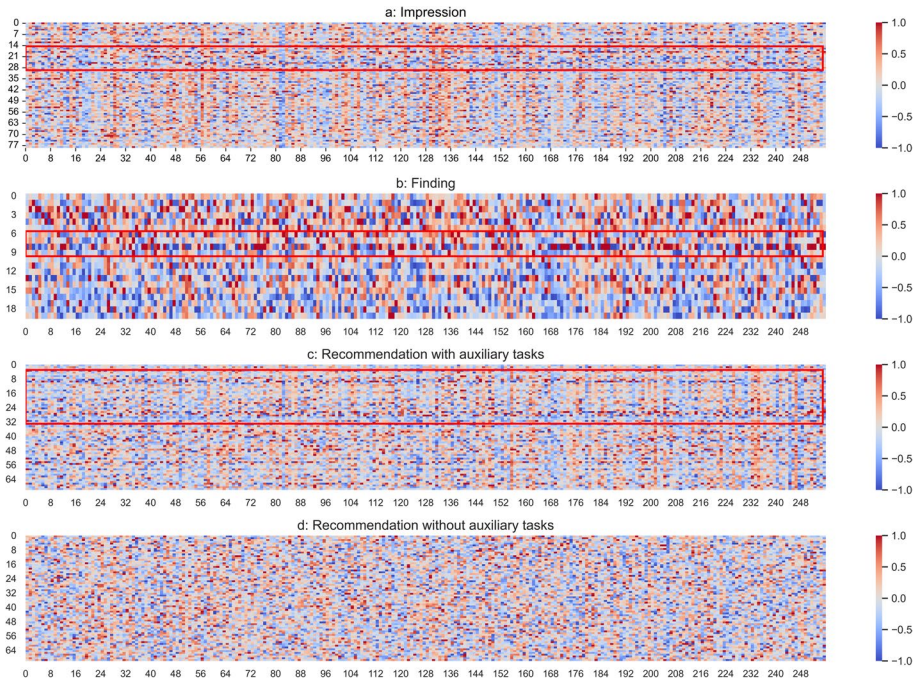

Fig. 8 The visualization of an ultrasound report

task, with and without the help of auxiliary tasks. In each figure, rows and columns represent words and semantic distribution respectively. It is observed in Fig. 8a and b that the representation vectors of some words are obviously different from the others, indicating that these words may have special relations and are critical to analyze the conditions of patients. As shown in Fig. 8c, the ATR task inherits these important words, which will benefit the generation of recommendation. In opposite, without the help of auxiliary tasks, shown in Fig. 8d, the visualization of the predicted

recommendation tends to be evenly distributed, indicating that ATR task cannot find informative words and lead to the insufficient performances. Consequently, the learned knowledge of auxiliary tasks can be used to increase the performance of the main task.

Space cost

It is intuitive that the space complexity of MODE is larger than existing models since MODE adopts multiple NN models. However, with the help of SPS, MODE has fewer parameters than a Multi-Task method should have. To illustrate this conclusion, we use the parameter size to show the impacts of multiple training objectives on space cost. Figure 9 illustrates the number of parameters of MODE, as well as its competitors, with different layers. In general, more layers will lead to high complexity, and the memory cost grows in proportion to the number of layers. Still, too large memory may introduce redundant and invalid information so as to negatively affect the generation process and lead to over-fitting. In comparison, the number of parameters of MODE grows slower than existing models, for most of the parameters are stored as the Template, and MODE only stores kernels and bias, which has much fewer parameters than the linear modules. It is worth noting that each model of LSTM, CNN and Transformer contains only one task, but MODE contains all of the three tasks. Although MODE has multiple tasks, its parameters are much fewer than existing models and have a relatively flatten increase trend. It is demonstrated that merely small amount of parameters are introduced when adding tasks and models in the memory. This observation suggests that the proposed MODE is effective and efficient in space cost. Similarly, Fig. 10 shows the number of parameters with different hidden sizes, which reaches the same outcome with the situation of different layers.

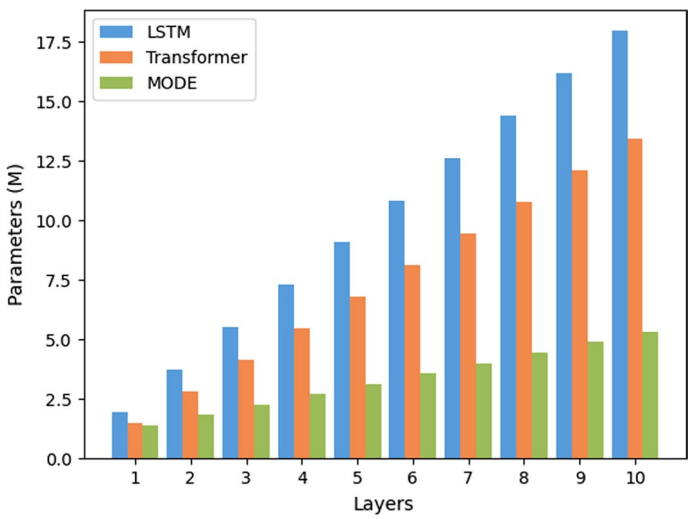

Fig. 9 The number of parameters with different number of layers

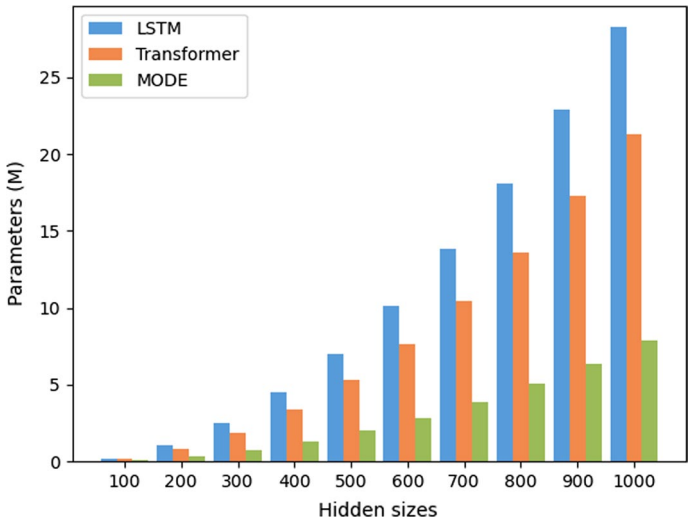

Fig. 10 The number of parameters with different hidden sizes

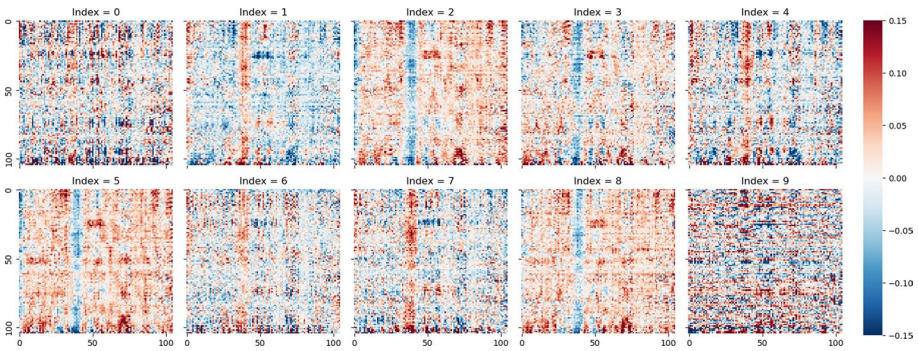

Fig. 11 Differences among template slices

Interpretability

We explore the parameter distribution of MODE to study its interpretability. To make clarity representations, we set template size with [10, 105, 105] and the slices of the Template are illustrated in Fig. 11. In the figure, we use heat-maps to visualize the parameters, the red dots represent positive values and blue dots represent negative values. At the first glance, there are three kinds of parameter distributions, in which the 0th and the 9th indices have little distinguishable information, the 1, 4, 6, 7th slices and the remains have similar structures but opposite values. The first and the last slices may store common knowledge or unimportant features, while other slices may store semantic features related to different training objectives. Each slice in the 2–8th slices has two notable vertical lines and a set of distributed clusters, and the second line has the same sign with the corresponding clusters. Obviously, red and blue lines, as well as clusters, indicate that the corresponding slices would like to pay more attention to these areas, which may be the key factors or important context words.

To further explain how MODE utilizes the Template, we present the distribution of SPS kernels. Figure 12 illustrates an example of a SPS kernel which is used to generate

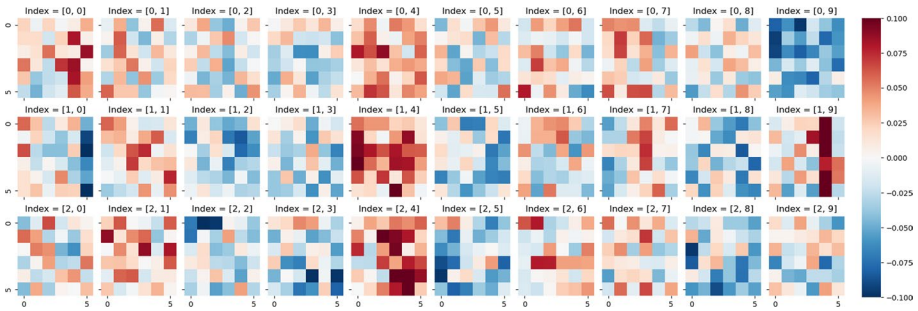

Fig. 12 Parameter distribution of SPS kernel

Table 3 Statistics of the final results

Preference	Time (ms)	tst. MLL	Param (M)
LSTM [19]	875	1.49	14.1
Transformer [22]	545	1.52	9.6
LSTM [20]	993	1.56	14.5
Transformer [26]	568	1.54	10.2
MODE	557	1.60	4.12

parameters for the last Attention module of the Transformer. Corresponding to the Fig. 11, we set the hidden size of Transformer with 100, and the kernel size should be set with [10, 1, 6, 6]. In the figure, the first term of indices $\{0, 1, 2\}$ denote the kernels of $\boldsymbol{W}_q$, $\boldsymbol{W}_k$ and $\boldsymbol{W}_v$ respectively, the second term of indices $\{0 \sim 9\}$ denote the weights of each slice in Fig. 12. At the first glance, we can see that the SPS kernels of $\boldsymbol{W}_q$ and $\boldsymbol{W}_k$ have relatively large values in the 4 and 9th slices and small values in others. $\boldsymbol{W}_v$ has a similar situation but the 4, 5, 8th slices are large (both positive and negative). It can be noting that, in Fig. 11, the vertical line of the 4th slice is positive, which has the same sign of the 4th kernel. This indicates that the 4th slice is strongly enhanced by the kernel. In opposite, although the parameters of the 5th kernel of $\boldsymbol{W}_v$ is large, the corresponding kernel of $\boldsymbol{W}_q$ has small values, near to zero, indicating that the 5th slice is suppressed by the kernel. The similar situation also happens in other kernels and slices. Consequently, each module will weight specific areas of the Template, and activating different areas of the Template will produce various semantic features.

Final results

The final results on the test set are shown in Table 3. In addition to training time per epoch, test times are additionally reported. We use the best settings on the development dataset for all models.

As shown in Table 3, the final results on the Ultrasound dataset are consistent with the development results. Internal and external methods have their own merits in handling the ATR task. Specifically, internal methods [19, 22] have fewer parameters and test time while external methods [20, 26] have better performance. The reason is external methods have to learn a large scale of dataset in their training stages, and more complex structures are needed. In comparison, internal methods have relatively small structures since too large models will lead to over-fitting in limited datasets.

These results illustrate that the scale of model structure should match the dataset scale, a larger model can extract abundant and highly qualified features but need more training samples. From this viewpoint, many existing methods have tried trade-off, not essential solutions, to solve small-scale and professional datasets.

MODE gives highly competitive results when compared with existing methods in the literature and reports relatively less parameters and test times. To improve the performance, MODE adopted a larger and more complex structure. To fully train MODE, we indirectly scale up the dataset by defining multiple training objectives to learn different aspects of knowledge. Meanwhile, we use SPS to take advantage of learned knowledge and decrease the space cost. Consequently, MODE has the merits of both internal and external methods, and achieved superior performance.

Conclusion

This paper tackles the contraction between insufficient training samples and professional knowledge in medical datasets. We propose the Multi-Objective Data Enhancement framework for learning and sharing various aspects of knowledge in the limited dataset. Compared with existing methods, MODE has two merits, (1) having the ability to scale up the dataset without external knowledge and (2) concentrating its parameters into a global parameter matrix to decrease the space cost.

The limitation of this paper would be that we only used text data in the experiments. Intuitively, using multi-modal information, such as signal and image data, to train the MODE will achieve better performance. But in applications, there is a quality problem in different data sources. An ultrasound report is a detailed description of a patient's condition, which is a formal document in clinical diagnosis and has a set of specific writing specifications. In opposite, ultrasound signal or video data are the records of clinical diagnosis, which contains much irrelative information to the report. It is difficult for an NN model to handle such irregular, or even vague data. Considering these two situations, we decided to use text data in the experiments. In addition, the MODE is theoretically a language-insensitive method, and it will function normally in different languages. But we only conducted the experiments in a single-language dataset. The reason is the ultrasound reports were collected and labeled by the cooperative clinicians, which are not familiar with English terms.

In the future work, we will try to utilize multi-modal information and seek for more multilingual datasets.

Acknowledgements
We appreciated the clinicians of Cangzhou Municipal Haixing Hospital for collecting the ultrasound reports.

Author contributions
CP wrote the main manuscript text. ML, SW and RZ collected and labeled the dataset. YW and JW prepared all figures. All authors reviewed the manuscript. All authors read and approved the final manuscript.

Funding
This work was partially supported by the National Key R &D Programs of China (2018YFC1603800, 2018YFC1603802, 2020YFA0908700, 2020YFA0908702), the National Natural Science Foundation of China (61772288, 61872115) and the Natural Science Foundation of Tianjin City (18JCZDJC30900).

Availability of data and materials
The experimental data is available at https://github.com/andrewpark3474/ultrasound-reports.

Declarations

Ethics approval and consent to participate
All experimental protocols were approved by the Ethics committee of Cangzhou Municipal Haixing Hospital. All methods were carried out in accordance with relevant guidelines and regulations. The informed consent was obtained from all subjects and/or their legal guardian(s).

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 31 May 2022 Accepted: 10 October 2022
Published online: 20 October 2022

References

1. Chen MC, Ball RL, Yang L, Moradzadeh N, Chapman BE, Larson DB, Langlotz CP, Amrhein TJ, Lungren MP. Deep learning to classify radiology free-text reports. Radiology. 2018;286(3):845–52. https://doi.org/10.1148/radiol.2017171115.
2. Noriega-Atala E, Hein PD, Thumsi SS, Wong Z, Wang X, Hendryx SM, Morrison CT. Extracting inter-sentence relations for associating biological context with events in biomedical texts. IEEE/ACM Trans Comput Biol Bioinf. 2020;17(6):1895–906. https://doi.org/10.1109/TCBB.2019.2904231.
3. Alafif T, Tehame AM, Bajaba S, Barnawi A, Zia S. Machine and deep learning towards Covid-19 diagnosis and treatment: survey, challenges, and future directions. Int J Environ Res Public Health. 2021;18(3):1117–40. https://doi.org/10.3390/ijerph18031117.
4. Kumar M, Gupta V. Benefits of using particle swarm optimization and Voronoi diagram for coverage in wireless sensor networks. In: 2017 international conference on emerging trends in computing and communication technologies (ICETCCT). Dehradun: IEEE; (2017). p. 1–7 https://doi.org/10.1109/ICETCCT.2017.8280300
5. Lotter W, Sorensen G, Cox D. A multi-scale CNN and curriculum learning strategy for mammogram classification. In: Cardoso MJ, Arbel T, Carneiro G, Syeda-Mahmood T, Tavares JMRS, Moradi M, Bradley A, Greenspan H, Papa JP, Madabhushi A, Nascimento JC, Cardoso JS, Belagiannis V, Lu Z, editors. Deep learning in medical image analysis and multimodal learning for clinical decision support. Cham: Springer; 2017. p. 169–77.
6. Lin S, Li Z, Fu B, Chen S, Li X, Wang Y, Wang X, Lv B, Xu B, Song X, Zhang Y-J, Cheng X, Huang W, Pu J, Zhang Q, Xia Y, Du B, Ji X, Zheng Z. Feasibility of using deep learning to detect coronary artery disease based on facial photo. Eur Heart J. 2020;41(46):4400–11. https://doi.org/10.1093/eurheartj/ehaa640.
7. Bria A, Marrocco C, Tortorella F. Addressing class imbalance in deep learning for small lesion detection on medical images. Comput Biol Med. 2020;120: 103735. https://doi.org/10.1016/j.compbiomed.2020.103735.
8. Zheng Z, Yan H, Setzer FC, Shi KJ, Mupparapu M, Li J. Anatomically constrained deep learning for automating dental CBCT segmentation and lesion detection. IEEE Trans Autom Sci Eng. 2021;18(2):603–14. https://doi.org/10.1109/TASE.2020.3025871.
9. Yap MH, Goyal M, Osman F, Martí R, Denton E, Juette A, Zwiggelaar R. Breast ultrasound region of interest detection and lesion localisation. Artif Intell Med. 2020;107: 101880. https://doi.org/10.1016/j.artmed.2020.101880.
10. Goel T, Murugan R, Mirjalili S, Chakrabartty DK. OptCoNet: an optimized convolutional neural network for an automatic diagnosis of COVID-19. Appl Intell. 2021;51(3):1351–66. https://doi.org/10.1007/s10489-020-01904-z.
11. Wang L, Zhang L, Zhu M, Qi X, Yi Z. Automatic diagnosis for thyroid nodules in ultrasound images by deep neural networks. Med Image Anal. 2020;61: 101665. https://doi.org/10.1016/j.media.2020.101665.
12. Benhammou Y, Achchab B, Herrera F, Tabik S. Breakhis based breast cancer automatic diagnosis using deep learning: taxonomy, survey and insights. Neurocomputing. 2020;375:9–24. https://doi.org/10.1016/j.neucom.2019.09.044.
13. Wang S, Zha Y, Li W, Wu Q, Li X, Niu M, Wang M, Qiu X, Li H, Yu H, Gong W, Bai Y, Li L, Zhu Y, Wang L, Tian J. A fully automatic deep learning system for COVID-19 diagnostic and prognostic analysis. Eur Respir J. 2020;56(2):2000775. https://doi.org/10.1183/13993003.00775-2020.
14. Zhou H, Yang Y, Ning S, Liu Z, Lang C, Lin Y, Huang D. Combining context and knowledge representations for chemical-disease relation extraction. IEEE/ACM Trans Comput Biol Bioinf. 2019;16(6):1879–89. https://doi.org/10.1109/TCBB.2018.2838661.
15. Enarvi S, Amoia M, Del-Agua Teba M, Delaney B, Diehl F, Hahn S, Harris K, McGrath L, Pan Y, Pinto J, Rubini L, Ruiz M, Singh G, Stemmer F, Sun W, Vozila P, Lin T, Ramamurthy R. Generating medical reports from patient-doctor conversations using sequence-to-sequence models. In: Proceedings of the first workshop on natural language processing for medical conversations. Association for Computational Linguistics; 2020. p. 22–30. https://doi.org/10.18653/v1/2020.nlpmc-1.4
16. Agrawal S, Jain SK. In: Jain V, Chatterjee JM, editors. Medical text and image processing: applications, issues and challenges. Cham: Springer; 2020. p. 237–262. https://doi.org/10.1007/978-3-030-40850-3-11
17. Shen Z, Zhang S. A novel deep-learning-based model for medical text classification. In: Proceedings of the 2020 9th international conference on computing and pattern recognition ICCPR 2020. New York, NY, USA: Association for Computing Machinery; 2020. p. 267–273. https://doi.org/10.1145/3436369.3436469
18. Ramesh N, Devi GL, Rao KS. A frame work for classification of multi class medical data based on deep learning and Naive Bayes classification model. Int J Inf Eng Electron Bus. 2020;10(1):37. https://doi.org/10.5815/ijieeb.2020.01.05.
19. Grissette H, Nfaoui EH. Adversarial LSTM-based sequence-to-sequence model for drug-related reactions understanding. In: Yang X-S, Sherratt S, Dey N, Joshi A, editors. Proceedings of sixth international congress on information and communication technology. Singapore: Springer; 2022. p. 49–59

20. Edara DC, Vanukuri LP, Sistla V, Kolli VKK. Sentiment analysis and text categorization of cancer medical records with LSTM. J Ambient Intell Humaniz Comput. 2019. https://doi.org/10.1007/s12652-019-01399-8.
21. He B, Guan Y, Dai R. Classifying medical relations in clinical text via convolutional neural networks. Artif Intell Med. 2019;93:43–9. https://doi.org/10.1016/j.artmed.2018.05.
22. Chen Z, Song Y, Chang TH, Wan X. Generating radiology reports via memory-driven transformer. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics; 2020. p. 1439–1449. https://doi.org/10.18653/v1/2020.emnlp-main.112
23. Li K, Chen C, Quan X, Ling Q, Song Y. Conditional augmentation for aspect term extraction via masked sequence-to-sequence generation. In: Proceedings of the 58th annual meeting of the association for computational linguistics. Association for Computational Linguistics; 2020. p. 7056–7066. https://doi.org/10.18653/v1/2020.acl-main.631
24. Cheng M, Yi J, Chen PY, Zhang H, Hsieh CJ. Seq2sick: Evaluating the robustness of sequence-to-sequence models with adversarial examples. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34; 2020. p. 3601–3608. https://doi.org/10.1609/aaai.v34i04.5767
25. Bressem KK, Adams LC, Gaudin RA, Tröltzsch D, Hamm B, Makowski MR, Schüle C-Y, Vahldiek JL, Niehues SM. Highly accurate classification of chest radiographic reports using a deep learning natural language model pre-trained on 3.8 million text reports. Bioinformatics. 2020;36(21):5255–61. https://doi.org/10.1093/bioinformatics/btaa668.
26. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, Cistac P, Rault T, Louf R, Funtowicz M, Davison J, Shleifer S, von Platen P, Ma C, Jernite Y, Plu J, Xu C, Le Scao T, Gugger S, Drame M, Lhoest Q, Rush A. Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations. Association for Computational Linguistics; 2020. p. 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6
27. Wang J, Zhang G, Wang W, Zhang K, Sheng Y. Cloud-based intelligent self-diagnosis and department recommendation service using Chinese medical Bert. J Cloud Comput. 2021;10(1):1–12. https://doi.org/10.1186/s13677-020-00218-2.
28. Guan J, Li R, Yu S, Zhang X. A method for generating synthetic electronic medical record text. IEEE/ACM Trans Comput Biol Bioinf. 2021;18(1):173–82. https://doi.org/10.1109/TCBB.2019.2948985.
29. Hahn U, Oleynik M. Medical information extraction in the age of deep learning. Yearb Med Inform. 2020;29(01):208–20. https://doi.org/10.1055/s-0040-1702001.
30. Yazdani A, Ghazisaeedi M, Ahmadinejad N, Giti M, Amjadi H, Nahvijou A. Automated misspelling detection and correction in Persian clinical text. J Digit Imaging. 2020;33(3):555–62. https://doi.org/10.1007/s10278-019-00296-y.
31. Yue L, Tian D, Chen W, Han X, Yin M. Deep learning for heterogeneous medical data analysis. World Wide Web. 2020;23(5):2715–37. https://doi.org/10.1007/s11280-019-00764-z.
32. Li LJ, Niu CQ, Pu DX, Jin XY. Electronic medical data analysis based on word vector and deep learning model. In: 2018 9th international conference on information technology in medicine and education (ITME); 2018. p. 484–487 . https://doi.org/10.1109/ITME.2018.00114
33. Borjali A, Magnéli M, Shin D, Malchau H, Muratoglu OK, Varadarajan KM. Natural language processing with deep learning for medical adverse event detection from free-text medical narratives: a case study of detecting total hip replacement dislocation. Comput Biol Med. 2021;129(3): 104140. https://doi.org/10.1016/j.compbiomed.2020.104140.
34. Liu J, Zhang Z, Razavian N. Deep ehr: Chronic disease prediction using medical notes. In: Doshi-Velez F, Fackler J, Jung K, Kale D, Ranganath R, Wallace B, Wiens J, editors. Proceedings of the 3rd machine learning for healthcare conference. Proceedings of Machine Learning Research, vol 85. PMLR; 2018.p. 440–464
35. Prabhakar SK, Won DO, Maleh Y. Medical text classification using hybrid deep learning models with multihead attention. Intell Neurosci. 2021;2021:9425655. https://doi.org/10.1155/2021/9425655.
36. Rebane J, Samsten I, Papapetrou P. Exploiting complex medical data with interpretable deep learning for adverse drug event prediction. Artif Intell Med. 2020;109: 101942. https://doi.org/10.1016/j.artmed.2020.101942.
37. Qin Q, Hu W, Liu B. Feature projection for improved text classification. In: Proceedings of the 58th annual meeting of the association for computational linguistics. Association for Computational Linguistics; 2020. p. 8161–8171. https://doi.org/10.18653/v1/2020.acl-main.726
38. Ji Y, Zhou Z, Liu H, Davuluri RV. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. Bioinformatics. 2021;37(15):2112–20. https://doi.org/10.1093/bioinformatics/btab083.
39. Li F, Jin Y, Liu W, Rawat BPS, Cai P, Yu H. Fine-tuning bidirectional encoder representations from transformers (BERT)-based models on large-scale electronic health record notes: an empirical study. JMIR Med Inform. 2019;7(3):14830. https://doi.org/10.2196/14830.
40. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(56):1929–58.
41. Kingma DP, Ba J. Adam: a method for stochastic optimization. 2014. arXiv:1412.6980

Publisher's Note