Extracting Structural Knowledge for Professional Text Inference

Tianyu Xia[1], Jian Wang[1], Tianyuan Liu[1], Hailan Jiang[2], and Yuqing Sun[1(✉)]

[1] School of Software, Shandong University, Jinan, China
937885834@qq.com, wangjian026@126.com, zodiacg@foxmail.com, sun_yuqing@sdu.edu.cn
[2] Shandong Polytechnic, Jinan, China
1792@sdp.edu.cn

Abstract. Grading subjective questions of specialty text is a kind of text inference task. Since there are many specialty terms and concepts, it is difficult to judge the knowledge contained in a text as the usual way on inferring a general text. In this paper, we propose a specialty text inference model by extracting the structural knowledge from text. We first propose a knowledge graph construction method for the extraction of knowledge from specialty texts. By combining the constructed knowledge features with the text semantic features, we design the specialty text inference model. Finally, we use real datasets from a national professional exam to validate the soundness of the knowledge graph construction method and the performance of the inference model. The experiments under different training set sizes and network structures are also conducted to detailly analyze the design of our method. The experimental results show the effectiveness and practicality of our approach.

Keywords: Specialty Text · Structural Knowledge · Text Inference

1 Introduction

The specialty text inference task is the analysis and reasoning of specialty texts according to the given reference text, which is widely used the subjective questions, medical diagnosis, legal case analysis. Since there are many specialty terms and concepts, it is difficult to judge the knowledge contained in a text as the usual way on inferring a general text.

To solve this problem, the large pre-trained language models are used in the inference task. Since training these models are mostly on the general corpus, they are often not applicable for the specialty text. Researches indicate that fine-tuning large pre-trained language models with domain-specific datasets can enhance the performance [15]. In recent years, researchers have applied the above

This work was supported by the National Nature Science Foundation of China, NSFC (62376138) and the Innovative Development Joint Fund Key Projects of Shandong NSF (ZR2022LZH007).

Y. Sun et al. (Eds.): ChineseCSCW 2023, CCIS 2013, pp. 334–347, 2024.
https://doi.org/10.1007/978-981-99-9640-7_25

methods to the automatic grading of subjective questions and achieved good results [10]. But most of the existing methods focus on the text semantics without considering the knowledge in an explicit way [9].

In this paper, we propose a specialty text inference method based on structural knowledge extraction. Firstly, we propose a knowledge graph construction method for specialty texts, enabling structural extraction of knowledge. The knowledge graph consists of specialty elements like terms, entities, and important general words, along with their relationships. This kind of knowledge graph represents the key point of specialty knowledge in text in an understandable way that is used for the subsequent text inference.

Secondly, we design a specialty text inference model based on structural knowledge extraction, called KnowSTI. We construct the knowledge features and semantic features of the text. And then we use the consistency loss function to train the model, which combines these two features for text inference. Additionally, the graph provides an explainable way for the inference results.

The rest of this paper is organized as follows. Section 2 introduces related works. Section 3 introduces the knowledge graph construction method. We present the details of our model in Sect. 4. We validate our model on real datasets and analyze the experimental results in Sect. 5. Section 6 summaries this paper and presents the future work.

2 Related Work

2.1 Specialty Text Inference

The large pre-trained language models are used in the inference task [7,12]. Since training these models are mostly on the general corpus, they are often not applicable for the specialty text. Lee et al. [8] found these model are difficult to estimate their performance on datasets containing biomedical texts, they created BioBERT from BERT pre-trained on a biomedical corpus.

Data augmentation is a technique for specialty text inference. It extends datasets to improve the model performance of large pre-trained language models on specialty texts. Classical data enhancement techniques include methods such as EDA [18], back translation [14], mixup [20]and text generation [11]. Ding et al. [2] studied how to infer patient condition from the description text with the mixup data augmentation method. Lun et al. [13] utilized the BERT pre-trained language model to review subjective questions and experimented with various data enhancement methods to boost model performance. On the same task, Li et al. [10] employed back-translation for data enhancement. However, these methods lack an inferential basis for the results, leading to untrusted outcomes in practical scenarios and impacting model usability.

2.2 Knowledge Extraction on Specialty Text

Knowledge extraction includes named entity recognition (NER) and relation extraction (RE). For specialty scenarios, pre-trained models for NER need adaptation. Jia et al. [6] used a language model as a concomitant task for NER in

a new specialized domain using multi-task learning. Wu et al. [19] introduced the LTN model to reuse knowledge from a general NER task without refactoring. Many methods also use remotely supervised technique to address the lack of training data. Gu et al. [4] used the Comparative Toxicogenomics Database (CTD) for remote supervision, while Di et al. [1] optimized relational mention representation by selecting suitable knowledge bases and associated texts.

Researchers have also explored the open relation extraction task, aiming to extract relations without predefined types. Some methods extract features from entity pairs and then identify inter-entity relations through clustering [5,17]. There are also supervised methods which use labeled data with predefined relations to guide extraction in unsupervised data [3,21]. While the proposed technique overcomes the limitation of undefined relation extraction, its performance still requires improvement and may not be suitable for real scenarios.

3 Constructing the Knowledge Graph from Text

We propose the concept *knowledge graph* to describe the knowledge contained in a specialty text. For a given the student text $X = x_1 x_2 \ldots x_{|X|}$, the knowledge graph is denoted by $G_X = (V_X, E_X)$, where the node set $V_X = \{v_1, v_2, \ldots v_{k_X}\}$ refer to the specialty elements in the text, k_X is the total number of nodes, and $E_X = \{(v_i, v_j) \,|v_i, v_j \in V_X\}$ is the edge set. The proposed structural knowledge extraction method is shown in Fig. 1. The details are given below.

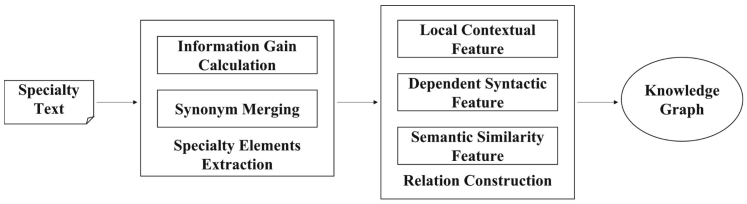

Fig. 1. A Knowledge Graph Construction Method for Specialty Texts

3.1 Expert Rule Based Specialty Elements Extraction

We choose the specific terms, entities, and important general words as the specialty elements, which are essential for understanding the specialty text and constructing the knowledge graph. Since information gain (IG for short) is often used to measure the importance of features, we use IG to calculate the importance of specialty elements. Let $H(C)$ denote the overall entropy of the predicated score. $H(C\,|T)$ denotes the conditional entropy, given the presence or absence of T: For word T, the information gain $IG(T)$ is computed as follow.

$$IG(T) = H(C) - H(C\,|T) \tag{1}$$

For the purpose of synonym merging, the information gain value $IG\left(T^{'}\right)$ of a synonymous specialty element T is factored into the information gain value $IG(T)$ of the word T in calculating the information gain:

$$IG(T) = \sum_{T^{'} \in V_T} IG\left(T^{'}\right) \tag{2}$$

where V_T denotes the set of synonymous specialty elements of the word T. Finally, the information gain values $IG(T)$ that have completed the synonym merging are sorted. The elements N_{ig} with the highest $IG(T)$ value constitute the specialty element table. These elements constitute as the set of nodes of the knowledge graph $V_X = \{v_1, v_2, \ldots v_{k_X}\}$, k_X is the total number of specialty elements in the professional text, based on which we to extract the specialty elements in text X.

3.2 Relation Construction Based on Multi-information Fusion

To obtain structural knowledge representation, we combine the local contextual features, dependent syntactic features, and semantic similarity features of the two specialty elements in the specialty text as the weights of their relation.

The local contextual feature $d_{ij}^{c} = 1$ is defined as the concurrence of two specialty elements v_i and v_j within a given widow c_n, which is used to capture the short-distance interactions. $d_{ij}^{c} = 0$ for otherwise.

The dependency syntactic feature d_{ij}^{n} focuses on the dependency relation between two specialty elements. We use stanza to parse an input sentence to a dependency syntax tree. If there is a parent-child relation between specialty elements v_i and v_j, $d_{ij}^{n_1} = 1$; otherwise, it is 0.

In order to better utilize the inter-lexical dependency information contained in the dependency syntax tree, we also consider the indirect associations $d_{ij}^{n_2}$ between two specialty elements, including the following three cases:

$$d_{ij}^{n_2} = \begin{cases} 1, \text{There is a grandchild relationship between } v_i \text{ and } v_j \\ 1, v_i \text{ and } v_j \text{are brother nodes} \\ 0, \text{otherwise} \end{cases} \tag{3}$$

The two features $d_{ij}^{n_1}$ and $d_{ij}^{n_2}$ consist of the syntactic feature d_{ij}^{n}:

$$d_{ij}^{n} = \left[d_{ij}^{n_1} : d_{ij}^{n_2}\right] \tag{4}$$

We use BERT, a pre-trained language model, to encode the text X into a sequence of vectors $\mathbf{C} = [\mathbf{c1}, \mathbf{c2}, \ldots, \mathbf{cn}]$, where each element v_i corresponds to vector c_i. The semantic similarity feature d_{ij}^{s} is computed as the cosine value between vectors c_i and c_j:

$$d_{ij}^{s} = cos\left(\mathbf{c_i}, \mathbf{c_j}\right) = \frac{\mathbf{c_i} \bullet \mathbf{c_j}}{|\mathbf{c_i}| \times |\mathbf{c_j}|} \tag{5}$$

Then the above features are combined for an edge e_{ij}, namely the local context features d_{ij}^{c}, the syntactic features d_{ij}^{n}, and the semantic similarity features d_{ij}^{s} for the specialty elements v_i and v_j in X:

$$\mathbf{d_{ij}} = \left[d_{ij}^{c}, d_{ij}^{n}, d_{ij}^{s}\right] \tag{6}$$

The weight e_{ij} of the relation between specialty elements v_i and v_j in the adjacency matrix is computed by multiplying these vectors with the weight vector $\mathbf{w} = [\alpha_p, \beta_p, \gamma_p]$ for information fusion. The parameter matrix w is trained in the subsequent task. The relation between the specialty elements in text X forms the edges in the knowledge graph $E_X = \{(v_i, v_j) \,|v_i, v_j \in V_X\}$.

$$e_{ij} = \mathbf{w}^T \bullet \mathbf{d_{ij}} = \alpha_p d_{ij}^{c} + \beta_p d_{ij}^{n} + \gamma_p d_{ij}^{s} \tag{7}$$

4 Structural Knowledge Based Specialty Text Inference

We propose a lightweight specialty text inference model based on structural knowledge. The specialty text inference task involves a subjective question Q, candidate text X, and reference answer text R. We extract knowledge graphs G_X and G_R using the model and make inferential scores $y \in S$ for X. S is the set of inference results with $|S|$ possible scores. G_X serves as the knowledge that explains the inferred results.

4.1 Model

We propose KnowSTI, a specialty text inference model based on structural knowledge extraction. The model takes candidates' texts and reference answer as inputs and outputs the inference results, as shown in Fig. 2.

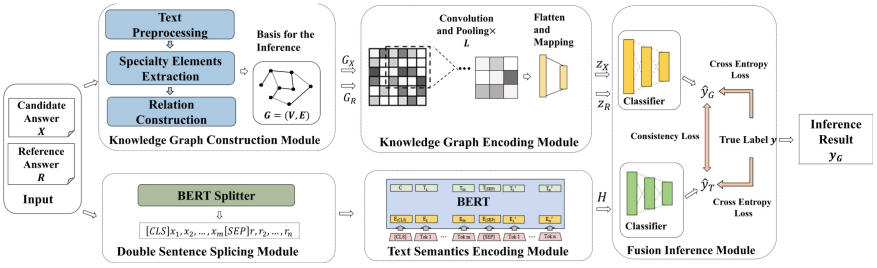

Fig. 2. Structural Knowledge Based Specialty Text Inference

4.2 Knowledge Graph Self-encoder

A knowledge graph is formed as an adjacency matrix A_X. We use a convolutional neural network with L layers for encoding A_X as the vector z_X. Let $M^0 = A_X$ and M^l denotes the representation matrix of the lth hidden layer. Formally,

$$M^l = MaxPooling\left(Conv2D\left(M^{l-1}, K^{l-1}\right)\right) \tag{8}$$

Then the $Flatten$ operation is applied to M_L and then M_L is downscaled using a multilayer perceptron to obtain the vector z_X for G_X:

$$\mathbf{z_X} = MLP_{enc}\left(Flatten\left(M_L\right)\right) \tag{9}$$

The decoder is MLP network that accepts z_X as input and outputs the graph matrix $\hat{A}_X$:

$$\hat{A}_X = MLP_{dec}\left(\mathbf{z_X}\right) \tag{10}$$

The mean square error between $\hat{A}_X$ and A_X is used as a loss function to train the encoder:

$$J = \frac{1}{N^2}\left|\left|\hat{A}_X - A_X\right|\right|_2^2 \tag{11}$$

4.3 Structural Knowledge and Semantics Based Text Inference

We use BERT to encode $X = x_1, x_2 \ldots x_m$ and the reference answer $R = r_1, r_2 \ldots r_n$, i.e. $H = BERT\left(R \oplus X\right)_{\{CLS\}}$. To consider both differences and similarities, we introduce the feature differences $|\mathbf{z_X} - \mathbf{z_R}|$ and feature correlation $\mathbf{z_X} \otimes \mathbf{z_R}$. Here, $\otimes$ denotes the outer product of two vectors, reflecting their similarity and interaction, and $\oplus$ denotes the splicing operation on the vectors. The knowledge feature Z is obtained after splicing:

$$\mathbf{Z} = \mathbf{z_X} \oplus \mathbf{z_R} \oplus |\mathbf{z_X} - \mathbf{z_R}| \oplus (\mathbf{z_X} \otimes \mathbf{z_R}) \tag{12}$$

Then the knowledge-based classifier MLP_G yields a probability distribution of inferred outcomes $\hat{y}_G$. The loss function MLP_G is cross entropy.

$$\hat{y}_G = softmax\left(MLP_G\left(\mathbf{Z}\right)\right) \tag{13}$$

$$J_1 = -\sum_{i=1}^{|S|} y_i log\left(\hat{y}_{G,i}\right) \tag{14}$$

where $|S|$ is the number of inferred outcome categories, and $\hat{y}_{G,i}$ is the probability for category i.

By using the text semantic features, the inference probability distribution $\hat{y}_T$ is obtained by the text semantics-based classifier MLP_T:

$$\hat{y}_T = softmax\left(MLP_T\left(\mathbf{H}\right)\right) \tag{15}$$

$$J_2 = -\sum_{i=1}^{|S|} y_i log\left(\hat{y}_{T,i}\right) \tag{16}$$

Then KL divergence is used to constrain these two classifiers, aiming for a consistent results:

$$J_{con} = KL\left(\hat{y}_G \,||\hat{y}_T\,\right) \tag{17}$$

Finally, the loss function is defined as:

$$L = \alpha_l J_1 + \beta_l J_2 + \left(1 - \alpha_l - \beta_l\right) J_{con} \tag{18}$$

where α_l and β_l are hyper-parameters. Besides, the Knowledge graph can be regarded as the interpreter for the inferred results, where the nodes and edges in the graph present the important words and relations.

5 Experiment

5.1 Dataset and Evaluation

We adopt eight subjective question datasets, which are selected from a national professional qualification examination. Each dataset includes the question title, reference answer, examinee's answer text, and corresponding score. The data is divided into training, validation, and test sets with ratios of 70%, 20% and 10%, respectively. To assess the model performance with varying training set sizes, training sets of 5% and 1% of the dataset are also constructed to simulate limited samples, as shown in Table 1. In the experiments, the accuracy is used as the overall evaluation metric.

Table 1. Amount of data in different settings

Category	Dataset							
	I	II	III	IV	V	VI	VII	VIII
70%training set	11104	11002	11084	11009	4199	9916	9764	9935
5%training set	793	786	792	786	300	708	698	710
1%training set	159	157	158	157	60	142	140	142
Test set	3173	3144	3167	3146	1200	2834	2790	2839
Validation set	1587	1572	1584	1573	600	1417	1396	1420
Total	15864	15718	15835	15728	5999	14167	13950	14194

5.2 Comparison Models

As grading subjective questions is a kind of text inference task, we choose several text inference models for comparison.

- **Base-BERT** [7]: Text is encoded with the pre-trained language model BERT and the encoding result is forwarded to a classifier.
- **Base-RoBERTa** [12]: Text is encoded with the pre-trained language model RoBERTa, followed by a classifier for grading.
- **LR+** [15]: The pre-trained BERT model is enhanced by fine-tuning it on textbooks as a specialty corpus. It encodes the text and reference answer, with a classifier for predicting the scores.
- **Conv-GRNN** [16]: The model encodes examinee text sentences using a convolutional neural network at the vocabulary level, generating document vectors with GRU at the sentence level for classification.
- **KnowSTI**: The model proposed in this paper.

5.3 Setting

In the knowledge graph construction process, the specialty element table size is selected from $\{30, 50, 100, 200\}$. Semantic similarity feature is calculated with BERT-base. The word window's size is set 2 and 7. The initial value for $\mathbf{W_e}$ is $[0.15, 0.15, 0.25, 0.25, 0.2]$. The knowledge graph encoder uses a 3 layer CNN with convolution kernel sizes of 5, 2 and 3, respectively, while the pooling kernel sizes are all set 2. The encoding vector dimension size is 800 and the decoder is a 2-layer MLP. The specialty text classifier uses a two-layer MLP with a hidden layer vector dimension of 1000. The RoBERTa-wmm pre-training language model is used for semantic encoding.

During training, the model uses a learning rate of 4e-4, a batch size of 32, and AdamW as the optimizer. The model has about 950K trainable parameters. In the use phase, the number of parameter in the model is about 650K, making it faster and more efficient compared to natural language models with billions of parameters. This approach can be trained and tested on a single Nvidia GeForce RTX 2080Ti, making it suitable for closed grading scenarios. Therefore, this model is highly applicable in real scenarios.

5.4 Comparison Results

We compare our method with baselines and the results on different sizes of training sets are shown in Table 2, Table 3, and Table 4, where 1%, 5%, and 70% denote the proportion of the training set to the total sample size.

The results show that our model consistently outperforms comparison methods, highlighting its effectiveness. LR+ slightly improves over the original BERT model, indicating the impact of parameter fine-tuning. Our method consistently outperforms LR+, showcasing knowledge graph-based specialty text inference's value. Requiring less training data and a smaller model size than pre-trained language models, our method is more practical.

Table 2. Accuracy of each model on the 70% training set

Model	Dataset							
	I	II	III	IV	V	VI	VII	VIII
Conv-GRNN	89.17%	83.82%	89.38%	89.54%	90.22%	86.39%	87.32%	89.28%
Base-BERT	95.19%	93.64%	90.51%	90.38%	93.81%	93.21%	91.43%	93.53%
Base-RoBERTa	96.87%	94.76%	94.75%	91.24%	94.97%	95.44%	94.00%	95.88%
LR+	96.03%	96.45%	96.66%	95.09%	96.54%	95.53%	94.63%	96.22%
KnowSTI	**97.28%**	**96.87%**	**97.92%**	**95.47%**	**97.07%**	**95.68%**	**96.83%**	**97.06%**

Table 3. Accuracy of each model on the 5% training set

Model	Dataset							
	I	II	III	IV	V	VI	VII	VIII
Conv-GRNN	83.40%	80.51%	88.74%	84.20%	89.97%	85.49%	86.50%	84.84%
Base-BERT	85.85%	82.97%	90.25%	86.54%	89.96%	90.57%	88.27%	89.44%
Base-RoBERTa	96.05%	94.36%	95.11%	89.89%	92.34%	91.00%	90.66%	90.38%
LR+	**96.79%**	93.00%	96.31%	91.09%	**96.03%**	94.89%	**93.55%**	91.24%
KnowSTI	96.51%	**94.46%**	**96.82%**	**93.90%**	94.57%	**95.04%**	92.83%	**93.06%**

Table 4. Accuracy of each model on the 1% training set

Model	Dataset							
	I	II	III	IV	V	VI	VII	VIII
Conv-GRNN	81.59%	80.65%	87.67%	84.03%	88.12%	84.61%	85.94%	82.51%
Base-BERT	90.50%	86.72%	91.90%	84.48%	85.71%	90.15%	83.95%	87.90%
Base-RoBERTa	92.27%	89.87%	93.00%	87.77%	84.80%	90.46%	84.00%	86.91%
LR+	**92.40%**	91.39%	**94.16%**	92.38%	91.15%	90.85%	90.13%	88.57%
KnowSTI	92.28%	**92.24%**	93.88%	**93.30%**	**92.69%**	**92.37%**	**91.55%**	**89.95%**

In real grading scenarios, training data is limited. We test our method's robustness by comparing it with others across different training set sizes. Our model achieves over 95% accuracy with a 70% training set and around 90% accuracy with 1% or 5% training sets. Notably, as shown in Table 3, our model attains over 95% accuracy for datasets I, III, and VI with a 5% training set, illustrating its capacity to excel with limited data.

5.5 Results of Ablation Experiments

This section conducts an in-depth analysis of the model's structure and performance, evaluating the contribution of each module. The experiments include three variants of our method:

- **Without reference answer**: The model takes only the student text as input, and during inference, it relies only on the encoded text for inference.
- **Without knowledge graph encoding**: The model uses only the semantic encoding of the text for inference.

Table 5. Model ablation experiments on specialty subjective datasets

Model structure	Dataset							
	I	II	III	IV	V	VI	VII	VIII
No reference answer	90.00%	92.64%	89.75%	87.71%	86.19%	91.53%	94.63%	94.44%
No KG encoding	96.88%	96.19%	97.32%	95.25%	96.81%	94.29%	95.98%	96.93%
No semantic encoding	97.15%	96.23%	97.80%	95.00%	97.00%	95.15%	96.80%	96.85%
KnowSTI	**97.28%**	**96.87%**	**97.92%**	**95.47%**	**97.07%**	**95.68%**	**96.83%**	**97.06%**

– **Without semantic encoding**: The model utilizes only knowledge graph encoding for inference.

The results in Table 5 show that omitting any module reduces the performance. The proposed model *KnowSTI* outperforms the cases without knowledge graph encoding or text semantic encoding module. Thus fusing structural knowledge and text semantics enhances the performance. Especially, omitting reference answers notably degrades the performance, which illustrates that the reference answer helps model understand the text.

5.6 Analysis on Knowledge Graph Construction Methods

In order to illustrate the effectiveness of the knowledge graph construction method, we have experimentally verified the role of each step in the knowledge graph construction method.

Impact of the Size of Specialty Element Table. As the nodes of knowledge graph are based on the element table, we explore how the size of specialty element table impacts the performance. The results in Table 6 show that the moderate number of specialty elements are desired. The performance approaches the best at above 50 for most datasets. It is necessary to select the based on data characteristics.

Table 6. Performance with different sizes of specialty element tables

Size	Dataset							
	I	II	III	IV	V	VI	VII	VIII
30	96.34%	93.63%	95.37%	93.71%	96.94%	95.53%	94.63%	96.96%
50	**97.28%**	**96.87%**	97.92%	**95.47%**	97.07%	**95.68%**	96.83%	97.06%
100	97.10%	93.56%	**98.00%**	94.81%	97.49%	95.31%	**97.05%**	97.14%
200	94.56%	93.74%	95.19%	94.00%	**97.57%**	95.44%	96.81%	**97.33%**

The Method on Constructing Element Relation. Then we assess how the parameters on constructing the relations of knowledge graph influence the model performance, namely d_{ij}^{c}, d_{ij}^{n}, and d_{ij}^{s}. Specifically, we construct specialty text knowledge graphs using one or two features to establish relations among elements. The specialty text inference model is then trained and tested on these graphs to gauge feature contributions.

In Table 7, combining all three features performs best in our method. Among these, the syntactic features are very important as it effectively captures both the syntactic and semantic information. Besides, combining syntactic and semantic similarity features also enhance knowledge graph construction. These results show the importance of long-distance dependencies among specialty elements.

Table 7. Performance with different relation construction features

Type	Dataset							
	I	II	III	IV	V	VI	VII	VIII
c	91.33%	89.14%	92.25%	91.89%	92.59%	90.54%	92.58%	91.00%
n	94.73%	92.06%	91.88%	92.61%	95.51%	90.48%	94.52%	96.52%
s	91.89%	90.87%	93.96%	92.54%	93.99%	92.36%	92.31%	94.81%
cn	95.41%	92.42%	95.15%	94.91%	95.87%	93.57%	94.69%	96.79%
cs	93.02%	90.91%	94.28%	93.85%	94.44%	94.16%	93.12%	95.50%
ns	94.85%	93.53%	96.51%	95.04%	96.12%	94.58%	95.44%	96.42%
cns	**97.28%**	**96.87%**	**97.92%**	**95.47%**	**97.07%**	**95.68%**	**96.83%**	**97.06%**

The Effect of Synonym Merging on Extracting Specialty Elements. We examine the effect of synonym merging on model performance during specialty elements extraction by creating knowledge graphs with and without this process. The results in Table 8 indicate a significant performance drop, particularly in dataset I, where the absence of synonym merging leads to a potential 10% decrease. In professional exams, the expressions vary from candidate to candidate. Synonym merging unifies synonymous elements under one node, enhancing both accuracy and recall in specialty element extraction.

Table 8. Comparison of model performances with/without synonym merging

Synonym merging	Dataset							
	I	II	III	IV	V	VI	VII	VIII
w/o	87.32%	89.64%	96.75%	90.19%	93.46%	86.86%	91.33%	90.35%
w	**97.28%**	**96.87%**	**97.92%**	**95.47%**	**97.07%**	**95.68%**	**96.83%**	**97.06%**

Distribution Analysis of Specialty Elements Under Different Training Set Sizes. To examine the effect of training set size on constructing specialty elements tables through information gain, a dataset was randomly selected. Information gain was calculated under various training set proportions, and the 50 words with the highest IG in the 70% proportion training set are selected. Scatter points of different colors indicate information gain across different scaled datasets. The Fig. 3 highlights a consistent trend in information gain values between words across various datasets, with minor variations in individual values. This method of selecting specialty elements based on information gain maintains stability regardless of training set size, ensuring a consistent table construction process.

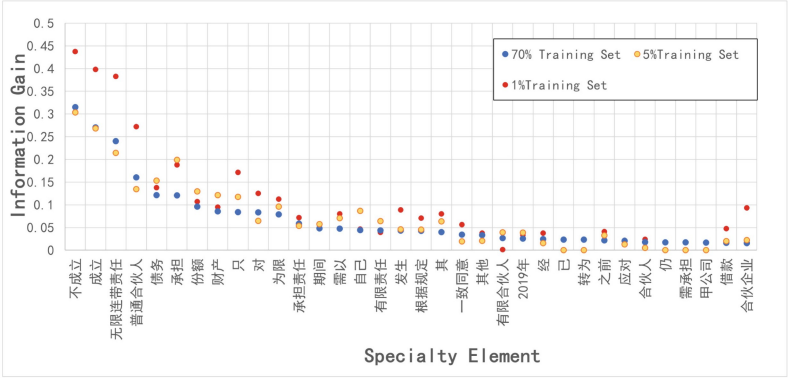

Fig. 3. Information gain values of specialty elements under different training set sizes

6 Conclusion

For the task of grading subjective questions in professional examinations, we propose a structural knowledge based specialty text inference model. We first design a knowledge graph construction method to extract structural knowledge from the specialty text, based on which the specialty text inference model is designed with the hybrid loss functions. We verify our model on real datasets from a national professional exam with different training set sizes. The ablation experiments are conducted to verify the validity of the components of the model. We also design a series of experiments to show the effectiveness of our knowledge graph construction method. In the future, we are planning to generate reusable knowledge when grading, which will enable model to provide feedback to the user, increasing the practicality of the automatic grading.

References

1. Di, S., Shen, Y., Chen, L.: Relation extraction via domain-aware transfer learning. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1348–1357 (2019)
2. Ding, X., Lybarger, K., Tauscher, J., Cohen, T.: Improving classification of infrequent cognitive distortions: domain-specific model vs. data augmentation. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Student Research Workshop, pp. 68–75 (2022)
3. Duan, B., Wang, S., Liu, X., Xu, Y.: Cluster-aware pseudo-labeling for supervised open relation extraction. In: Proceedings of the 29th International Conference on Computational Linguistics, pp. 1834–1841 (2022)
4. Gu, J., Sun, F., Qian, L., Zhou, G.: Chemical-induced disease relation extraction via attention-based distant supervision. BMC Bioinform. **20**, 1–14 (2019)
5. Hu, X., Zhang, C., Xu, Y., Wen, L., Yu, P.S.: Selfore: self-supervised relational feature learning for open relation extraction. arXiv preprint arXiv:2004.02438 (2020)
6. Jia, C., Liang, X., Zhang, Y.: Cross-domain NER using cross-domain language modeling. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 2464–2474 (2019)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, vol. 1, p. 2 (2019)
8. Lee, J., et al.: Biobert: a pre-trained biomedical language representation model for biomedical text mining. Bioinformatics **36**(4), 1234–1240 (2020)
9. Li, D., Liu, T., Pan, W., Liu, X., Sun, Y., Yuan, F.: Grading Chinese answers on specialty subjective questions. In: Sun, Y., Lu, T., Yu, Z., Fan, H., Gao, L. (eds.) ChineseCSCW 2019. CCIS, vol. 1042, pp. 670–682. Springer, Singapore (2019). https://doi.org/10.1007/978-981-15-1377-0_52
10. Li, Z., Tomar, Y., Passonneau, R.J.: A semantic feature-wise transformation relation network for automatic short answer grading. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 6030–6040 (2021)
11. Liu, D., et al.: Tell me how to ask again: question data augmentation with controllable rewriting in continuous space. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 5798–5810 (2020)
12. Liu, Y., et al.: Roberta: a robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
13. Lun, J., Zhu, J., Tang, Y., Yang, M.: Multiple data augmentation strategies for improving performance on automatic short answer scoring. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 13389–13396 (2020)
14. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 86–96 (2016)
15. Sung, C., Dhamecha, T., Saha, S., Ma, T., Reddy, V., Arora, R.: Pre-training bert on domain resources for short answer grading. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6071–6075 (2019)

16. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1422–1432 (2015)
17. Tran, T.T., Le, P., Ananiadou, S.: Revisiting unsupervised relation extraction. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7498–7505 (2020)
18. Wei, J., Zou, K.: EDA: easy data augmentation techniques for boosting performance on text classification tasks. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 6382–6388 (2019)
19. Wu, J., Liu, T., Sun, Y., Gong, B.: A light transfer model for Chinese named entity recognition for specialty domain. In: Sun, Y., Liu, D., Liao, H., Fan, H., Gao, L. (eds.) ChineseCSCW 2020. CCIS, vol. 1330, pp. 530–541. Springer, Singapore (2021). https://doi.org/10.1007/978-981-16-2540-4_38
20. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: beyond empirical risk minimization. arXiv preprint arXiv:1710.09412 (2017)
21. Zhao, J., Gui, T., Zhang, Q., Zhou, Y.: A relation-oriented clustering method for open relation extraction. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pp. 9707–9718 (2021)