Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

# Automatic ICD-10 coding: Deep semantic matching based on analogical reasoning

Yani Chen[a], Han Chen[b], Xudong Lu[a], Huilong Duan[a], Shilin He[b,**], Jiye An[a,*]

[a] College of Biomedical Engineering and Instrument Science, Zhejiang University, Zheda Road, 310027 Hanghzou, Zhejiang Province, China
[b] Department of Information, Hainan Hospital of Chinese PLA General Hospital, Haitang Bay, 572013 Sanya, Hainan Province, China

ARTICLE INFO

*Keywords:*
Automatic coding
ICD-10
Semantic matching
Analogical reasoning

ABSTRACT

*Background:* ICD-10 has been widely used in statistical analysis of mortality rates and medical reimbursement. Automatic ICD-10 coding is desperately needed because manually assigning codes is expensive, time-consuming, and labor-intensive. Diagnoses described in medical records differ significantly from those used in ICD-10 classification, making it impossible for existing automatic coding techniques to perform well enough to support medical billing, resource allocation, and research requirements. Meanwhile, most of the current automatic coding approaches are oriented toward English ICD-10. This method for automatically assigning ICD-10 codes to diagnoses extracted from Chinese discharge records was provided in this paper.
*Method:* First, BERT creates word representations of the two texts. Second, the context representation layer incorporates contextual information into the representation of each time step of the word representations using a bidirectional Long Short-Term Memory. Third, the matching layer compares each contextual embedding of the uncoded diagnosis record against a weighted version of all contextual character embeddings of the manually coded diagnosis record. The matching strategy is element-wise subtraction and element-wise multiplication and then through a neural network layer. Fourth, the matching vectors are combined using a one-layer convolutional neural network. A sigmoid is then used to output matching results.
*Results:* To evaluate the proposed method, 1,003,558 manually coded primary diagnoses were gathered from the homepage of the discharge medical records. The experimental results showed that the proposed method outperformed popular deep semantic matching algorithms, such as DSSM, ConvNet, ESIM, and ABCNN, and demonstrated state-of-the-art results in a single text matching with an accuracy of 0.986, a precision of 0.979, a recall of 0.983, and an F1-score of 0.981.
*Conclusion:* The automatic ICD-10 coding of Chinese diagnoses is successful when using the proposed deep semantic matching approach based on analogical reasoning.

## 1. Introduction

ICD-10 [1] has been widely used in the statistical analysis of mortality rates and medical reimbursement. In China, ICD-10 codes are typically assigned to medical records in China by coders working in the hospital's medical record department [2]. The coders must

[*] Corresponding author. Zhejiang University, 866 Yuhangtang Road, Hangzhou, Zhejiang Province, 310058, China.
[**] Corresponding author. Hainan Hospital of Chinese PLA General Hospital, Haitang Bay, 572013 Sanya, Hainan Province, China.
*E-mail addresses:* heshilin301@163.com (S. He), an_jiye@zju.edu.cn (J. An).

https://doi.org/10.1016/j.heliyon.2023.e15570
Received 12 July 2022; Received in revised form 12 April 2023; Accepted 13 April 2023
Available online 19 April 2023

master the knowledge of coding rules, medical terminologies, and the field of medicine in order to complete the assignment [3]. Because of the complexity of the ICD-10 structure and the enormously increasing number of ICD-10 codes, ICD-10 coding work has become much more laborious and time-consuming, even if a coder with professional abilities takes approximately 30 min per case on average [4]. Considering these constraints, there is an urgent need to develop an accurate and effective automatic ICD-10 coding method.

There have been numerous studies on automatic and semi-automatic ICD coding as a solution to the massive amounts of human labor required for manual coding. Early research typically used supervised machine learning approaches. Boytcheva et al. used the multiclass support vector machines method for automatic mapping of ICD-10 codes to diagnoses extracted from discharge letters [5]. Koopman et al. trained support vector machine classifiers to automatically assign ICD-10 codes to cancer texts from death certificates [6]. However, it is difficult for machine learning methods to deal with high-noise and highly redundant electronic medical records affected by the diverse expressions of doctors.

Many recent approaches apply deep learning to automatic ICD coding [7]. Xie et al. built a neural architecture, including a tree-of-sequences LSTM, an adversarial learning approach, isotonic constraints, and attentional matching, for automated coding of discharge diagnosis in discharge summaries [8]. Baumel et al. presented HA-GRU, a hierarchical approach to tagging a document by identifying the sentences relevant for each label, for assigning multiple ICD codes to discharge summaries [9]. Duarte et al. leveraged a deep neural network that combines word embeddings, recurrent units, and neural attention for the assignment of ICD-10 codes for causes of death by analyzing free-text descriptions in death certificates [10]. The findings showed that the deep learning-based methods performed better than other traditional methods. However, most of the automatic coding studies based on deep learning are oriented towards English ICD-10, and the approaches for English text cannot be applied directly to Chinese text due to the differences in the linguistic feature [11].

There are still some studies based on Chinese ICD-10 published by the Statistical Information Center of the NHFPC of the People's Republic of China [2]. Yu et al. presented a MA-BiRNN model to assign disease codes of Chinese clinical notes [12]. Chen et al. presented an improved approach based on the LCS and semantic similarity for automatic Chinese diagnosis, mapping from the Chinese disease names given by clinicians to the disease names in ICD-10 [2]. However, the difference between the diagnosis records and the ICD-10 names is too large because ICD-10 is not a clinical nomenclature standard rather than a clinical classification standard. For example, the diagnosis record "烫伤残余创面5%躯干、左上下肢 (Scald residual wound on 5% trunk, left upper and lower limbs)" needs to be coded with "T31.001", whereas the corresponding ICD-10 name is "少于体表面积10%轻度烧伤(Burns involving less than 10% of body surface)". Therefore, the performance of current automatic coding still cannot meet the needs of medical billing, health resource allocation, and medical research.

A large number of manual ICD-10 coding histories have been stored in many hospitals. After a patient's clinical treatment, ICD-10 coders manually assign ICD-10 codes to diagnosis records [2]. Analogical reasoning is a pattern shown in Fig. 1 [13]. The matching between an uncoded diagnosis record and an ICD-10 name turns into the matching between an uncoded diagnosis record and a manually coded diagnosis record, thereby inferring the ICD-10 codes of the uncoded diagnosis record with the help of manual ICD-10 coding histories. The issue of the significant discrepancy between diagnosis records and ICD-10 names can be easily solved using analogical reasoning.

However, the automatic ICD-10 coding using simple analogical reasoning may have a cold start problem, there may be no similar record in the manually coded diagnosis records. Therefore, in order to achieve a comprehensive automatic ICD-10 coding, the flow of automatic ICD-10 coding was designed and shown in Fig. 2. First, analogical reasoning was used to find similar manually coded diagnosis records. If there is no similar record, the traditional method was used to find a similar ICD-10 standard term.

In this paper, the idea of analogical reasoning and deep semantic matching was employed in the automatic ICD-10 coding of Chinese diagnoses, as shown in the following.

## 2. Materials and methods

Text matching is a hot research direction in the field of natural language processing. To overview the development of text matching, the following four methods can be used as the baseline.

DSSM [14]: The latent semantic model with a deep structure was developed to express the semantic word vectors, and the distance between the two semantic vectors is calculated by cosine similarity.

ConvNet [15]: The convolutional neural network architecture for reranking pairs of short texts was presented that learned the optimal representation of text pairs and a similarity function to relate them in a supervised way from the available training data.

ESIM [16]: The intra-sentence attention mechanism was utilized to realize local inference, and further incorporating syntactic parsing information contributes to realizing global inference.

ABCNN [17]: Three attention schemes were proposed to integrate mutual influence between sentences into CNNs for modeling a

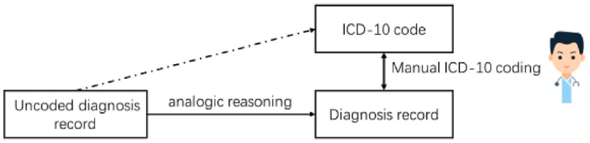

**Fig. 1.** The process of automatic ICD-10 coding with analogical reasoning.

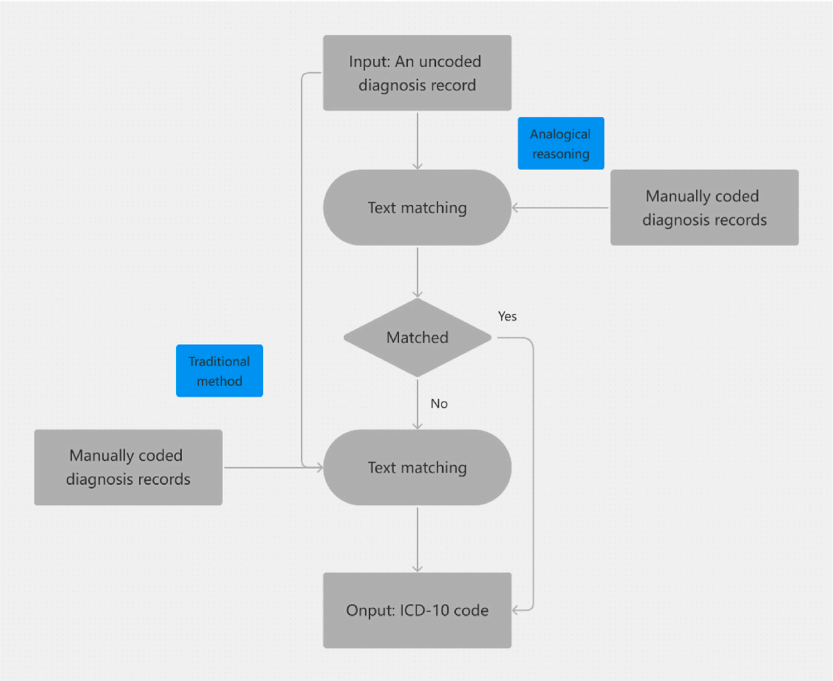
Input: An uncoded diagnosis record
Analogical reasoning
Text matching
Manually coded diagnosis records
Yes
Matched
Traditional method
No
Manually coded diagnosis records
Text matching
Onput: ICD-10 code

**Fig. 2.** The flow of automatic ICD-10 coding.

pair of sentences. The ABCNN-3 was used as the baseline.

In order to compare an uncoded diagnosis record with previously manually coded diagnosis records, we developed a deep semantic matching algorithm shown in Fig. 3. The deep semantic matching algorithm includes input, a word representation layer, a context representation layer, a matching layer, an aggregation layer, and an output layer. To train the model, the cross entropy of the training set was minimized, and Adaptive Moment Estimation was used for model optimization. The learning rate was set as 0.002.

### 2.1. Input

The model leverages two texts as inputs: an uncoded diagnosis record and a manually coded diagnosis record.

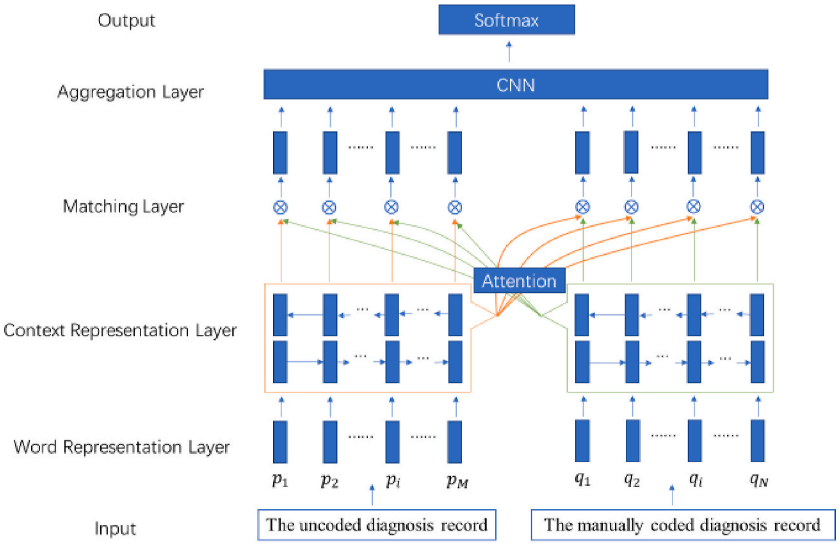
Output
Softmax
Aggregation Layer
CNN
Matching Layer
Attention
Context Representation Layer
Word Representation Layer
$p_1$ $p_2$ $p_i$ $p_M$
$q_1$ $q_2$ $q_i$ $q_N$
Input
The uncoded diagnosis record
The manually coded diagnosis record

**Fig. 3.** The proposed deep semantic matching algorithm architecture.

### 2.2. Word representation layer

There is no visible morphological marking or word boundary in Chinese text. Chinese characters serve as semantic indicators and bring certain levels of semantics [18]. Therefore, the BERT train Chinese character embeddings to represent semantics, using corpora from the open-source projects, Chinese-BERT-wwm [19]. Data includes Chinese Wikipedia, other encyclopedias, news, Q&A, and other data. The outputs of this layer are two sequences of word vectors, the uncoded diagnosis record is turned into P $= (p_1, p_2, \cdots, p_M)$ and the manually coded diagnosis record is turned into Q $= (q_1, q_2, \cdots, q_N)$. M and N denote the length of P and Q. The length of the word embeddings is 300-dimensional. For the out-of-vocabulary words, the word embeddings were randomly generated. During training, the pre-trained word embeddings were not updated.

### 2.3. Context representation layer

The context representation layer incorporates contextual information into the representation of each time step of P and Q. Every character embedding in P is given in both direct and reverse orders to a BiLSTM [20]. For each time-step of P, the concatenation of the forward and backward representations from the BiLSTM is employed as a contextual embedding. The hidden size in BiLSTM was set as 100.

$$\overrightarrow{h_i^p} = \overrightarrow{LSTM}\left(\overrightarrow{h_{i-1}^p}, p_i\right) \mathrm{i} = 1, \ldots, \mathrm{M} \tag{1}$$

$$\underset{i}{\overleftarrow{h}} p = \overleftarrow{L} STM\left(\underset{i+1}{\overleftarrow{h}} p, p_i\right) \mathrm{i} = \mathrm{M}, \ldots, 1 \tag{2}$$

Q is the same.

$$\overrightarrow{h_j^q} = \overrightarrow{LSTM}\left(\overrightarrow{h_{j-1}^q}, q_j\right) j = 1, \ldots, \mathrm{N} \tag{3}$$

$$\underset{j}{\overleftarrow{h}} q = \overleftarrow{L} STM\left(\underset{j+1}{\overleftarrow{h}} q, q_j\right) j = \mathrm{N}, \ldots, 1 \tag{4}$$

### 2.4. Matching layer

The matching layer compares each contextual character embedding of the uncoded diagnosis record against a weighted version of all contextual character embeddings of the manually coded diagnosis record. A multi-perspective matching is implemented among P and Q as shown in Fig. 4, matching each embedding of P against all embeddings of Q, and each embedding of Q against all embeddings of P. The outputs of this layer are two matching vectors, where each matching vector corresponds to the matching result of $h_i^p$ against all $h_j^q$.

The diagram of the matching calculation is shown in Fig. 3. First, the similarity or relatedness between each forward or backward contextual embedding of the uncoded diagnosis record ($\overrightarrow{h_i^p}$ or $\underset{i}{\overleftarrow{h}} p$) and every forward or backward contextual embedding of the manually coded diagnosis record ($\overrightarrow{h_j^q}$ or $\underset{j}{\overleftarrow{h}} q$, j = 1, …, N) is calculated using the cosine similarity.

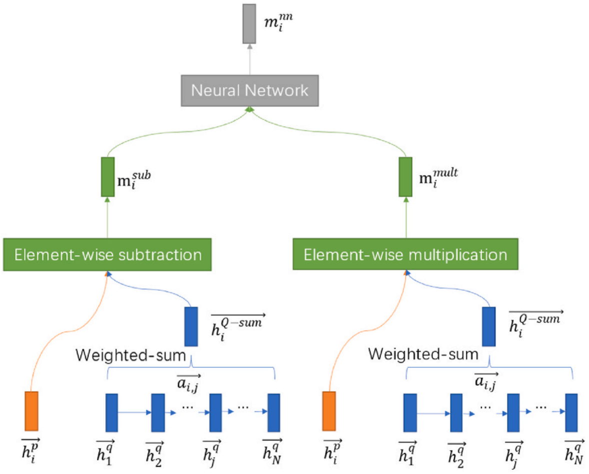

**Fig. 4.** The diagram of the matching calculation.

$$\overrightarrow{a_{i,j}} = \text{cosine}\left(\overrightarrow{h_i^p}, \overrightarrow{h_j^q}\right) j = 1, \ldots, \text{N} \tag{5}$$

$$\overleftarrow{a}_{i,j} = \text{cosine}\left(\underset{i}{\overleftarrow{h}}\, p, \underset{j}{\overleftarrow{h}}\, q\right) j = 1, \ldots, \text{N} \tag{6}$$

The attention-weighted vector is defined as $\overrightarrow{a_{i,j}}$ for the similarity between $\overrightarrow{h_i^p}$ and $\overrightarrow{h_j^q}$. The representation $\overrightarrow{h_i^{Q-sum}}$ or $\underset{i}{\overleftarrow{h}}\, Q - sum$ of the manually coded diagnosis records is formed by a weighted sum of all the contextual embeddings of Q.

$$\overrightarrow{h_i^{Q-sum}} = \frac{\sum_j^N \overrightarrow{a_{i,j}} \bullet \overrightarrow{h_j^q}}{\sum_j^N \overrightarrow{a_{i,j}}} \tag{7}$$

$$\underset{i}{\overleftarrow{h}}\, Q - sum = \frac{\sum_j^N \overleftarrow{a}_{i,j} \bullet \underset{j}{\overleftarrow{h}}\, q}{\sum_j^N \overleftarrow{a}_{i,j}} \tag{8}$$

The element-wise subtraction (SUB) and element-wise multiplication (MULT) operate on the two vectors in an element-wise manner that would not lose some useful information from the original vectors. The $\overrightarrow{m_i^{sub}}$ carries subtraction between each element in $\overrightarrow{h_i^p}$ and each element in $\overrightarrow{h_i^{Q-sum}}$, then takes the absolute value of the result. The $\overrightarrow{m_i^{mult}}$ carries multiplication between each element in $\overrightarrow{h_i^p}$ and each element in $\overrightarrow{h_i^{Q-sum}}$. The same calculation is performed in the backward direction. The concatenation of results in two directions becomes the final $\mathrm{m}_i^{sub}$ and $\mathrm{m}_i^{mult}$.

$$\overrightarrow{m_i^{sub}} = \left|\overrightarrow{h_i^p} - \overrightarrow{h_i^{Q-sum}}\right| \tag{9}$$

$$m_i^{sub} = \left|h_i^p - h_i^{Q-sum}\right| \tag{10}$$

$$\mathrm{m}_i^{sub} = \left(\overrightarrow{m_i^{sub}}, m_i^{sub}\right) \overrightarrow{m_i^{mult}} = \overrightarrow{h_i^p} \odot \overrightarrow{h_i^{Q-sum}} \tag{11}$$

$$m_i^{mult} = h_i^p \odot h_i^{Q-sum} \tag{12}$$

$$\mathrm{m}_i^{mult} = \left(\overrightarrow{m_i^{mult}}, m_i^{mult}\right) \tag{13}$$

A neural network layer that consists of a linear transformation and a non-linear activation function was carried out. The non-linear activation function used the ReLU.

$$\mathrm{m}_\mathrm{i}^\mathrm{nn}\left(h_i^p, \sum_{j=1}^{N} h_j^q\right) = \text{ReLU}\left(W \begin{bmatrix} \mathrm{m}_i^{sub} \\ \mathrm{m}_i^{mult} \end{bmatrix} + b\right) \tag{14}$$

### 2.5. *Aggregation layer*

A one-layer CNN is utilized to aggregate the matching vectors $m_i^{nn}$.

$$f = CNN\left(\left[m_1^{nn}, m_2^{nn}, \ldots, m_M^{nn}\right]\right) \tag{15}$$

### 2.6. *Output layer*

Finally, a 2-class classifier is utilized to output mapping results, which consists of multiple fully connected layers and a sigmoid layer.

### 2.7. *Evaluation metrics*

Accuracy, precision, recall, and F1-score are widely used to evaluate the performance of a binary classifier. Accuracy, precision, and recall are defined as follows:

$$\text{accuracy} = \frac{TP + TN}{TP + \text{NP} + \text{TN} + FP} \ \text{precision} = \frac{TP}{TP + FP} \ \text{recall} = \frac{TP}{TP + FN} \tag{16}$$

True Positive (TP) is an outcome where the model correctly predicts the positive class.

True Negative (TN) is an outcome where the model correctly predicts the negative class.

False Positive (FP) is an outcome where the model incorrectly predicts the positive class.
False Negative (FN) is an outcome where the model incorrectly predicts the negative class.

The F1-score is the harmonic mean of precision and recall with equal weights according to the following formula:

$$\mathrm{F1-score} = \frac{2 * \mathrm{precision} * \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{17}$$

## 3. Results

### 3.1. Datasets

From February 1, 2016, to August 30, 2021, 1,003,558 manually coded primary diagnoses on the homepage of the discharge medical records were collected as the data set, from a hospital in Hainan, China. In order to evaluate the validity of the proposed model, the raw diagnostic data needs to be de-duplicated. 162,085 raw diagnoses were obtained after the de-duplication. Each manually coded primary diagnosis consisted of a diagnostic statement and the corresponding ICD-10 name and code. Fig. 5 showed the histogram of the number of ICD-10 codes per chapter, and chapter IX (Diseases of the circulatory system) contains the most codes (175,138). There were 57,168 unencodable primary diagnoses, such as diagnostic texts including “post-surgery”, that were the status of patients after undergoing surgical procedures. The top three codes were I10 06, I25.101, and K29.502, which appeared 27,015, 22,381, and 20,059 times and represented essential hypertension grade III, coronary atherosclerotic heart disease, and Chronic gastritis.

The 143,680 manually coded primary diagnoses were used for the training set, 9214 for the development set, and 9191 for the test set. An equal number of mismatched negative samples were randomly generated. The data preprocessing included text segmentation, data deduplication, error text deletion, irregular text replacement, and symbol processing rules [21].

### 3.2. Evaluation of the proposed method

To verify the validity of analogical reasoning, the comparative experiments between the matching of Chinese diagnosis and manual ICD-10 coding history (Analogical reasoning method) and the matching of Chinese diagnosis and ICD-10 standard names (Traditional method) were carried out. Table 1 shows the comparison results, which prove the necessity of analogical reasoning in Automatic ICD-10 coding.

Considering automatic ICD coding as a text-matching problem rather than searching for the possible ICD-10 codes level by level, a constructive deep semantic matching algorithm based on analogical reasoning was designed and implemented. Many text matching algorithms were used in clinical diagnosis for the task of automatic coding. Table 2 shows a performance comparison of our proposal with previous algorithms advanced in the literature, in the average value and the 95% confidence interval.

Table 2 presents the results obtained by each model. Our model exceeded all baselines with an accuracy of 0.986, a precision of 0.979, a recall of 0.983, and an F1-score of 0.981. Among the compared text matching models, our proposed method achieves the best results in the ICD-10 automatic coding, proving that the text matching method is the most effective to reduce the coding burden manually.

The above confidence interval calculation used random sampling that can investigate the sensitivity of the model parameters on a specific dataset. But for some datasets whose distribution is unknown, it cannot show the performance of models. To estimate the

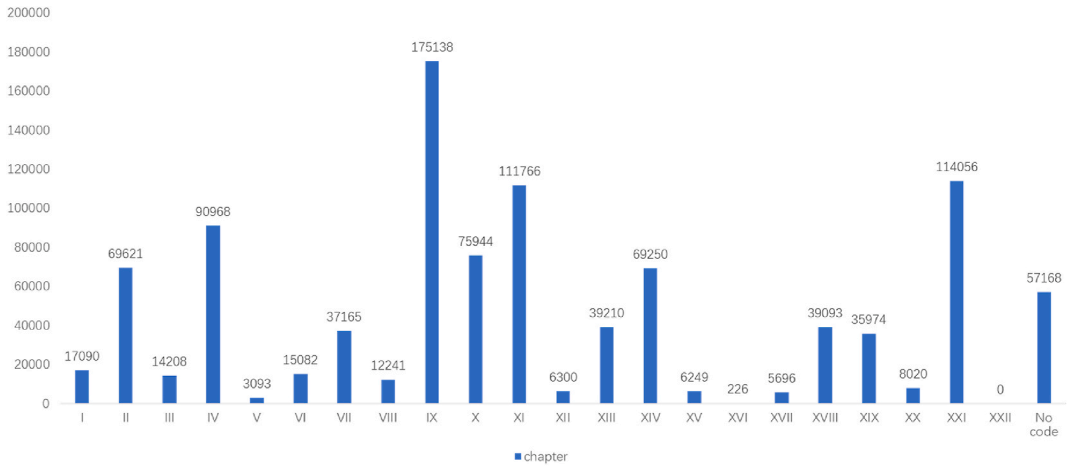

**Fig. 5.** Number of diagnosis codes per Chapter.

**Table 1**
Verification of analogical reasoning.

| Method | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Traditional method | 0.697 | 0.722 | 0.652 | 0.685 |
| Analogical reasoning method | 0.986 | 0.979 | 0.983 | 0.981 |

**Table 2**
The results of deep semantic matching models.

| Network | Accuracy (CI: 95%) | Precision (CI: 95%) | Recall (CI: 95%) | F1-score (CI: 95%) |
|---|---|---|---|---|
| DSSM | 0.719 [0.700–0.736] | 0.725 [0.711–0.737] | 0.836 [0.828–0.845] | 0.777 [0.773–0.782] |
| ConvNet | 0.954 [0.942–0.967] | 0.956 [0.945–0.970] | 0.956 [0.947–0.966] | 0.956 [0.952–0.962] |
| ESIM | 0.971 [0.964–0.984] | 0.966 [0.958–0.975] | 0.980 [0.975–0.989] | 0.973 [0.968–0.981] |
| ABCNN | 0.974 [0.963–0.984] | 0.977 [0.964–0.988] | 0.973 [0.965–0.980] | 0.975 [0.970–0.979] |
| Our model | 0.986 [0.974–0.994] | 0.979 [0.969–0.990] | 0.983 [0.976–0.992] | 0.981 [0.976–0.988] |

performance of the model on unseen data, an experiment with bootstrap confidence intervals had been also carried out and shown in Table 3.

Table 3 presents the results obtained by each model on bootstrap confidence intervals. The results of BCI and CI are very close, which fully proves that the model can effectively handle unseen data.

### *3.3. Evaluation of word representation layer*

There are many word representation models that can represent words as feature vectors to reveal semantic dependencies. For example, the 'lung' and 'pulmonary' have similar word vectors but irrelevant glyphs. Table 4 compares a few traditional word representation models. The word representation layer using BERT outperformed all baselines. This indicates that, by pretraining deep bidirectional representations from the unlabeled text by jointly conditioning on both left and right context, our model can better capture the semantic relations of words, and learn more accurate word representations [22].

### *3.4. Evaluation of context representation layer*

Many models can capture each time step of text, such as RNN, GRU, LSTM, and BiLSTM [28–30]. Table 5 displays a comparison between these models (None is Not using the text representation layer). The result of None is the worst, indicating that the text representation layer is necessary for the model. The GRU and LSTM were superior to RNN with the additional "gates" for the purpose of memorizing longer sequences of input data. Our model using BiLSTM outperformed GRU and LSTM, enabling additional training by traversing the input data twice, left-to-right and right-to-left.

### *3.5. Evaluation of matching layer*

Six ablation matching strategies were compared in order to demonstrate the effectiveness of the proposed matching strategy, and the results are reported in Table 6. Our model outperformed all ablation matching strategies, and all the calculations were really necessary for acquiring better performance. The SUB and MULT methods outperformed the Cosine similarity and Euclidean methods. The SUB method is similar to the idea of Euclidean distance, that is the square root of the sum of squared differences between corresponding elements of the two vectors. The SUB method preserves original information about the different dimensions of the two vectors instead of summing them up. The MULT method is similar to the idea of cosine similarity. When combining SUB and MULT followed by an NN layer, the performance improves significantly. Therefore, our matching strategy is really effective for automatic ICD-10 coding, and eliminating any method would hurt the performance.

**Table 3**
The results of deep semantic matching models on bootstrap confidence intervals.

| Network | Accuracy (BCI: 95%) | Precision (BCI: 95%) | Recall (BCI: 95%) | F1-score (BCI: 95%) |
|---|---|---|---|---|
| DSSM | 0.721 [0.708–0.738] | 0.727 [0.716–0.741] | 0.833 [0.824–0.843] | 0.776 [0.771–0.784] |
| ConvNet | 0.949 [0.940–0.961] | 0.954 [0.948–0.965] | 0.953 [0.944–0.961] | 0.953 [0.948–0.959] |
| ESIM | 0.978 [0.960–0.989] | 0.963 [0.951–0.974] | 0.976 [0.962–0.985] | 0.971 [0.965–0.980] |
| ABCNN | 0.975 [0.962–0.984] | 0.976 [0.963–0.986] | 0.972 [0.964–0.983] | 0.974 [0.968–0.981] |
| Our model | 0.982 [0.973–0.993] | 0.981 [0.967–0.992] | 0.985 [0.977–0.994] | 0.983 [0.977–0.990] |

**Table 4**
The results of semantic matching with different word representation models.

| Word representation models | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| FastText [23] | 0.728 | 0.731 | 0.698 | 0.714 |
| Glove [24] | 0.749 | 0.745 | 0.701 | 0.722 |
| word2Vec [25] | 0.866 | 0.872 | 0.862 | 0.867 |
| word2gm [26] | 0.890 | 0.903 | 0.854 | 0.878 |
| prob-fasttext [27] | 0.865 | 0.924 | 0.929 | 0.926 |
| BERT [22] | 0.986 | 0.979 | 0.983 | 0.981 |

**Table 5**
The results of semantic matching with different context representation layers.

| Context representation models | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| None | 0.533 | 0.578 | 0.413 | 0.481 |
| RNN [30] | 0.557 | 0.580 | 0.413 | 0.482 |
| GRU [29] | 0.702 | 0.764 | 0.573 | 0.655 |
| LSTM [31] | 0.723 | 0.740 | 0.626 | 0.678 |
| BiLSTM [28] | 0.986 | 0.979 | 0.983 | 0.981 |

**Table 6**
The results of semantic matching with different matching strategies.

| Matching strategies | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Cosine similarity [32] | 0.863 | 0.817 | 0.827 | 0.822 |
| Euclidean | 0.820 | 0.832 | 0.826 | 0.829 |
| COS + EUC | 0.844 | 0.864 | 0.829 | 0.846 |
| Neural Network | 0.957 | 0.966 | 0.974 | 0.970 |
| Subtraction | 0.964 | 0.956 | 0.952 | 0.954 |
| Multiplication | 0.931 | 0.965 | 0.964 | 0.964 |
| SUB + MULT | 0.981 | 0.973 | 0.979 | 0.976 |
| SUB + MULT + NN | 0.986 | 0.979 | 0.983 | 0.981 |

### 3.6. *Evaluation of aggregation layer*

In general, the summation and concatenation are the most commonly used aggregation layers but may lose some information. The BiLSTM and CNN are also effective for aggregation. Table 7 compares three different aggregation layers. The CNN merges the feature using the convolution kernel with the same weight to preserve more original information, so it outperformed all other aggregation layers.

### 3.7. *Evaluation of model efficiency*

One automatic ICD-10 coding calculation requires multiple text matching calculations, so it is necessary to evaluate the efficiency of the model. The comparative experiments of the time spend on performing 1000 text matching with different text matching models under the same hardware conditions, and the results are shown in Table 8.

## 4. Discussion

The fast and effective automatic ICD-10 coding of Chinese diagnosis is important because manually assigning codes is expensive and time-consuming [33]. With the specific goal of improving coding quality and efficiency, this study developed an automatic ICD-10 coding methodology of Chinese diagnosis using deep semantic matching based on analogical reasoning. This methodology shows success in a single text matching with an accuracy of 0.986, a precision of 0.979, a recall of 0.983, and an F1-score of 0.981. Compared with existing classical text matching models, the performance of DSSM received noticeably lower results than other models. The other models are interaction-based, while the DSSM is a representation-based text matching model. The interaction-based text matching models fully compare two texts that can achieve better performance but have lower operating speeds. While the representation-based can calculate and cache the word vectors in advance, so it has higher efficiency. Because the interaction between texts is shallow in the representation-based algorithms, it is usually less effective than the interaction-based.

Although many studies have focused on automatic ICD-10 coding, we want to highlight the following advantages. First, the validity of the compare-aggregate framework was verified in the task of automatic ICD-10 coding of Chinese diagnoses. The compare-aggregate framework compares vector representations of smaller units such as words and then aggregates these comparison results to make the final decision. Although some studies have shown the effectiveness of the compare-aggregate framework for text matching, automatic

**Table 7**
The results of semantic matching with different aggregation layers.

| Aggregation strategies | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Summation | 0.951 | 0.947 | 0.951 | 0.949 |
| BiLSTM | 0.976 | 0.978 | 0.981 | 0.979 |
| CNN | 0.986 | 0.979 | 0.983 | 0.981 |

**Table 8**
The results of time consumption with different text matching models.

| Model | Time |
|---|---|
| DSSM | 2571.72 s |
| ConvNet | 3218.84 s |
| ESIM | 4755.79 s |
| ABCNN | 5156.54s |
| Our model | 3318.50 s |

coding is not equal to the exact text matching task [34]. In our study, automatic ICD-10 coding using the compare-aggregate framework was proved to get excellent results. Second, the proposed method developed an improved compare-aggregate framework. Although this model follows more or less the past compare-aggregate framework, some notable differences were made. The context representation matches Chinese diagnosis P and manual ICD-10 coding histories Q in two directions (P→Q and P←Q) [35]. When comparing how semantically similar the two texts were, the matching strategies (SUB + MULT + NN) were better than the usual standard feedforward network [31]. The CNN layer was applied for aggregation instead of normal summation and concatenation. Third, the proposed method applies analogical reasoning in automatic ICD-10 coding. ICD-10 is a classification standard, so the match between current Chinese diagnosis and manual ICD-10 coding histories outperformed the match between current Chinese diagnosis and ICD-10 standard names. In summary, the proposed method has a certain performance improvement over other text matching models and outperforms other interaction-based text matching models in terms of running speed, effectively balancing performance and efficiency, and laying a good foundation for the practical application of the model.

There are also shortcomings in our study. First, the proposed method using the compare-aggregate framework was slower than traditional text matching. The deep semantic matching method can meet the reliability of automatic ICD-10 coding but lacks timeliness. Some methods for improving efficiency will be used in follow-up research, such as using the simple method to filter the top-k possible matches and reduce the number of complicated text matches [36]. Second, the proposed method cannot address the situation that draws on more patient information other than the diagnostic text in the process of ICD-10 coding. ICD-10 coding is a very complex process, and coders usually refer to all patients' data, such as discharge summaries, examinations, radiology reports, etc. The follow-up research will introduce more clinical information.

## 5. Conclusion

The proposed deep semantic matching approach based on analogical reasoning is well-suited for automatic ICD-10 coding of Chinese diagnosis. The experimental results prove the feasibility of the deep text matching method for automatic ICD-10 coding. Automatic ICD-10 coding cannot replace manual coding, but the proposed method can effectively recommend the most probable coding results to reduce the coding burden and improve coding efficiency.

## Author contribution statement

Yani Chen: conceived and designed the experiments; performed the experiments; analyzed and interpreted the data; wrote the paper.

Han Chen: conceived and designed the experiments; contributed reagents, materials, analysis tools or data; wrote the paper.

Jiye An: analyzed and interpreted the data.

Xudong Lu: analyzed and interpreted the data; wrote the paper.

Huilong Duan: contributed reagents, materials, analysis tools or data; wrote the paper.

Silin He: contributed reagents, materials, analysis tools or data; wrote the paper.

## Funding statement

This work was supported by the National Natural Science Foundation of China [grant numbers 82172070]; the research fund of academician Yu Mengsun's workstation of Hainan province.

## Data availability statement

The authors do not have permission to share data.

## Declaration of competing interest

The authors declare no competing interests.

## Acknowledgments

Yani Chen and Han Chen contributed equally to this work. We appreciate the clinical experts and clinicians participating in the evaluation experiment.

## Abbreviations

| | |
|---|---|
| ICD-10 | The International Statistical Classification of Disease and Related Health Problems, 10th Revision |
| LSTM | Long Short-Term Memory |
| BiLSTM | Bidirectional Long Short-Term Memory |
| HA-GRU | Hierarchical Attention-bidirectional Gated Recurrent Unit |
| NHFPC | National Health and Family Planning Commission |
| LCS | Longest Common Subsequence |
| BERT | bidirectional Encoder Representations from Transformers |
| SUB | Subtraction |
| MULT | Multiplication |
| NN | Neural Network |
| CNN/ConvNet | Convolutional Neural Network |
| DSSM | Deep Structured Semantic Model |
| ESIM | Enhanced Long Short-term Memory |
| ABCNN | Attention-Based Convolutional Neural Network |
| GRU | Gated Recurrent Unit |
| RNN | Recurrent Neural Network |
| CI | confidence interval |
| BCI | bootstrap confidence intervals |

## References

[1] W.H. Organization, International Statistical Classification of Diseases and Related Health Problems: Alphabetical Index, World Health Organization, 2004.

[2] Y. Chen, H. Lu, L. Li, Automatic ICD-10 coding algorithm using an improved longest common subsequence based on semantic similarity, PLoS One 12 (2017) 1–17.

[3] M. Zeng, M. Li, Z. Fei, Y. Yu, Y. Pan, J. Wang, Automatic ICD-9 coding via deep transfer learning, Neurocomputing 324 (2019) 43–50.

[4] P. Chen, S. Wang, W. Liao, L. Kuo, K. Chen, Y. Lin, C. Yang, C. Chiu, S. Chang, F. Lai, Automatic ICD-10 coding and training system: deep neural network based on supervised learning, JMIR Med. Inf. 9 (2021) e23230–e23230.

[5] S. Boytcheva, Automatic matching of ICD-10 codes to diagnoses in discharge letters, in: Proceedings of the Second Workshop on Biomedical Natural Language Processing, 2011, pp. 11–18.

[6] B. Koopman, G. Zuccon, A. Nguyen, A. Bergheim, N. Grayson, Automatic ICD-10 classification of cancers from free-text death certificates, Int. J. Med. Inf. 84 (2015) 956–965.

[7] B. Shickel, P.J. Tighe, A. Bihorac, P. Rashidi, Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis, IEEE J. Biomed. Health Inf. 22 (2017) 1589–1604.

[8] P. Xie, E. Xing, A neural architecture for automated ICD coding, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 1066–1076.

[9] T. Baumel, J. Nassour-Kassis, R. Cohen, M. Elhadad, N. Elhadad, Multi-label classification of patient notes: case study on ICD code assignment, in: Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[10] F. Duarte, B. Martins, C.S. Pinto, M.J. Silva, Deep neural models for ICD-10 coding of death certificates and autopsy reports in free-text, J. Biomed. Inf. 80 (2018) 64–77.

[11] Q. Liu, H. Zhang, H. Yu, X.-Q. Cheng, Chinese lexical analysis using cascaded hidden markov model, J. Comput. Res. Dev. 41 (2004) 1421–1429.

[12] Y. Yu, M. Li, L. Liu, Z. Fei, F. Wu, J. Wang, Automatic ICD code assignment of Chinese clinical notes based on multilayer attention BiRNN, J. Biomed. Inf. 91 (2019), 103114-103114.

[13] D. Gentner, F. Maravilla, Analogical Reasoning, the Routledge International Handbook of Thinking and Reasoning, Routledge, 2017, pp. 186–203.

[14] P.-S. Huang, X. He, J. Gao, L. Deng, A. Acero, L. Heck, Learning deep structured semantic models for web search using clickthrough data, in: Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, 2013, pp. 2333–2338.

[15] A. Severyn, A. Moschitti, Learning to rank short text pairs with convolutional deep neural networks, in: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2015, pp. 373–382.

[16] Q. Chen, X. Zhu, Z.-H. Ling, S. Wei, H. Jiang, D. Inkpen, Enhanced LSTM for natural language inference, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1657–1668.

[17] W. Yin, H. Schütze, B. Xiang, B. Zhou, Abcnn: attention-based convolutional neural network for modeling sentence pairs, Trans. Assoc. Comput. Linguist. 4 (2016) 259–272.
[18] X. Chen, L. Xu, Z. Liu, M. Sun, H. Luan, Joint learning of character and word embeddings, in: Twenty-fourth International Joint Conference on Artificial Intelligence, 2015.
[19] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang, Pre-training with whole word masking for Chinese bert, IEEE/ACM Trans. Audio Speech Lang. Process. 29 (2021) 3504–3514.
[20] G. Xu, Y. Meng, X. Qiu, Z. Yu, X. Wu, Sentiment analysis of comment texts based on BiLSTM, IEEE Access 7 (2019) 51522–51532.
[21] Y. Chen, Q. Tian, H. Cai, X. Lu, A Semi-automatic Data Cleaning & Coding Tool for Chinese Clinical Data Standardization, MEDINFO 2021: One World, One Health–Global Partnership for Digital Innovation, IOS Press, 2022, pp. 106–110.
[22] J.D.M.-W.C. Kenton, L.K. Toutanova, BERT: pre-training of deep bidirectional Transformers for language understanding, in: Proceedings of NAACL-HLT, 2019, pp. 4171–4186.
[23] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov, Enriching word vectors with subword information, Trans. Assoc. Comput. Linguist. 5 (2017) 135–146.
[24] J. Pennington, R. Socher, C.D. Manning, Glove: global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[25] T. Mikolov, É. Grave, P. Bojanowski, C. Puhrsch, A. Joulin, Advances in pre-training distributed word representations, in: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), 2018.
[26] H. Wang, Y. Wang, X. Zhang, M. Lu, Y. Choe, J. Cao, English out-of-vocabulary lexical evaluation task, in: 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), IEEE, 2019, pp. 1468–1472.
[27] B. Athiwaratkun, A.G. Wilson, A. Anandkumar, Probabilistic fasttext for multi-sense word embeddings, in: 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Association for Computational Linguistics (ACL), 2018, pp. 1–11.
[28] S. Siami-Namini, N. Tavakoli, A.S. Namin, The performance of LSTM and BiLSTM in forecasting time series, in: 2019 IEEE International Conference on Big Data (Big Data), IEEE, 2019, pp. 3285–3292.
[29] R. Dey, F.M. Salem, Gate-variants of gated recurrent unit (GRU) neural networks, in: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), IEEE, 2017, pp. 1597–1600.
[30] L.R. Medsker, L. Jain, Recurrent neural networks, Des. Appl. 5 (2001) 64–67.
[31] S. Wang, J. Jiang, Learning natural language inference with LSTM, in: Proceedings of NAACL-HLT, 2016, pp. 1442–1451.
[32] F. Rahutomo, T. Kitasuka, M. Aritsugi, Semantic cosine similarity, in: The 7th International Student Conference on Advanced Science and Technology ICAST, 2012, p. 1.
[33] L. Zhou, C. Cheng, D. Ou, H. Huang, Construction of a semi-automatic ICD-10 coding system, BMC Med. Inf. Decis. Making 20 (2020) 1–12.
[34] H. He, J. Lin, Pairwise word interaction modeling with deep neural networks for semantic similarity measurement, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 937–948.
[35] Z. Wang, W. Hamza, R. Florian, Bilateral multi-perspective matching for natural language sentences, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 4144–4150.
[36] P. Cremonesi, Y. Koren, R. Turrin, Performance of recommender algorithms on top-n recommendation tasks, in: Proceedings of the Fourth ACM Conference on Recommender Systems, 2010, pp. 39–46.