Question Answering based Clinical Text Structuring Using Pre-trained Language Model

Jiahui Qiu[1], Yangming Zhou[1,*], Zhiyuan Ma[1], Tong Ruan[1,*], Jinlin Liu[1], and Jing Sun[2]
[1]School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China
[2]Ruijin Hospital, School of Medicine, Shanghai Jiao Tong University, Shanghai 200025, China
*Corresponding authors
Emails: {ymzhou,ruantong}@ecust.edu.cn

***Abstract*—Clinical text structuring is a critical and fundamental task for clinical research. Traditional methods such as task-specific end-to-end models and pipeline models usually suffer from the lack of dataset and error propagation. In this paper, we present a question answering based clinical text structuring (QA-CTS) task to unify different specific tasks and make dataset shareable. A novel model that aims to introduce domain-specific features (e.g., clinical named entity information) into pre-trained language model is also proposed for QA-CTS task. Experimental results on Chinese pathology reports collected from Ruijing Hospital demonstrate our presented QA-CTS task is very effective to improve the performance on specific tasks. Our proposed model also competes favorably with strong baseline models in specific tasks.**

***Index Terms*—Question answering, Clinical text structuring, Pre-trained language model, Electronic health records.**

I. Introduction

Clinical text structuring (CTS) is a critical task for fetching medical research data from electronic health records (EHRs), where structural patient medical data, such as whether the patient has specific symptoms, diseases, or what the tumor size is, how far from the tumor is cut at during the surgery, or what the specific laboratory test result is, are obtained. It is important to extract structured data from clinical text because bio-medical systems or bio-medical researches greatly rely on structured data but they cannot obtain them directly. In addition, clinical text often contains abundant healthcare information. CTS is able to provide large-scale extracted structured data for enormous down-stream clinical researches.

However, end-to-end CTS is a very challenging task. Different CTS tasks often have non-uniform output formats, such as specific-class classifications (e.g. tumor stage), strings in the original text (e.g. result for a laboratory test) and inferred values from part of the original text (e.g. calculated tumor size). Researchers have to construct different models for it, which is already costly, and hence it calls for a lot of labeled data for each model. Moreover, labeling necessary amount of data for training neural network requires expensive labor cost. To handle it, researchers turn to some rule-based structuring methods which often have lower labor cost.

Traditionally, CTS tasks can be addressed by rule and dictionary based methods [1]–[3], task-specific end-to-end methods [4]–[7] and pipeline methods [8]–[10]. Rule and dictionary based methods suffer from costly human-designed extraction rules, while task-specific end-to-end methods have non-uniform output formats and require task-specific training dataset. Pipeline methods break down the entire process into several pieces which improves the performance and generality. However, when the pipeline depth grows, error propagation will have a greater impact on the performance.

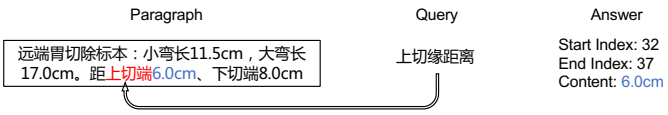

Fig. 1. An illustrative example of QA-CTS task.

To reduce the pipeline depth and break the barrier of non-uniform output formats, we present a question answering based clinical text structuring (QA-CTS) task (see Fig. 1). Unlike the traditional CTS task, our QA-CTS task aims to discover the most related text from original paragraph text. For some cases, it is already the final answer in deed (e.g., extracting sub-string). While for other cases, it needs several steps to obtain the final answer, such as entity names conversion and negative words recognition. Our presented QA-CTS task unifies the output format of the traditional CTS task and make the training data shareable, thus enriching the training data. The main contributions of this work can be summarized as follows.

- We first present a question answering based clinical text structuring (QA-CTS) task, which unifies different specific tasks and make dataset shareable. We also propose an effective model to integrate clinical named entity information into pre-trained language model.
- Experimental results show that QA-CTS task leads to significant improvement due to shared dataset. Our proposed model also achieves significantly better performance than the strong baseline methods. In addition, we also show that two-stage training mechanism has a great improvement on QA-CTS task.

The rest of the paper is organized as follows. We briefly review the related work on clinical text structuring in Section II. Then, we present question answer based clinical text

TABLE I
AN ILLUSTRATIVE EXAMPLE OF NAMED ENTITY FEATURE TAGS

Character Sequence	远	端	胃	切	除	标	本	:	小	弯	长	1	1	.	5	c	m
Tag Sequence	B-o	I-o	I-o	I-o	E-o	O	O	O	O	O	O	B-n	I-n	I-n	E-n	B-u	E-u
Entity Type	operation											number				unit word	

* The B-tag indicates the beginning of an entity. The I-tag indicates the inside of an entity. The E-tag indicates the end of an entity. The O-tag indicates the character is outside an entity. The S-tag indicates the character is merely a single-character entity. As for entity types, the o-tag indicates the entity is an operation, the n-tag indicates the entity is an number, and the u-tag indicates the entity is an unit word.

structuring task in Section III. In Section IV, we present an effective model for this task. Section V is devoted to computational studies and several investigations on the key issues of our proposed model. Finally, conclusions are given in Section VI.

II. RELATED WORK

A. Clinical Text Structuring

Clinical text structuring is a final problem which is highly related to practical applications. Most of existing studies are case-by-case. Few of them are developed for the general purpose structuring task. These studies can be roughly divided into three categories: rule and dictionary based methods, task-specific end-to-end methods and pipeline methods.

Rule and dictionary based methods [1]–[3] rely extremely on heuristics and handcrafted extraction rules which is more of an art than a science and incurring extensive trial-and-error experiments. Fukuda et al. [1] identified protein names from biological papers by dictionaries and several features of protein names. Wang et al. [2] developed some linguistic rules (i.e. normalised/expanded term matching and substring term matching) to map specific terminology to SNOMED CT. Song et al. [3] proposed a hybrid dictionary-based bio-entity extraction technique and expands the bio-entity dictionary by combining different data sources and improves the recall rate through the shortest path edit distance algorithm. This kind of approach features its interpretability and easy modifiability. However, with the increase of the rule amount, supplementing new rules to existing system will turn to be a rule disaster.

Task-specific end-to-end methods [4], [5] use large amount of data to automatically model the specific task. Topaz et al. [4] constructed an automated wound information identification model with five output. Tan et al. [5] identified patients undergoing radical cystectomy for bladder cancer. Although they achieved good performance, none of their models could be used to another task due to output format difference. This makes building a new model for a new task a costly job.

Pipeline methods [8]–[10] break down the entire task into several basic natural language processing tasks. Bill et al. [8] focused on attributes extraction which mainly relied on dependency parsing and named entity recognition [11]–[13]. Meanwhile, Fonferko et al. [10] used more components like noun phrase chunking [14]–[16], part-of-speech tagging [17]–[19], sentence splitter, named entity linking [20]–[22], relation extraction [23], [24]. This kind of method focus on language itself, so it can handle tasks more general. However, as the depth of pipeline grows, it is obvious that error propagation will be more and more serious. In contrary, using less components to decrease the pipeline depth will lead to a poor performance. So the upper limit of this method depends mainly on the worst component.

B. Pre-trained Language Model

Recently, some works focused on pre-trained language representation models to capture language information from text and then utilizing the information to improve the performance of specific natural language processing tasks [25]–[28] which makes language model a shared model to all natural language processing tasks. Radford et al. [25] proposed a framework for fine-tuning pre-trained language model. Peters et al. [26] proposed ELMo which concatenates forward and backward language models in a shallow manner. Devlin et al. [27] used bidirectional Transformers to model deep interactions between the two directions. Yang et al. [28] replaced the fixed forward or backward factorization order with all possible permutations of the factorization order and avoided using the [MASK] tag which causes pretrain-finetune discrepancy that BERT is subject to.

The main motivation of introducing pre-trained language model is to solve the shortage of labeled data and polysemy problem. Although polysemy problem is not a common phenomenon in biomedical domain, shortage of labeled data is always a non-trivial problem. Lee et al. [29] applied BERT on large-scale biomedical unannotated data and achieved improvement on biomedical named entity recognition, relation extraction and question answering. Kim et al. [30] adapted BioBERT into multi-type named entity recognition and discovered new entities. Both of them demonstrates the usefulness of introducing pre-trained language model into biomedical domain.

III. QUESTION ANSWERING BASED CLINICAL TEXT STRUCTURING

Given a sequence of paragraph text $X =< x_1, x_2, ..., x_n >$, clinical text structuring (CTS) can be regarded to extract or generate a key-value pair where key Q is typically a query term such as proximal resection margin and value V is a result of query term Q according to the paragraph text X.

Generally, researchers solve CTS problem in two steps. Firstly, the answer-related text is pick out. And then several steps such as entity names conversion and negative words recognition are deployed to generate the final answer. While final answer varies from task to task, which truly causes non-uniform output formats, finding the answer-related text is a

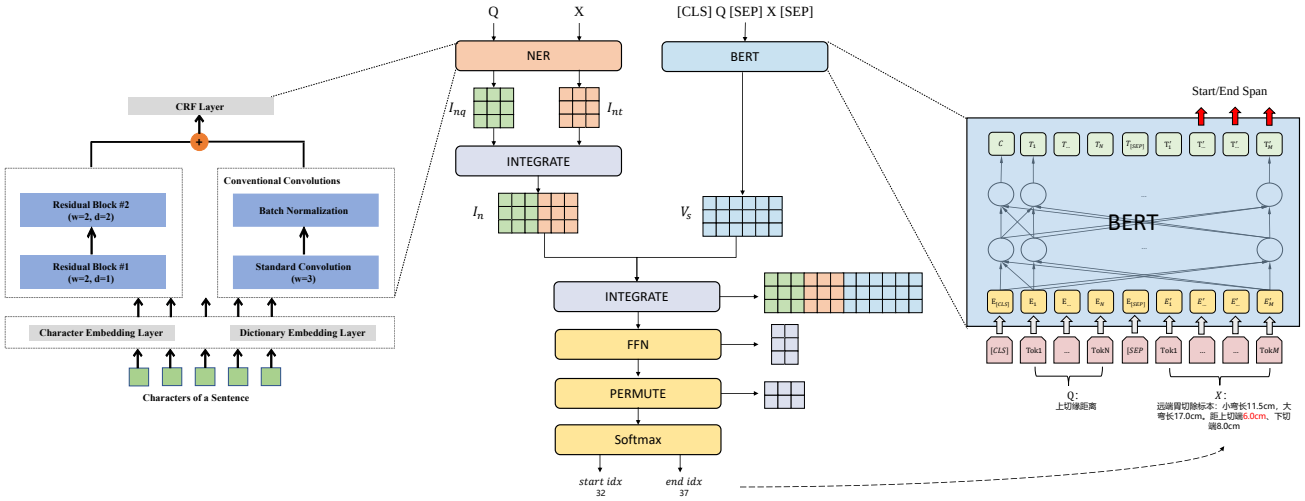

Fig. 2. The architecture of our proposed model for QA-CTS task

common action among all tasks. Traditional methods regard both the steps as a whole. In this paper, we focus on finding the answer-related substring $Xs =< X_i, X_i + 1, X_i + 2, ...X_j > (1 <= i < j <= n)$ from paragraph text X. For example, given sentence "远端胃切除标本：小弯长11.5cm，大弯长17.0cm。距上切端6.0cm、下切端8.0cm" (Distal gastrectomy specimen: measuring 11.5cm in length along the lesser curvature, 17.0cm in length along the greater curvature; 6.0cm from the proximal resection margin, and 8.0cm from the distal resection margin) and query "上切缘距离"(proximal resection margin), the answer should be 6.0cm which is located in original text from index 32 to 37. With such definition, it unifies the output format of CTS tasks and therefore make the training data shareable, in order to reduce the training data quantity requirement.

Since BERT [27] has already demonstrated the usefulness of shared model, we suppose extracting commonality of this problem and unifying the output format will make the model more powerful than dedicated model and meanwhile, for a specific clinical task, use the data for other tasks to supplement the training data.

IV. The Proposed Model for QA-CTS Task

In this section, we present an effective model for the question answering based clinical text structuring (QA-CTS). As shown in Fig. 2, paragraph text X is first passed to a clinical named entity recognition (CNER) model [13] to capture named entity information and obtain one-hot CNER output tagging sequence for query text I_{nq} and paragraph text I_{nt} with BIEOS (Begin, Inside, End, Outside, Single) tag scheme. I_{nq} and I_{nt} are then integrated together into I_n. Meanwhile, the paragraph text X and query text Q are organized and passed to contextualized representation model which is pre-trained language model BERT [27] here to obtain the contextualized representation vector V_s of both text and query. Afterwards, V_s and I_n are integrated together and fed into a feed forward network to calculate the start and end index of answer-related text. Here we define this calculation problem as a classification for each word to be the start or end word.

A. Contextualized Representation of Sentence Text and Query Text

For any clinical free-text paragraph X and query Q, contextualized representation is to generate the encoded vector of both of them. Here we use pre-trained language model BERT-base [27] model to capture contextual information.

The text input is constructed as '[CLS] Q [SEP] X [SEP]'. For Chinese sentence, each word in this input will be mapped to a pre-trained embedding e_i. To tell the model Q and X is two different sentence, a sentence type input is generated which is a binary label sequence to denote what sentence each character in the input belongs to. Positional encoding and mask matrix is also constructed automatically to bring in absolute position information and eliminate the impact of zero padding respectively. Then a hidden vector V_s which contains both query and text information is generated through BERT-base model.

B. Clinical Named Entity Information

Since BERT is trained on general corpus, its performance on biomedical domain can be improved by introducing biomedical domain-specific features. In this paper, we introduce clinical named entity information into the model.

The CNER task aims to identify and classify important clinical terms such as diseases, symptoms, treatments, exams, and body parts from Chinese EHRs. It can be regarded as a sequence labeling task. A CNER model typically outputs a sequence of tags. Each character of the original sentence will be tagged a label following a tag scheme. In this paper we

recognize the entities by the model of our previous work [13] but trained on another corpus which has 44 entity types including operations, numbers, unit words, examinations, symptoms, negative words, etc. An illustrative example of named entity information sequence is demonstrated in Table I. In Table I, "远端胃切除" is tagged as an operation, '11.5' is a number word and 'cm' is an unit word. The named entity tag sequence is organized in one-hot type. We denote the sequence for clinical sentence and query term as I_{nt} and I_{nq}, respectively.

C. Integration Method

There are two ways to integrate two named entity information vectors I_{nt} and I_{nq} or hidden contextualized representation V_s and named entity information I_n, where $I_n = [I_{nt}; I_{nq}]$. The first one is to concatenate them together because they have sequence output with a common dimension. The second one is to transform them into a new hidden representation. For the concatenation method, the integrated representation is described as follows.

$$H_i = [I_n; V_s] \tag{1}$$

While for the transformation method, we use multi-head attention [31] to encode the two vectors. It can be defined as follows where h is the number of heads and W_o is used to projects back the dimension of concatenated matrix.

$$\begin{aligned} H_i = [&Attention_1(W_{q1}Q', W_{k1}K, W_{v1}V); \\ &...; Attention_h(Q', K, V)]W_o \end{aligned} \tag{2}$$

$Attention$ denotes the traditional attention and it can be defined as follows.

$$Attention(Q', K, V) = softmax(\frac{Q'K^T}{\sqrt{d_k}}) \tag{3}$$

where d_k is the length of hidden vector.

D. Final Prediction

The final step is to use integrated representation H_i to predict the start and end index of answer-related text. Here we define this calculation problem as a classification for each word to be the start or end word. We use a feed forward network (FFN) to compress and calculate the score of each word H_f which makes the dimension to $\langle l_s, 2\rangle$ where l_s denotes the length of sequence.

$$H_f = FFN(H_i) \tag{4}$$

Then we permute the two dimensions for softmax calculation. The calculation process of loss function can be defined as followed.

$$L = -\sum_{i=1}^{l_s} y_{s_i} log(O_{s_i}) - \sum_{i=1}^{l_s} y_{e_i} log(O_{e_i}) \tag{5}$$

where $O_s = softmax(permute(H_f)_0)$ denotes the probability score of each word to be the start word and similarly $O_e = softmax(permute(H_f)_1)$ denotes the end. y_s and y_e denotes the true answer of the output for start word and end word respectively.

E. Two-Stage Training Mechanism

Two-stage training mechanism is previously applied on bilinear model in fine-grained visual recognition [32]–[34]. Two CNNs are deployed in the model. One is trained at first for coarse-graind features while freezing the parameter of the other. Then unfreeze the other one and train the entire model in a low learning rate for fetching fine-grained features.

Inspired by this and due to the large amount of parameters in BERT model, to speed up the training process, we fine tune the BERT model with new prediction layer first to achieve a better contextualized representation performance. Then we deploy the proposed model and load the fine tuned BERT weights, attach named entity information layers and retrain the model.

V. Experimental Studies

In this section, we devote to experimentally evaluating our proposed task and approach. The best results in tables are in bold.

A. Dataset and Evaluation Metrics

Our dataset is annotated based on Chinese pathology reports provided by the Department of Gastrointestinal Surgery, Ruijin Hospital. It contains 17,833 sentences, 826,987 characters and 2,714 question-answer pairs. All question-answer pairs are annotated and reviewed by four clinicians with three types of questions, namely tumor size, proximal resection margin and distal resection margin. These annotated instances have been partitioned into 1,899 training instances (12,412 sentences) and 815 test instances (5,421 sentences). Each instance has one or several sentences. Detailed statistics of different types of entities are listed in Table II.

TABLE II
Statistics of different types of question answer instances

Type	Training Set	Test Set
Proximal Resection Margin	643	290
Distal Resection Margin	681	270
Tumor Size	575	255
Total	1,899	815

In the following experiments, two widely-used performance measures (i.e., EM-score [35] and (macro-averaged) F_1-score [36]) are used to evaluate the methods. The Exact Match (EM-score) metric measures the percentage of predictions that match any one of the ground truth answers exactly. The F_1-score metric is a looser metric measures the average overlap between the prediction and ground truth answer.

B. Experimental Settings

To implement deep neural network models, we utilize the Keras library [37] with TensorFlow [38] backend. Each model is run on a single NVIDIA GeForce GTX 1080 Ti GPU. The models are trained by Adam optimization algorithm [39] whose parameters are the same as the default settings except for learning rate set to 5×10^{-5}. Batch size is set to 3 or 4

due to the lack of graphical memory. We select BERT-base as the pre-trained language model in this paper. Due to the high cost of pre-training BERT language model, we directly adopt parameters pre-trained by Google in Chinese general corpus. The named entity recognition is applied on both pathology report texts and query texts.

C. Comparison with State-of-the-art Methods

Since BERT has already achieved the state-of-the-art performance of question-answering, in this section we compare our proposed model with state-of-the-art question answering models (i.e. QANet [**?**]) and BERT-Base [27]. As BERT has two versions: BERT-Base and BERT-Large, due to the lack of computational resource, we can only compare with BERT-Base model instead of BERT-Large. Prediction layer is attached at the end of the original BERT-Base model and we fine tune it on our dataset. In this section, the named entity integration method is chosen to pure concatenation (Concatenate the named entity information on pathology report text and query text first and then concatenate contextualized representation and concatenated named entity information). Comparative results are summarized in Table III.

TABLE III
COMPARATIVE RESULTS BETWEEN BERT AND OUR PROPOSED MODEL

Methods	EM-score	F_1-score
QANet	85.45	93.62
BERT-Base	86.20	90.06
Our Proposed Model	**91.84**	**93.75**

Table III indicates that our proposed model achieved the best performance both in EM-score and F_1-score with EM-score of 91.84% and F_1-score of 93.75%. QANet outperformed BERT-Base with 3.56% score in F_1-score but underperformed it with 0.75% score in EM-score. Compared with BERT-Base, our model led to a 5.64% performance improvement in EM-score and 3.69% in F_1-score. Although our model didn't outperform much with QANet in F_1-score (only 0.13%), our model significantly outperformed it with 6.39% score in EM-score.

D. Ablation Analysis

To further investigate the effects of named entity information and two-stage training mechanism for our model, we apply ablation analysis to see the improvement brought by each of them, where $\times$ refers to removing that part from our model.

TABLE IV
COMPARATIVE RESULTS FOR DIFFERENT VARIANTS OF OUR PROPOSED MODEL

Named Entity Information	Two-Stage Training	EM-score	F_1-score
✓	✓	**91.84**	**93.74**
✓	×	87.48	89.94
×	×	86.20	90.06

As demonstrated in Table IV, with named entity information enabled, two-stage training mechanism improved the result by 4.36% in EM-score and 3.8% in F_1-score. Without two-stage training mechanism, named entity information led to an improvement by 1.28% in EM-score but it also led to a weak deterioration by 0.12% in F_1-score. With both of them enabled, our proposed model achieved a 5.64% score improvement in EM-score and a 3.69% score improvement in F_1-score. The experimental results show that both named entity information and two-stage training mechanism are helpful to our model.

E. Comparisons Between Two Integration Methods

There are two methods to integrate named entity information into existing model, we experimentally compare these two integration methods. As named entity recognition has been applied on both pathology report text and query text, there will be two integration here. One is for two named entity information and the other is for contextualized representation and integrated named entity information. For multi-head attention [31], we set heads number $h = 16$ with 256-dimension hidden vector size for each head.

TABLE V
COMPARATIVE RESULTS FOR DIFFERENT INTEGRATION METHOD OF OUR PROPOSED MODEL

Period One	Period Two	EM-score	F_1-score
Concatenation	Concatenation	**91.84**	**93.74**
Concatenation	Multi-head Attention	80.74	84.42
Multi-head Attention	Concatenation	89.08	92.88

* Period One is for integrating I_{nt} and I_{nq}. Period Two is for integrating I_n and V_s.
* Applying Multi-head Attention on both period one and period two can not reach convergence in our experiments.

From Table V, we can observe that applying concatenation on both periods achieved the best performance on both EM-score and F_1-score. Unfortunately, applying multi-head attention on both period one and period two can not reach convergence in our experiments. This probably because it makes the model too complex to train. The difference on other two methods are the order of concatenation and multi-head attention. Applying multi-head attention on two named entity information I_{nt} and I_{nq} first achieved a better performance with 89.87% in EM-score and 92.88% in F_1-score. Applying Concatenation first can only achieve 80.74% in EM-score and 84.42% in F_1-score. This is probably due to the processing depth of hidden vectors and dataset size. BERT's output has been modified after many layers but named entity information representation is very close to input. With big amount of parameters in multi-head attention, it requires massive training to find out the optimal parameters. However, our dataset is significantly smaller than what pre-trained BERT uses. This probably can also explain why applying multi-head attention method on both periods can not converge.

Although Table V shows the best integration method is concatenation, multi-head attention still has great potential. Due to the lack of computational resources, our experiment fixed the head number and hidden vector size. However, tuning these hyper parameters may have impact on the result. Tuning

TABLE VI
COMPARATIVE RESULTS FOR DATA INTEGRATION ANALYSIS (WITHOUT TWO-STAGE TRAINING AND NAMED ENTITY INFORMATION)

	Tumor Size		Proximal Resection Margin		Distal Resection Margin	
	EM-score	F_1-score	EM-score	F_1-score	EM-score	F_1-score
Pure Tumor Size	**96.47**	**96.47**	0.00	17.59	0.00	21.48
Pure Proximal Resection Margin	0.00	18.82	80.86	**83.45**	6.85	41.48
Pure Distal Resection Margin	0.00	21.18	5.00	45.17	86.48	**85.93**
Mixed Data	94.90	93.33	**81.55**	80.34	**86.85**	85.16

TABLE VII
COMPARATIVE RESULTS FOR DATA INTEGRATION ANALYSIS (WITH TWO-STAGE TRAINING AND NAMED ENTITY INFORMATION)

	Tumor Size		Proximal Resection Margin		Distal Resection Margin	
	EM-score	F_1-score	EM-score	F_1-score	EM-score	F_1-score
Pure Tumor Size	**96.27**	**96.08**	0.00	17.93	0.00	21.48
Pure Proximal Resection Margin	0.00	19.22	84.48	85.86	6.67	40.00
Pure Distal Resection Margin	0.00	21.18	4.65	44.83	88.33	87.41
Mixed Data	95.10	94.51	**88.45**	**88.28**	**92.41**	**91.48**

TABLE VIII
COMPARATIVE RESULTS FOR DATA INTEGRATION ANALYSIS (USING MIXED-DATA PRE-TRAINED PARAMETERS)

	Tumor Size		Proximal Resection Margin		Distal Resection Margin	
	EM-score	F_1-score	EM-score	F_1-score	EM-score	F_1-score
Pure Tumor Size	**96.27**	**96.08**	30.86	27.93	43.52	41.48
Pure Proximal Resection Margin	71.18	61.96	85.00	87.25	69.26	70.74
Pure Distal Resection Margin	64.31	55.69	73.62	78.97	90.93	90.37
Mixed Data	95.10	94.51	**88.45**	**88.28**	**92.41**	**91.48**

integration method and try to utilize larger datasets may give help to improving the performance.

F. Data Integration Analysis

To investigate how shared task and shared model can benefit, we split our dataset by query types, train our proposed model with different datasets and demonstrate their performance on different datasets. Firstly, we investigate the performance on model without two-stage training and named entity information.

As indicated in Table VI, The model trained by mixed data outperforms 2 of the 3 original tasks in EM-score with 81.55% for proximal resection margin and 86.85% for distal resection margin. The performance on tumor size declined by 1.57% score in EM-score and 3.14% score in F_1-score but they were still above 90%. 0.69% and 0.37% score improvement in EM-score was brought by shared model for proximal and distal resection margin prediction. Meanwhile F_1-score for those two tasks declined 3.11% and 0.77% score.

Then we investigate the performance on model with two-stage training and named entity information. In this experiment, pre-training process only use the specific dataset not the mixed data. From Table VII we can observe that the performance on proximal and distal resection margin achieved the best performance on both EM-score and F_1-score. Compared with Table VI, the best performance on proximal resection margin improved by 6.9% in EM-score and 7.94% in F_1-score. Meanwhile, the best performance on distal resection margin improved by 5.56% in EM-score and 6.32% in F_1-score. Other performances also usually improved a lot. This proves the usefulness of two-stage training and named entity information as well.

Lastly, we fine tune the model for each task with a pre-trained parameter. Table VIII summarizes the result. (Add some explanations for the Table VIII). Comparing Table VIII with Table VII, using mixed-data pre-trained parameters can significantly improve the model performance than task-specific data trained model. Except tumor size, the result was improved by 0.52% score in EM-score, 1.39% score in F_1-score for proximal resection margin and 2.6% score in EM-score, 2.96% score in F_1-score for distal resection margin. This proves mixed-data pre-trained parameters can lead to a great benefit for specific task. Meanwhile, the model performance on other tasks which are not trained in the final stage was also improved from around 0 to 60 or 70 percent. This proves that there is commonality between different tasks and our proposed QA-CTS task make this learnable. In conclusion, to achieve the best performance for a specific dataset, pre-training the model in multiple datasets and then fine tuning the model on the specific dataset is the best way.

VI. CONCLUSION

In this paper, we present a question answering based clinical text structuring (QA-CTS) task, which unifies different clinical text structuring tasks and utilize different datasets. A novel model is also proposed to integrate named entity information into a pre-trained language model and adapt it to QA-CTS task. Initially, sequential results of named entity recognition

on both paragraph and query texts are integrated together. Contextualized representation on both paragraph and query texts are transformed by a pre-trained language model. Then, the integrated named entity information and contextualized representation are integrated together and fed into a feed forward network for final prediction. Experimental results on real-world dataset demonstrate that our proposed model competes favorably with strong baseline models in all three specific tasks. The shared task and shared model introduced by QA-CTS task has also been proved to be useful for improving the performance on most of the task-specific datasets. In conclusion, the best way to achieve the best performance for a specific dataset is to pre-train the model in multiple datasets and then fine tune it on the specific dataset.

ACKNOWLEDGMENT

We would like to thank Ting Li and Xizhou Hong (Ruijin Hospital) who have helped us very much in data fetching and data cleansing. This work is supported by the National Key R&D Program of China for "Precision Medical Research" (No. 2018YFC0910500).

REFERENCES

[1] K.-i. Fukuda, T. Tsunoda, A. Tamura, T. Takagi *et al.*, "Toward information extraction: identifying protein names from biological papers," in *Pacific Symposium on Biocomputing*, vol. 707, no. 18, 1998, pp. 707–718.

[2] Y. Wang, J. Patrick, G. Miller, and J. O'Halloran, "Linguistic mapping of terminologies to SNOMED CT," in *Semantic Mining Conference on SNOMED CT*. Citeseer, 2006.

[3] M. Song, H. Yu, and W.-S. Han, "Developing a hybrid dictionary-based bio-entity recognition technique," *BMC Medical Informatics and Decision Making*, vol. 15, no. 1, p. S9, 2015.

[4] M. Topaz, K. Lai, D. Dowding, V. J. Lei, A. Zisberg, K. H. Bowles, and L. Zhou, "Automated identification of wound information in clinical notes of patients with heart diseases: Developing and validating a natural language processing application," *International Journal of Nursing Studies*, vol. 64, pp. 25–31, 2016.

[5] H.-J. Tan, R. Clarke, K. Chamie, A. Kaplan, A. Chin, M. S. Litwin, C. Saigal, and A. S. Hackbarth, "Development and validation of an automated method for identifying patients undergoing radical cystectomy for bladder cancer using natural language processing," *Urology Practice*, vol. 4, 11 2016.

[6] A. P. Glaser, B. J. Jordan, J. Cohen, A. Desai, P. Silberman, and J. J. Meeks, "Automated extraction of grade, stage, and quality information from transurethral resection of bladder tumor pathology reports using natural language processing," *JCO Clinical Cancer Informatics*, vol. 2, pp. 1–8, 2018.

[7] J. T. Senders, A. V. Karhade, D. J. Cote, A. Mehrtash, N. Lamba, A. DiRisio, I. S. Muskens, W. B. Gormley, T. R. Smith, M. L. Broekman *et al.*, "Natural language processing for automated quantification of brain metastases reported in free-text radiology reports," *JCO Clinical Cancer Informatics*, vol. 3, pp. 1–9, 2019.

[8] R. Bill, S. Pakhomov, E. S. Chen, T. J. Winden, E. W. Carter, and G. B. Melton, "Automated extraction of family history information from clinical notes," in *AMIA Annual Symposium Proceedings*, vol. 2014. American Medical Informatics Association, 2014, p. 1709.

[9] E. Iqbal, R. Mallah, D. Rhodes, H. Wu, A. Romero, N. Chang, O. Dzahini, C. Pandey, M. Broadbent, R. Stewart *et al.*, "ADEPt, a semantically-enriched pipeline for extracting adverse drug events from free-text electronic health records," *PloS one*, vol. 12, no. 11, p. e0187121, 2017.

[10] B. Fonferko-Shadrach, A. S. Lacey, A. Roberts, A. Akbari, S. Thompson, D. V. Ford, R. A. Lyons, M. I. Rees, and W. O. Pickrell, "Using natural language processing to extract structured epilepsy data from unstructured clinic letters: development and validation of the ExECT (extraction of epilepsy clinical text) system," *BMJ Open*, vol. 9, no. 4, p. e023232, 2019.

[11] J. Qiu, Q. Wang, Y. Zhou, T. Ruan, and J. Gao, "Fast and accurate recognition of chinese clinical named entities with residual dilated convolutions," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2018, pp. 935–942.

[12] Q. Wang, Y. Zhou, T. Ruan, D. Gao, Y. Xia, and P. He, "Incorporating dictionaries into deep neural networks for the chinese clinical named entity recognition," *Journal of Biomedical Informatics*, vol. 92, p. 103133, 2019.

[13] J. Qiu, Y. Zhou, Q. Wang, T. Ruan, and J. Gao, "Chinese clinical named entity recognition using residual dilated convolutional neural network with conditional random field," *IEEE Transactions on NanoBioscience*, vol. 18, no. 3, pp. 306–315, July 2019.

[14] A. O. Muis and W. Lu, "Weak semi-markov CRFs for NP chunking in informal text," *arXiv preprint arXiv:1810.08567*, 2018.

[15] A. L. Meyers, Y. He, Z. Glass, J. Ortega, S. Liao, A. Grieve-Smith, R. Grishman, and O. Babko-Malaya, "The termolator: terminology recognition based on chunking, statistical and search-based scores," *Frontiers in Research Metrics and Analytics*, vol. 3, p. 19, 2018.

[16] M. D. Anderson and D. Vilares, "Increasing NLP parsing efficiency with chunking," *Multidisciplinary Digital Publishing Institute Proceedings*, vol. 2, no. 18, p. 1160, 2018.

[17] A. Kemos, H. Adel, and H. Schütze, "Neural semi-markov conditional random fields for robust character-based part-of-speech tagging," *arXiv preprint arXiv:1808.04208*, 2018.

[18] Y. Xiong, Z. Wang, D. Jiang, X. Wang, Q. Chen, H. Xu, J. Yan, and B. Tang, "A fine-grained chinese word segmentation and part-of-speech tagging corpus for clinical text," *BMC Medical Informatics and Decision Making*, vol. 19, no. 2, p. 66, 2019.

[19] W. Khan, A. Daud, K. Khan, J. A. Nasir, M. Basheri, N. Aljohani, and F. S. Alotaibi, "Part of speech tagging in urdu: Comparison of machine and deep learning approaches," *IEEE Access*, vol. 7, pp. 38 918–38 936, 2019.

[20] T. Kilias, A. Löser, F. Gers, Y. Zhang, R. Koopmanschap, and M. Kersten, "IDEL: In-database neural entity linking," in *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 2019, pp. 1–8.

[21] A. Sil, G. Kundu, R. Florian, and W. Hamza, "Neural cross-lingual entity linking," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[22] J. R. Raiman and O. M. Raiman, "Deeptype: multilingual entity linking by neural type system evolution," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[23] Q. Wang, J. Qiu, Y. Zhou, T. Ruan, D. Gao, and J. Gao, "Automatic severity classification of coronary artery disease via recurrent capsule network," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2018, pp. 1587–1594.

[24] Q. Wang, C. Xu, Y. Zhou, T. Ruan, D. Gao, and P. He, "An attention-based Bi-GRU-CapsNet model for hypernymy detection between compound entities," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2018, pp. 1031–1035.

[25] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[26] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, Jun. 2018, pp. 2227–2237.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[28] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[29] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: pre-trained biomedical language representation model for biomedical text mining," *arXiv preprint arXiv:1901.08746*, 2019.
[30] D. Kim, J. Lee, C. H. So, H. Jeon, M. Jeong, Y. Choi, W. Yoon, M. Sung, and J. Kang, "A neural named entity recognition and multi-type normalization tool for biomedical text mining," *IEEE Access*, vol. 7, pp. 73 729–73 740, 2019.
[31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
[32] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1449–1457.
[33] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2414–2423.
[34] M. Moghimi, S. J. Belongie, M. J. Saberian, J. Yang, N. Vasconcelos, and L.-J. Li, "Boosted convolutional neural networks." in *BMVC*, 2016, pp. 24–1.
[35] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. [Online]. Available: https://www.aclweb.org/anthology/D16-1264
[36] Y. Liu, Y. Zhou, S. Wen, and C. Tang, "A strategy on selecting performance metrics for classifier evaluation," *International Journal of Mobile Computing and Multimedia Communications (IJMCMC)*, vol. 6, no. 4, pp. 20–35, 2014.
[37] F. Chollet *et al.*, "Keras," https://keras.io, 2015.
[38] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: a system for large-scale machine learning." in *OSDI*, vol. 16, 2016, pp. 265–283.
[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.