BMC Medical Informatics and Decision Making

RESEARCH

Open Access

# An explainable CNN approach for medical codes prediction from clinical text

Shuyuan Hu[1], Fei Teng[1*], Lufei Huang[1,2], Jun Yan[3] and Haibo Zhang[4]

*From* The China Conference on Health Information Processing (CHIP) 2020 Shenzhen, Guangdong, China. 30-31 November 2020

## Abstract

**Background:** Clinical notes are unstructured text documents generated by clinicians during patient encounters, generally are annotated with International Classification of Diseases (ICD) codes, which give formatted information about the diagnosis and treatment. ICD code has shown its potentials in many fields, but manual coding is labor-intensive and error-prone, lead to researches of automatic coding. Two specific challenges of this task are (1) given an annotated clinical notes, the reasons behind specific diagnoses and treatments are implicit; (2) explainability is important for practical automatic coding method, the method should not only explain its prediction output but also have explainable internal mechanics. This study aims to develop an explainable CNN approach to address these two challenges.

**Method:** Our key idea is that for the automatic ICD coding task, the presence of informative snippets in the clinical text that correlated with each code plays an important role in the prediction of codes, and an informative snippet can be considered as a local and low-level feature. We infer that there exists a correspondence between a convolution filter and a local and low-level feature. Base on the inference, we come up with the Shallow and Wide Attention convolutional Mechanism (SWAM) to improve the CNN-based models' ability to learn local and low-level features for each label.

**Results:** We evaluate our approach on MIMIC-III, an open-access dataset of ICU medical records. Our approach substantially outperforms previous results on top-50 medical code prediction on MIMIC-III dataset, the precision of the worst-performing 10% labels in previous works is increased from 0% to 53% on average. We attribute this improvement to SWAM, by which the wide architecture with attention mechanism gives the model ability to more extensively learn the unique features of different codes, and we prove it by an ablation experiment. Besides, we perform manual analysis of the performance imbalance between different codes, and preliminary conclude the characteristics that determine the difficulty of learning specific codes.

**Conclusions:** Our main contributions can be summarized into the following three: (1) We present local and low-level features, a.k.a. informative snippets play an important role in the automatic ICD coding task, and the informative snippets extracted from the clinical text provide explanations for each code. (2) We propose that there exists a correspondence between a convolution filter and a local and low-level feature. A combination of wide and shallow

*Correspondence: fteng@swjtu.edu.cn
[1] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China
Full list of author information is available at the end of the article

convolutional layer and attention layer can help the CNN-based models better learn local and low-level features. (3) We improved the precision of the worst-performing 10% labels from 0 to 53% on average.

**Keywords:** ICD coding, Machine learning, Attention mechanism, Convolutional neural network

## Background

Clinical notes are written by clinicians during patient encounters, they are usually unstructured text narratives and accompanied by a set of metadata codes from the International Classification of Diseases (ICD), which present a standardized way of indicating diagnoses and procedures that were performed during the encounter. There is much research that demonstrates the practical application with ICD codes [1–3]. For example in work by Choi et al. [4], they proposed the Doctor AI system based on the presence of ICD codes to predict future patient states from learning patient representation from a large dataset of patient records.

Manual coding is time-consuming and error-prone, so much research on automatic coding has been done in the pastdecades, some recent works are Zhang et al. [5]; Kavuluru et al. [6] and Avati et al. [3]. Automatic coding is considered a multi-label classification task, and two domain-specific challenges are facing this task. First, a reasonable guess is that for a certain code prediction task, most of the text is not informative, only a few snippets are related to the code. However given the annotated text, the connections between code and its corresponding informative snippets are lost, in other words, the model has to learn the reasons behind specific diagnoses and treatments.Second, interpretability is a crucial obstacle for practical automatic coding in both perspective of inferring and internal mechanics, the method is supposed to explain its prediction as well as have an explainable internal mechanics.

To address these two specific challenges together, in this paper, we develop CNN-based methods for automatic ICD coding assignment based on text discharge summaries from ICU stays, we come up with Shallow and Wide Attention convolutional Mechanism (SWAM), which allows our model to learn local and low-level features for each label. Our model design is motivated by the way human clinicians manual label the clinical notes, which is to look for informative snippets that are relevant to each code. We consider the informative snippets as local and low-level features. SWAM address the two challenges in automatic coding: first, by transferring the base representation (i.e. clinical notes in the word-embedding form) to the convolution representation which represents the presence of informative snippets, the model could filter out the irrelevant information in the text, and through the attention mechanism the model could learn the correlation between informative snippets and labels Second, SWAM gives informative snippets extracted from clinical notes as explanations of its prediction result, and provides a new perspective for understanding the internal mechanics of the machine learning method.

We evaluate our approach on the MIMIC-III dataset [7], an open dataset of ICU medical records. With the Shallow and Wide Attention CNN mechanism, the model can learn non-generic features associated with specific labels that are not informative for other labels, which the narrow one are failed to learn. With the performance improvement gained from these specific labels, our approach outperforms previous results on medical code prediction on MIMIC-III dataset.

### Related work

#### *Automatic ICD coding*

ICD coding has been a long-established task in the medical informatics community for decades, from the perspective of data, the current approaches of this task can be divided into two factions: much recent research focuses on unstructured text data [6, 8], while the other incorporates structured data as well [9]. We develop our methods on unstructured text data from the MIMIC-III. From the perspective of the code set, many approaches [10, 11] evaluate on a subset of the full ICD label space, while there are also methods [12] developed on the full code set. We develop our methods on the top-50 code set because the advantage of SWAM is learning specific features associated with specific labels that are not generic feature for other labels, so instead of carrying out a surprisingly large network to learn all non-generic features on the full code set, using ensemble method to cover the whole code set is preferred, which is discussed in later part.

A tendency in recent years is developing Neural Network-based methods for this task. Shi et al. [13] applied attentional LSTMs to form a soft matching between sentence representations from discharge summaries and the top 50 codes. Prakash et al. [11] generated predictions of the top 50 codes by memory networks built from discharge summaries and Wikipedia. Mullenbach et al. [12] applied a per-label mechanism to extract the most important snippet for each code from discharge summaries. SWAM is compared with the published result from all these papers, and it achieves state-of-the-art results across many indicators. We attribute these

improvements to the ability to learn non-generic features associated with specific labels that are not informative for other labels, which bring significant performance improvements on these specific labels.

#### *Attentional convolution for NLP and explainable text classification*

Combing convolution with attention has been proved is efficient in different tasks among NLP [14–18]. Yang et al. [19] and Mullenbach et al. [12] utilize attentional convolution to select the most relevant parts of the clinical text of each code. We refer to the per-label attention mechanism from those of Mullenbach et al. [12], in which per-label parameter vectors are used to compute attention over specific locations in the text. Our work differs in that SWAM establishes the correspondences between the "informative snippet" and convolution filter, which makes the network a wider one comparing to its of Mullenbach et al. [12] and is better tuned to our goal of learning low-level feature, a.k.a. informative snippet with explainable internal mechanics.

Attentional Convolution has also been applied to make explainable text classification. Some prior works like Rushet al. [20] and Rocktäschel et al. [21] employ attention to highlight salient features of the text. The per-label attention mechanism [12] we referred extract snippet from the text as automatically generated explanation of the prediction in the same medical codes prediction task, and the informativeness of explanations are rated by a physician. Their work illustrates that the neural network work in an explainable way for this task: the model will try to find parts of the text that are most relevant to each code. Our work differs in that instead of making the model explainable by explaining its prediction, we take a further step forward to make the internal mechanics of the method explainable by opening the black box of the neural network to establish the correspondences between the "informative snippet" and convolution filter. We also bring out a preliminary analysis of the imbalance performance between the labels, provide a rational explanation of why the model performs terribly on certain codes.

#### *Neural network architecture design for text classification work*

Hoa et al. [22] compared the deep CNN and shallow CNN under text classification task, a practical rule is summarized that deep models do not seem to bring a significant advantage over shallow networks for text classification, another observation they made is that a global max-pooling [23], which retrieves the most influential feature could already be good enough for the text classification task. The authors believe one possible reason may be related to these facts that images are represented as real and dense values, as opposed to the discrete, artificial, and sparse representation of text. Their work indicates that local and low-level features extracted by shallow CNN work well for text classification tasks and inspires us to explore the underlying correspondences between local and low-level features and snippets in the text.

Gong and Ji [24] find that in CNN for the text classification task, the convolution filters have learned division of labor. More than half of the kernels have a preference for one specific label. Their work inspires us to associate the width of the network with the learning of features of specific labels that are not generic for other labels.

## Methods

In this paper, we use notations shown in Table 1.

We present SWAM, a CNN-based method for automatic ICD coding from the clinical text, which provides a good explanation of its internal mechanics.

SWAM is motivated by the way human clinicians manual label the clinical notes, to help the reader understand the method, firstly here is a brief introduction of the way human clinicians manual label the clinical notes. Normally, human clinicians will look for informative snippets that are relevant to each code. For example, as shown in Fig. 1, given code 96.04 in the figure, a human clinician will look for the presence of relevant snippets in the clinical notes. In this case, the relevant snippets are "intubation" and "endotracheal intubation", if the human clinician finds the relevant snippets, he/she will give a positive prediction of code 96.04.

SWAM refers to the same idea of manual coding. As shown in Fig. 1, the first step, through the convolutional layer the model will extract informative snippets that could be relevant to any code. In the second step, the attention layer will assign importance weight to snippets to select the relevant snippets of each code, and in the final step the model summary the weighted score of all relevant snippets of each code to give the predictions of the presence of each code.

### Correspondences between informative snippet and convolution filter

Our explanation of the internal mechanics of SWAM builds on correspondences between "informative snippet" and convolution filter. Firstly, we classify the "informative snippet" into two categories: "generic snippet" and "non-generic snippet". "generic snippet" refers to snippets that are informative for multiple labels, for example, in our task, "experience fever" is likely to be a "generic snippet" since it is the symptom correlated with multiple diagnoses. "non-generic snippet" refers to snippets that are only considered as informative to a

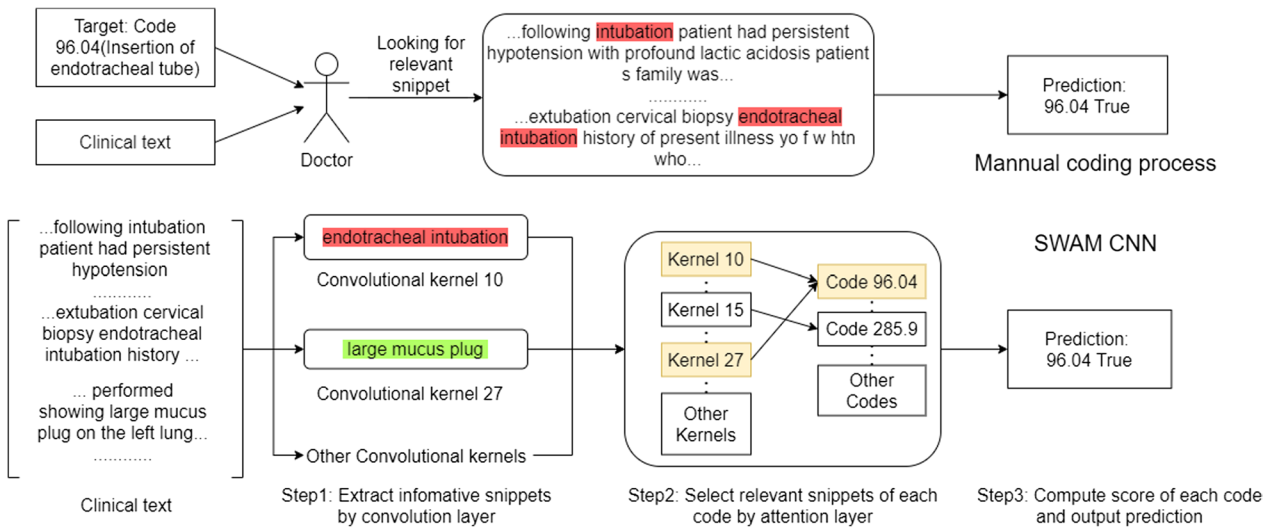


**Fig. 1** Internal mechanism of SWAM

**Table 1** Table of Notations

| Notation | Description |
|---|---|
| $\mathcal{L}$ | The set of ICD-9 codes |
| $y_{i,\ell} \in 0, 1$ | The true value of the label task for instance i and $\ell \in \mathcal{L}$, 1 indicates the label is true for instance i |
| $d_e$ | The size of the input embedding |
| $d_c$ | The size of the convolution output, a.k.a. the number of convolution filters |
| $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ | The matrix of a document instance, where $\mathbf{N}$ is the length of the document and $\mathbf{x}_i$ is the vector representation of the word |
| $\mathbf{W}_c \in \mathbb{R}^{k \times d_e \times d_c}$ | Convolution filters, where k is the width of filter window |
| $\mathbf{H} \in \mathbb{R}^{d_c \times N}$ | Convolutional representation of the document |
| $*$ | Convolution operator |
| $g$ | An element-wise nonlinear transformation |
| $\mathbf{b}_c \in \mathbb{R}^{d_c}$ | The bias in convolutional operation |
| $\mathbf{u}_\ell \in \mathbb{R}^{d_c}$ | Attention parameter vector for label $\ell$ |
| $\boldsymbol{\alpha}_\ell \in \mathbb{R}^N$ | Attention result vector for label $\ell$ |
| $b_\ell$ | Scalar offset in linear layer for label $\ell$ |
| $\boldsymbol{\beta}_\ell \in \mathbb{R}^{d_c}$ | Vector of prediction weights |
| $\sigma$ | Sigmoid function |
| SoftMax () | $\text{SoftMax}(\mathbf{x}) = \frac{\exp(\mathbf{x})}{\sum_j \exp(x_j)}$, where exp $(\mathbf{x})$ is the element-wise exponentiation of the vector $\mathbf{x}$ |

specific or a few labels, for example, in top-50 code task, "endotracheal intubation" will be considered as a "non-generic snippet" since it brings little information gain to the other 49 labels than it brings to the code 96.04 "Insertion of endotracheal tube".

Then we infer that there exists a correspondence between "informative snippet" and convolution filter, which means one convolution filter can only generate a high activation value for a specific "informative snippet". Given that in the CNN context, the "informative snippet" can be considered as a set of word embedding sequences that are close in the embedding space. For example, "large mucus plug" and "big mucus plug" are the same "informative snippet" since they have similar meanings and therefore are close in the embedding space. It is most likely that for different "informative snippets", they will have very little chance to be close in the embedding space. For each filter, it

will be "highly activated" ouput exceeds threshold when the snippet in its window is close to its parameters in embedding space, and this snippet can be considered as the "informative snippet" corresponds to this filter.

Based on the above inference, an obvious conclusion is that the choice of the width of the convolution layers, a.k.a. the number of convolution filters should depend on the total numbers of "informative snippets" in the task, more accurately, the number of "non-generic snippet" since it will be much larger than the number of "generic snippet" in the large-scale coding task. Besides, empirical guidance in our architecture design is that there could be multiple "non-generic snippets" for each code [24].

Therefore we develop Shallow and Wide Attention CNN for this task: the presence of the informative snippet of each code could be considered as a local, low-level feature learned by the shallow CNN, and we also need the convolutional layer to be wide since the model needs to learn the "non-generic snippets" of all codes.

The mechanism behind Shallow and Wide Attention CNN is general for a set of similar text classification tasks that informative snippets relevant to each label scattered at random locations in the input document. So SWAM can be regarded as a general architecture with the following three characteristics, and implement details can be varied (e.g. the attention layer in the model can be either per-label attention mechanism [12] or the full connected layer in textCNN [25]).

1. The convolutional layer should be sufficiently wide, a.k.a. enough convolution filters to not only extract all generic snippets that are informative for multiple labels, but also all non-generic features that are correlated to specific label and not informative to other labels, the certain number of filters depends on task context, a.k.a. the total number of generic features and non-generic features in the task.
2. The network architecture should be shallow, this model is designed to extract snippets of text, which can be considered as local, low-level features, so a deeper network is unnecessary since informative snippets relevant to each label scattered at random locations in the input document, it is not likely that we can earn any benefit from the global, high-level features by combining the adjacent snippets.
3. Attention mechanism should be introduced to learn the correlations between important/informative snippets and each code.

#### *Word embedding*

The word embedding model used in this paper is the word2vec CBOW method by Mikolov et al. [26], we pre-train word embedding of size $d_e = 100$ on the preprocessed text from all discharge summaries in MIMIC3, which is the same dataset for training our model. Details about the dataset can be found in Dataset. We treat ICD code prediction as a multilabel text classification problem [27]. For clinical note instance i, we want to determine $y_{i,\ell} \in 0, 1$ for all $\ell \in \mathcal{L}$. We train a neural network which passes text through a convolutional layer to compute a base representation of the text of each document [25], and makes $|\mathcal{L}|$ binary classification decisions.

#### *Convolutional Layer*

The input of convolutional layer is the clinical notes in form of pre-trained embeddings representing by the matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$. The convolution of adjacent embeddings are computed with a convolution filter $W_c \in \mathbb{R}^{k \times d_e \times d_c}$. At step n, we compute

$$\mathbf{h}_n = g(\mathbf{W}_c * \mathbf{x}_{n:n+k-1} + \mathbf{b}_c) \tag{1}$$

The input is padded on both sides with zeros so the base representation $\mathbf{H}$ keeps the same length as $\mathbf{X}$.

#### *Attention layer*

Nowadays attention mechanism has been generalized and has been employed in many different forms [28]. The core idea of the attention mechanism can be regarded as "giving weight to different parts of the input, to select the part in the input that is more important for the current task". So the full connected layer in textCNN [25] can also be regarded as a kind of "attention" since it weighs input separately for each label.

As we mention in Correspondences between "informative snippet" and convolution filter, SWAM can be regarded as a general CNN architecture, and implement details can be varied. We adopt two different implements of the attention layer in our model for different consider considerations. The first one is the per-label attention mechanism by Mullenbach et al. [12], we adopt it because it can extract snippets from the clinical text as explanations of the model prediction, which can be used to verify our conjecture about the correspondence between "informative snippet" and convolution filter. The second one is the common full connected layer in textCNN [25], we adapt it since the textCNN is the basis of many works so it can prove the versatility of SWAM.

For the per-label attention mechanism (the implement shown in Fig. 2), the idea is to calculate the

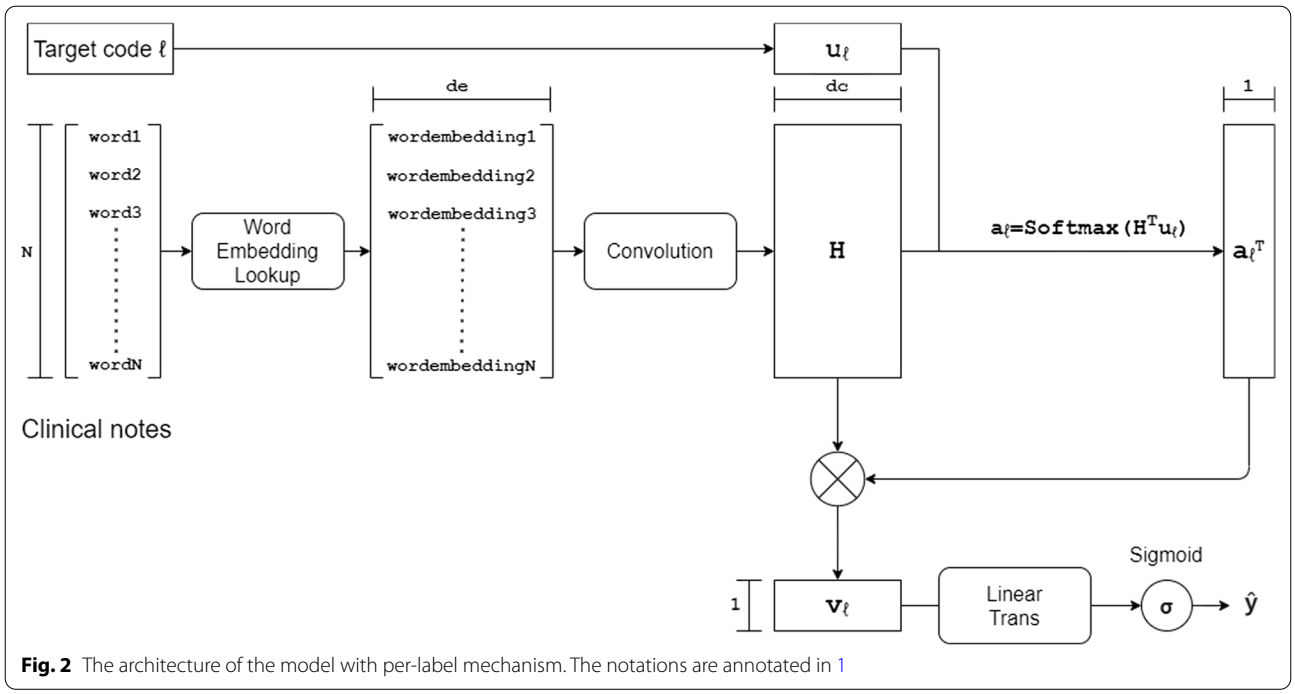


**Fig. 2** The architecture of the model with per-label mechanism. The notations are annotated in 1

per-label representation of the document and use an attention vector $\boldsymbol{\alpha}_\ell$ to represents importance distribution over locations in the document. To obtain the per-label representation of the document, formally a vector parameter $\mathbf{u}_\ell \in \mathbb{R}^{d_c}$ is used to compute the matrix-vector product, $\mathbf{H}^\top \mathbf{u}_\ell$, which can be taken as that the base representation $\mathbf{H}$ is weighted for label $\ell$. The resulting vector is then normalized using a SoftMax operation, obtaining $\boldsymbol{\alpha}_\ell$, the attention vector, the value of the element in the attention vector is the weighted sum of convolutional features from all kernels in the same place of the document.

$$\boldsymbol{\alpha}_\ell = \text{SoftMax}(\mathbf{H}^\top \mathbf{u}_\ell) \tag{2}$$

$\boldsymbol{\alpha}_\ell$ is also taken as the location indicator of the most important snippet for label $\ell$, every element in $\boldsymbol{\alpha}_\ell$ is corresponding to a location in the document, the value of the element is seen as the importance of the corresponding location for label $\ell$. The highest element value in $\boldsymbol{\alpha}_\ell$ means the snippet in this location is most important (a.k.a. most informative) for the prediction of label$\ell$. Therefore we obtain an explanation of the prediction in the form of extracted snippets from the document.

$\alpha_{\ell,n}\mathbf{h}_n$, the element-wise vector product is then computed, applies the attention vectors on the base representation to get the vector document representations $\mathbf{v}_\ell$ for label $\ell$,

$$\mathbf{v}_\ell = \sum_{n=1}^{N} \alpha_{\ell,n}\mathbf{h}_n \tag{3}$$

For full connected attention, we instead use max-pooling to filter the base representation down to a vector $\mathbf{v} \in d_c$ where every element in $\mathbf{v}$ corresponds to the highest action value of a convolution filter in the text,

$$v_j = \max_n h_{n,j} \tag{4}$$

#### *Classification*

Given the vector representation $v_\ell$, the likelihood for label $\ell$ is computed using a linear layer and a non-linear function sigmoid:

$$\hat{y}_\ell = \sigma\left(\boldsymbol{\beta}_\ell^\top \boldsymbol{v}_\ell + b_\ell\right) \tag{5}$$

#### *Loss function*

The training procedure use BCE (binary cross-entropy) as the loss function, the optimization goal is to minimize the loss.

$$L_{\text{BCE}}(X, y) = -\sum_{\ell=1}^{\mathcal{L}} y_\ell \log\left(\hat{y}_\ell\right) + \left(1 - y_\ell\right) \log\left(1 - \hat{y}_\ell\right) \tag{6}$$

## Results

### Dataset

MIMIC-III [7] is an open-access dataset comprising health data in the form of text and structured records of ICU admissions. Since MIMIC was built, it has become the basis of many works on multi-label classification [10, 29]. Following previous works, we train our model on discharge summaries in MIMIC, which summary records about one stay into a single document. We focus on the raw text of the data and ignore the attached features like admission time. Every discharge summary is corresponding to an admission, and each admission is annotated with a set of ICD-9 codes, describing both diagnoses and treatments that occurred during the patient's stay. We train and evaluate SWAM on a label set consisting of the 50 most frequent labels. We filter the dataset down to the instances that have at least one of the top 50 most frequent codes. Some patients have multiple admissions and therefore multiple discharge summaries. To prevent the model from learning unnecessary correlations, we split the data by patient ID, so that no patient appears in both the training and test sets. After the split, there are 8,067 summaries for training, 1,574 for validation, and 1,730 for testing. Other detailed statistics for the setting are summarized in Table 2.

#### *Preprocessing*

We remove the tokens that contain no alphabetic characters (e.g., removing '100' but keeping '100ml' ). For those tokens that appear too few times to make their semantics difficult to learn, a threshold that only remains tokens that appear in no fewer than 3 training documents is setting, and all tokens that failed the threshold are replaced with an 'UNK' token. The distribution of discharge summaries conforms to the long-tailed distribution, 90% of discharge summaries are short than 2500 tokens, so we truncated discharge summaries to a maximum length of 2500 tokens.

### Baselines

As mentioned in Correspondences between "informative snippet" and convolution filter, SWAM can be regarded as a general CNN architecture and implement details can be varied. In model part Attention layer two different implements of attention layer are adapt for different considerations, we name those two implements as "SWAM-textCNN" [25] and "SWAM-CAML" [12] separately to indicate the attention approaches they refers.

The baseline we compare against is a bag-of-words logistic regression model, we also compare SWAM-CAML with the origin implement of CAML [12] at the same setting.

For SWAM-textCNN and SWAM-CAML we initialize the embedding weights using the same pre-trained word-2vec vectors. The logistic regression model consists of $|\mathcal{L}|$ binary one-vs-rest classifiers acting on unigram bag-of-words features.

**Table 2** Descriptive statistics for MIMIC3

| | MIMIC-III full | MIMIC-III 50 |
|---|---|---|
| Training documents | 47,724 | 8,067 |
| Vocabulary size | 51,917 | 51,917 |
| Mean tokens per document | 1,485 | 1,530 |
| Mean labels per document | 15.9 | 5.7 |
| Total labels | 8922 | 50 |

#### *Parameter tuning*

We tune the hyper-parameters of our SWAM models using grid search. We sample parameter values for the learning rate $\eta$, as well as filter size $k$, number of filters $d_c$, and dropout probability $q$ as shown in Table 4. We also adopt hyper-parameter tuning in the previous works as empirical guidance [12, 25, 31]. We use a fixed batch size of 16, and train the model with early stopping, in the case that the f1-macro does not improve for 10 epochs the training will terminate.

### Evaluation metrics

We focus on two metrics: Macro-averaged F1 and precision at n (denoted as 'P@n'), which is the fraction of the $n$ highest-scored labels that are present in the ground truth. The reason we focus on Macro-averaged F1 is that it pays attention to per-label performance, which can reflect the average performance of the model on different label tasks. As for P@n, we choose it because it reflects the performance of the model as a practical decision support system which presents a fixed number of predicted codes to help user annotated the clinical text. To facilitate comparison with both future and prior work, we also report a variety of metrics includes the area under the ROC curve (AUC) and micro-averaged F1. For recall, Macro-averaged values are calculated by averaging metrics computed per-label. Micro-averaged values are calculated by treating each (document, code) pair as a separate prediction.

### Results on quantitative evaluation

Our main quantitative evaluation involves predicting the 50-code set of ICD-9 codes based on the text of the MIMIC-III discharge summaries. These results are shown in Table 3. We adopt two different implements of attention layer, named "SWAM-textCNN" and "SWAM-CAML" The SWAM models give the strongest results on all metrics, especially on F1-Macro, which emphasis

**Table 3** Results on MIMIC-III, 50 labels

| Model | AUC | | F1 | | P@5 |
|---|---|---|---|---|---|
| | Macro | Micro | Macro | Micro | |
| C-MemNN [11] | 0.833 | – | – | – | 0.42 |
| Shi et al. [13] | - | 0.900 | - | 0.532 | - |
| CAML [12] | 0.875 | 0.909 | 0.532 | 0.614 | 0.609 |
| Logistic regression | 0.828 | 0.862 | 0.477 | 0.530 | 0.545 |
| SWAM-CAML | **0.900*** | **0.924*** | 0.593 | 0.648 | **0.625*** |
| SWAM-textCNN | 0.892 | 0.919 | **0.603*** | **0.652*** | 0.620 |

(*) by the bold (best) result indicates significantly improved results compared to the other methods, the bootstrapping method [30] is used for the statistical significance analysis, $p < 0.01$

**Table 4** Hyper-parameter tuning ranges and optimal values for SWAM model

| | Range | Optimal value |
|---|---|---|
| $\eta$ (learning rate) | 0.0001,0.0003, 0.001,0.003 | 0.001 |
| $k$ (filter size) | 1–10 | 4 |
| $d_c$ (number of filters) | 50–500 | 500 |
| $q$ (dropout probability) | 0.2–0.8 | 0.2 |

average performance over different labels. We attribute this improvement to SWAM, by which the wide architecture gives the model ability to more extensively learn the unique features of different codes.

#### Ablation experiment on the width of the network

According to our inference about the correspondence between the informative snippet and convolution filter, since each "non-generic snippet" has to correspond to a convolution filter, if the network is too narrow, the model will fail to learn the "non-generic snippets" of some labels. Therefore the impact of the width of the network can be observed from the perspective of per-label performance.

We carry out an ablation experiment on the width of the network by comparing the per-label performance of the wide model (wide-SWAM) and the narrow model (narrow-SWAM) (Fig. 3). The only difference between wide-SWAM and narrow-SWAM are the width of the network, the former has 500 convolution filters and the latter has 50. The experiment results are in line with our expectations. For the narrow model, 5 labels have a precision of 0, while on the opposite, the wide model makes significant performance improvement that 4 of 5 labels that with a 0 precision in the narrow model now have an average precision of 0.53, at the same time the overall performance of the model is improved. As for the only ICD-9 code 285.9 "Anemia, unspecified" that has a 0 precision in both models, we make a manual analysis in section analysis of the reason behind bad performance code.

### Secondary evaluation

#### *Comparing informative snippets extracted by narrow and wide models*

To verify our inference about Correspondences between "informative snippet" and convolution filter, we also compare the informative snippets extracted by both the narrow and the wide model. In order to make the cases representative, from the five labels that has 0 precision in the narrow model, we pick up a label (276.1: Hyposmolality and/or hyponatremia) that has a improved precision in the wide model, and the only label(285.9 Anemia, unspecified) that has 0 precision in both models for analyse.

Table 5 shows the informative snippets extracted by the wide-SWAM and the narrow-SWAM model during the prediction of code 276.1 in two random selected documents. Through a simple analysis, it can be found that the word "hyponatremia" extracted by the wide-SWAM model that appears in both the document and the code description plays an important role in the prediction. While on the opposite, the snippet extracted by the narrow-SWAM model is not informative since it has 0 precision on this code.

The word "hyponatremia", as a local and low-level feature can be learned by a single convolution filter according to our inference Correspondences between "informative snippet" and convolution filter. Since the only difference between the wide-SWAM model and the narrow-SWAM model is the number of filters in the convolutional layer. Table 5 proves that the performance difference between the narrow and the wide model comes from the learning of non-generic "informative snippet".

#### *Factors determine which "non-generic snippet" will fail to be learned by narrow model*

According to our inference, a narrow network will fail to learn the "non-generic snippet" of a part of labels, which naturally raises a question: what factors determine which "non-generic snippet" will not be learned? The essence of failing to learn different "non-generic snippet" is that the model converges to different local optimal parameters. The convergence result of the model is related to the distribution of data during

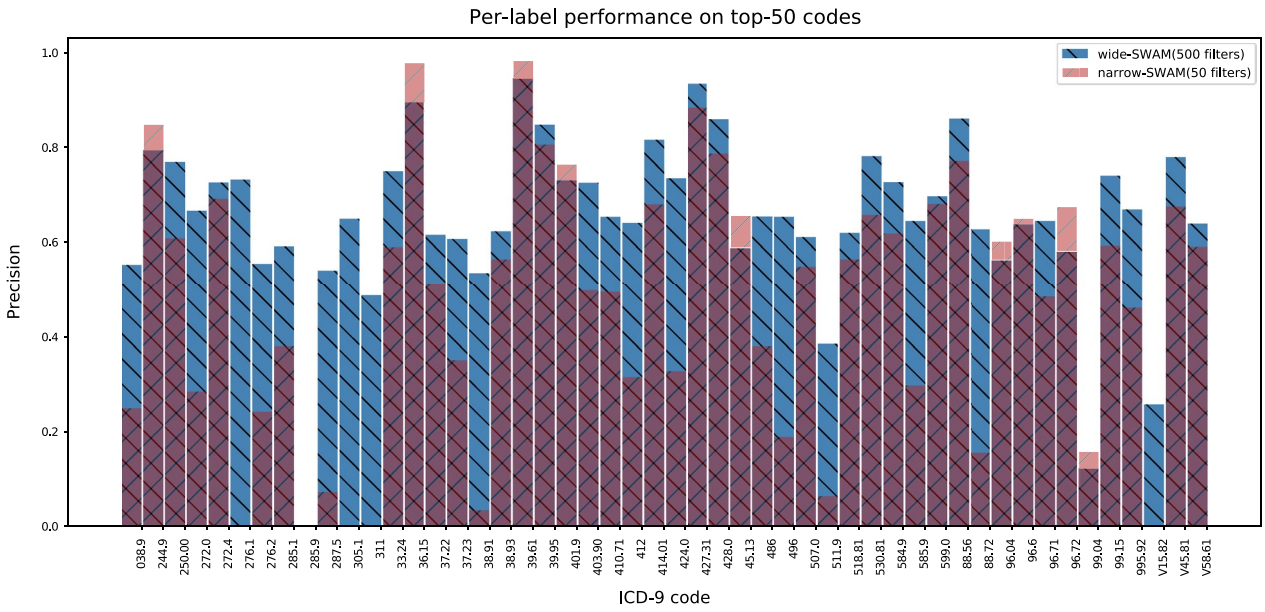


**Fig. 3** Ablation experiment on the width of the network, the only difference between two models are the width of the network. wide-SWAM has 500 filters, while narrow-SWAM has 50 filters

training, in other words, the order that "non-generic snippets" appear during training: once a filter learns a specific "non-generic snippet" for some labels, the loss function will encourage it to keep its parameters unchanged, and after all the filters have learned corresponding "informative snippet", it's difficult for the model to leave the current local optimal solution and learn new "informative snippet".Therefore we shuffle the training dataset with a different random seed and retrain the models with the shuffled dataset. The results are shown in Table 6.

The results in Table 6 are in line with our expectations, after shuffle and re-training, the 0 precision labels in the narrow model change. On the opposite, the only 0 precision label 285.9 in the wide model still can not be learned. The distribution of data during training is a factor that determines which "non-generic snippet" will fail to be learned by the narrow model. And the reason

**Table 6** ICD Code with 0 precision in both the wide model and the narrow model before and after data shuffle, bold code indicates the codes with 0 precision emerged after the shuffle

| Before shuffle | ICD Code with 0 precision |
|---|---|
| Wide-SWAM | 285.9 "Anemia, unspecified" |
| Narrow-SWAM | 285.9 "Anemia, unspecified" |
| | V15.82 "History of tobacco use" |
| | 276.1: "Hyposmolality and/or hyponatremia" |
| | 305.1 "Tobacco use disorder" |
| | 311 "Depressive disorder, not elsewhere classified" |
| After shuffle | ICD Code with 0 precision |
| Wide-SWAM | 285.9 "Anemia, unspecified" |
| Narrow-SWAM | 285.9 "Anemia, unspecified" |
| | V15.82 "History of tobacco use" |
| | **272.0 "Pure hypercholesterolemia"** |
| | **V45.81 "Postsurgical aortocoronary bypass status"** |

**Table 5** Informative snippets extracted by the wide model and the narrow model for prediction of ICD code 276.1, bold snippet indicates the snippets evaluated as informative

| **ICD code 276.1: "Hyposmolality and/or hyponatremia"** | |
|---|---|
| Wide-SWAM | ...**Dehydration** and increased abd... |
| Wide-SWAM | ...**Hyponatremia** and possible initiation of chemotherapy... |
| Narrow-SWAM | ...Peritonitis renal failure and ileus on the floor the patient was followed by... |
| Narrow-SWAM | ...Renal failure and small bowel obstruction of note the provided information on... |

behind the bad performance ICD code 285.9 "Anemia, unspecified" is something beyond the local optimum.

#### *Analysis of the reason behind bad performance code*

The ICD-9 code 285.9 "Anemia, unspecified" is failed to be predicted by both narrow and wide models. Through manual analyzing, we found there are more than 50 codes in the ICD-9 that are in form of "Anemia + specific reason", which means the snippets related to anemia cannot are necessary but not sufficient for prediction of code 285.9. The prediction of 285.9 Anemia, unspecified is not only based on the presence of snippet related to 'Anemia', it is also based on the information that all possible reasons are absent. This is a blind spot in all current machine learning models. It is difficult for models to learn inferences based on missing information.

### Why shallow and wide attention CNN

Through the above experiments, we show that compared to other methods, SWAM significantly improves the precision of the worst-performing 10% labels meanwhile achieves better overall performance on the automatic coding task. We proved that this improvement is closely related to SWAM's ability to learn a large scale of local and low-level features, which makes it suitable for the multi-label text classification task that informative snippets relevant to each label are not shared. Also, SWAM addresses the challenges of interpretability: it provides a satisfactory explanation of the internal mechanics of the deep learning method by establishing the correspondence between "informative snippet" and convolution filter.

## Discussion

For future work, we are considering several different directions. From the application perspective, since our approach does work well on 50 labels task, the next step is to apply the approach to the full code set. A major challenge is for full code set, we may need tens of thousands of convolution filters, as the number of filters in the network increases, unnecessary overlap in the features captured by the network's filters will also increase [32]. We plan to address this challenge by adapting the ensemble method, we plan to cluster the ICD codes and train a classifier for each clustered subset. From the linguistic perspective, we plan to explore reasons behind hard-to-learn codes such as ICD-9 code 285.9 "Anemia, unspecified", and leverage the hierarchy of ICD codes to improve performance on these codes. From the architecture perspective, in the current model, the filter size has to be fixed before training, which means the model can only learn the "informative snippet" less than a certain fixed length. A possible solution to this limitation is to ensemble sub-CNN models with different filter sizes, we would like to explore this direction in the future.

## Conclusion

Our main contributions can be summarized into the following three: 1) SWAM has achieved a significant improvement in the overall performance of the automatic coding task by emphasizing the learning of local and low-level features, thus validating that local and low-level features, a.k.a. informative snippets play an important role in the automatic ICD coding task. The informative snippets extracted from the clinical text provide explanations for each code, which address half of the explanatory challenge mentioned in Background that the model should provide explanations for its predictions. 2) Through ablation experiment on the width of the network, we validate that there exists a correspondence between a convolution filter and a local and low-level feature, and a combination of wide and shallow convolutional layer and attention layer can help the CNN-based models better learn local and low-level features. This finding can help understand the internal mechanics of deep learning methods on tasks like automatic coding, thus bringing progress in the other half of the explanatory challenge mentioned in Background. 3) We improved the precision of the worst-performing 10% labels from 0 to 53% on average.

#### Abbreviations

ICD: International Classification of Diseases; CNN: Convolutional neural network; ICU: Intensive care unit; SWAM: Shallow and wide attention convolutional mechanism; MIMIC-III: Medical information mart for intensive care III; NLP: Natural language; CBOW: Continuous bag-of-words; BCE: Binary cross-entropy; CAML: Convolutional attention for multi-label classification; F1: F-measure.

#### Acknowledgements

None.

#### About thissupplement

This article has been published as part ofBMC Medical Informatics and Decision Making Volume 21 Supplement 9 2021:Health Natural Language Processing and Applications. The full contents of thesupplement are available athttps://bmcmedinformdecismak.biomedcentral.com/articles/supplements/volume-21-supplement-9

#### Author's contributions

HSY and TF conceived of the presented idea. HSY developed the theory and carry out the experiments. TF analysed the results of the model and supervised the fifindings of this work. HSY wrote the manuscript under supervising from TF. All authors discussed the results. YJ and ZHB are responsible for the test on the baseline models and the statistical signifificance analysis. HLF performs a human evaluation of the quality of the explanations provided by the models. All authors have read and approved the final manuscript.

**Funding**
This work (including publication costs) is funded by the Sichuan Key R&D Project (2021YFG0136) and the Fundamental Research Funds for the Central Universities (2682020ZT92). The funders did not play any role in the design of the study, the collection, analysis, and interpretation of data, or in writing of the manuscript.

**Data availability**
The datasets is available from https://mimic.physionet.org/.

## Declarations

**Ethics approval and consent to participate**
Not applicable.

**Consent for publication**
Not applicable.

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

**Author details**
[1]School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu, China. [2]The Third People's Hospital of Chengdu, Chengdu, China. [3]AI Lab, Yidu Cloud (Beijing) Technology Co. Ltd., Beijing, China. [4]Department of Computer Science, University of Otago, Dunedin, New Zealand.

Received: 8 August 2021 Accepted: 23 August 2021
Published online: 16 November 2021

**References**

1. Denny JC, Ritchie MD, Basford MA, Pulley JM, Bastarache L, Brown-Gentry K, et al. PheWAS: demonstrating the feasibility of a phenome-wide scan to discover gene-disease associations. Bioinformatics. 2010;26(9):1205–10.
2. Ranganath R, Perotte AJ, Elhadad N, Blei DM. The survival filter: joint survival analysis with a latent time series. In: Proceedings of Uncertainty in artificial intelligence (UAI); 2015. p. 742–751.
3. Avati A, Jung K, Harman S, Downing L, Ng A, Shah NH. Improving palliative care with deep learning. BMC Med Inf Decis Mak. 2018;18(4):122.
4. Choi E, Bahadori MT, Schuetz A, Stewart WF, Sun J. Doctor AI: predicting clinical events via recurrent neural networks. In: Proceedings of Machine Learning for Healthcare Conference (MLHC); 2016. p. 301–318.
5. Zhang D, He D, Zhao S, Li L. Enhancing automatic ICD-9-CM code assignment for medical texts with pubmed. In: Proceedings of Biomedical Natural Language Processing Workshop (BioNLP); 2017. p. 263–271.
6. Kavuluru R, Rios A, Lu Y. An empirical evaluation of supervised learning approaches in assigning diagnosis codes to electronic medical records. Artif Intell Med. 2015;65(2):155–66.
7. Johnson AE, Pollard TJ, Shen L, Li-Wei HL, Feng M, Ghassemi M, et al. MIMIC-III, a freely accessible critical care database. Sci Data. 2016;3(1):1–9.
8. Subotin M, Davis A. A system for predicting ICD-10-PCS codes from electronic health records. In: Proceedings of Biomedical Natural Language Processing Workshop (BioNLP); 2014. p. 59–67.
9. Scheurwegs E, Cule B, Luyckx K, Luyten L, Daelemans W. Selecting relevant features from the electronic health record for clinical code prediction. J Biomed Inf. 2017;74(1):92–103.
10. Wang S, Chang X, Li X, Long G, Yao L, Sheng QZ. Diagnosis code assignment using sparsity-based disease correlation embedding. IEEE Trans Knowl Data Eng. 2016;28(12):3191–202.
11. Prakash A, Zhao S, Hasan SA, Datla V, Lee K, Qadir A, et al. Condensed memory networks for clinical diagnostic inferencing. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence; 2017. p. 3274–3280.
12. Mullenbach J, Wiegreffe S, Duke J, Sun J, Eisenstein J. Explainable prediction of medical codes from clinical text. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2018;1(1):1101–11.
13. Shi H, Xie P, Hu Z, Zhang M, Xing EP. Towards automated ICD coding using deep learning; 2017. p. 1–11. arXiv preprint arXiv:171104075.
14. Allamanis M, Peng H, Sutton C. A convolutional attention network for extreme summarization of source code. In: International conference on machine learning (ICML); 2016. p. 2091–2100.
15. Yin W, Schütze H, Xiang B, Zhou B. Abcnn: attention-based convolutional neural network for modeling sentence pairs. Trans Assoc Comput Linguist. 2016;4:259–72.
16. Santos CD, Tan M, Xiang B, Zhou B. Attentive pooling networks; 2016;p. 1–10. arXiv preprint arXiv:160203609.
17. Yin W, Schütze H. Attentive convolution: equipping CNNs with RNN-style attention mechanisms; 2017;p. 1–16. arXiv preprint arXiv:171000519.
18. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: Proceedings of 3rd International Conference on Learning Representations; 2014. p. 1–15.
19. Yang Z, Yang D, Dyer C, He X, Smola A, Hovy E. Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2016. p. 1480–1489.
20. Rush AM, Chopra S, Weston J. A neural attention model for abstractive sentence summarization. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing; 2015. p. 379–389.
21. Rocktäschel T, Grefenstette E, Hermann KM, Kočiskỳ T, Blunsom P. Reasoning about Entailment with Neural Attention. In: 4th International Conference on Learning Representations(ICLR); 2015. p. 1–9.
22. Le HT, C, Denis A. Do convolutional networks need to be deep for text classification? In: AAAI Workshop on Affective Content Analysis. 2017;p. 1–12.
23. Collobert R, Weston J. A unified architecture for natural language processing: Deep neural networks with multitask learning. Proceedings of the 25th international conference on Machine learning. 2008;7(3):160–7.
24. Gong L, Ji R. What Does a TextCNN Learn? 2018. p. 1–9. arXiv preprint arXiv:180106287.
25. Kim Y. Convolutional Neural Networks for Sentence Classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2014. p. 1746–1751.
26. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. In: Proceedings of Advances in Neural Information Processing Systems; 2013. p. 3111–3119.
27. McCallum AK. Multi-label text classification with a mixture model trained by EM. In: Proceedings of Association for the Advancement of Artificial Intelligence 99 workshop on text learning; 1999. p. 1–7.
28. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: Proceedings of Advances in Neural Information Processing Systems; 2017. p. 5998–6008.
29. Rajkomar A, Oren E, Chen K, Dai AM, Hajaj N, Hardt M, et al. Scalable and accurate deep learning with electronic health records. NPJ Digit Med. 2018;1(1):1–10.
30. Lyu S, Liu W. Estimation methods of p value of nonparametric test based on the Bootstrap idea. J Fuzhou Univ (Natural Science Edition). 2018;46(221(01)):20–6.
31. Aghaebrahimian A, Cieliebak M. Hyperparameter tuning for deep learning in natural language processing. In: Proceedings of 4th Swiss Text Analytics Conference (SwissText); 2019. p. 1–7.
32. Prakash A, Storer J, Florencio D, Zhang C. Repr: improved training of convolutional filters. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2019. p. 10666–10675.

## Publisher's Note