Improving Medical Code Prediction from Clinical Text via Incorporating Online Knowledge Sources

Tian Bai
Temple University
Philadelphia, PA, USA
tue98264@temple.edu

Slobodan Vucetic
Temple University
Philadelphia, PA, USA
vucetic@temple.edu

ABSTRACT

Clinical notes contain detailed information about health status of patients for each of their encounters with a health system. Developing effective models to automatically assign medical codes to clinical notes has been a long-standing active research area. Despite a great recent progress in medical informatics fueled by deep learning, it is still a challenge to find the specific piece of evidence in a clinical note which justifies a particular medical code out of all possible codes. Considering the large amount of online disease knowledge sources, which contain detailed information about signs and symptoms of different diseases, their risk factors, and epidemiology, there is an opportunity to exploit such sources. In this paper we consider Wikipedia as an external knowledge source and propose **K**nowledge **S**ource **I**ntegration (KSI), a novel end-to-end code assignment framework, which can integrate external knowledge during training of any baseline deep learning model. The main idea of KSI is to calculate matching scores between a clinical note and disease related Wikipedia documents, and combine the scores with output of the baseline model. To evaluate KSI, we experimented with automatic assignment of ICD-9 diagnosis codes to the emergency department clinical notes from MIMIC-III data set, aided by Wikipedia documents corresponding to the ICD-9 codes. We evaluated several baseline models, ranging from logistic regression to recently proposed deep learning models known to achieve the state-of-the-art accuracy on clinical notes. The results show that KSI consistently improves the baseline models and that it is particularly successful in assignment of rare codes. In addition, by analyzing weights of KSI models, we can gain understanding about which words in Wikipedia documents provide useful information for predictions.

KEYWORDS

Multi-label classification; document similarity learning; attention mechanism; healthcare

ACM Reference Format:
Tian Bai and Slobodan Vucetic. 2019. Improving Medical Code Prediction from Clinical Text via Incorporating Online Knowledge Sources. In *Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3308558.3313485

1 INTRODUCTION

Clinical notes are free-text documents generated by healthcare professionals documenting clinical status of patients during hospital stay or outpatient care. Clinical notes contain various nuanced information such as a history of illness, lifestyle, symptoms, treatment, and medications. Clinical notes are used to facilitate communication between medical professionals treating the same patient over time, to provide a basis for billing, and to create a record trail in case of legal issues. Medical coding, which refers to translating a clinical note into a set of medical codes from a medical ontology, is necessary to obtain reimbursement for the provided healthcare services. The most popular medical coding ontology is the International Classification of Diseases (ICD, with the most recent versions ICD-9 and ICD-10), whose codes provide alphanumeric encoding of diagnoses and treatments as shown in Table 1. Interestingly, beyond their primary billing purpose, medical codes have been widely used in healthcare research such as predictive modeling [1, 3, 4, 8, 33, 34] and retrospective epidemiologic studies [24, 36, 40]. The reason for popularity of medical codes in research is that it abstracts away fine details of care from the free-text clinical notes and normalizes a patient encounter to a standardized computer-readable format.

Medical coding is typically performed by professional staff trained to understand clinical notes containing complex medical terminology with frequent misspellings and abbreviations and transcribe the notes into a limited set of appropriate medical codes from a large menu of options (e.g., ICD-10 contains more than 100,000 codes). Thus, medical coding is an expensive, time consuming, and inexact process. Due to these issues, there has been a significant interest in developing automatic code assignment models and it has been a long-standing research challenge [6, 10, 11, 13, 22, 25, 29–31, 35, 37, 39].

Recently, deep neural networks with attention mechanism have been shown to achieve the state-of-the-art performance on automatic code assignment task [6, 29, 35, 39]. The attention mechanism [2, 26] uses one or more layers of neurons to identify words indicative of a specific disease and assigns them large weights. The success of the attention mechanism could be attributed to the property of medical code assignment: the relevant information for a specific medical code is often located in a small portion of a clinical note.

Despite the success of deep neural networks with the attention mechanism, there are still outstanding challenges that remain to be addressed. In this paper we address an issue related to recognition of rare codes. To illustrate the issue, let us consider MIMIC-III [18], the largest publicly-available database containing 59,652 discharge notes from the Beth Israel Deaconess Medical Center's Intensive Units. There are 942 unique 3-digit ICD-9 codes occurring in the dataset in a highly unbalanced way, where the most common 10

WWW '19, May 13–17, 2019, San Francisco, CA, USA

ACM ISBN 978-1-4503-6674-8/19/05.
https://doi.org/10.1145/3308558.3313485

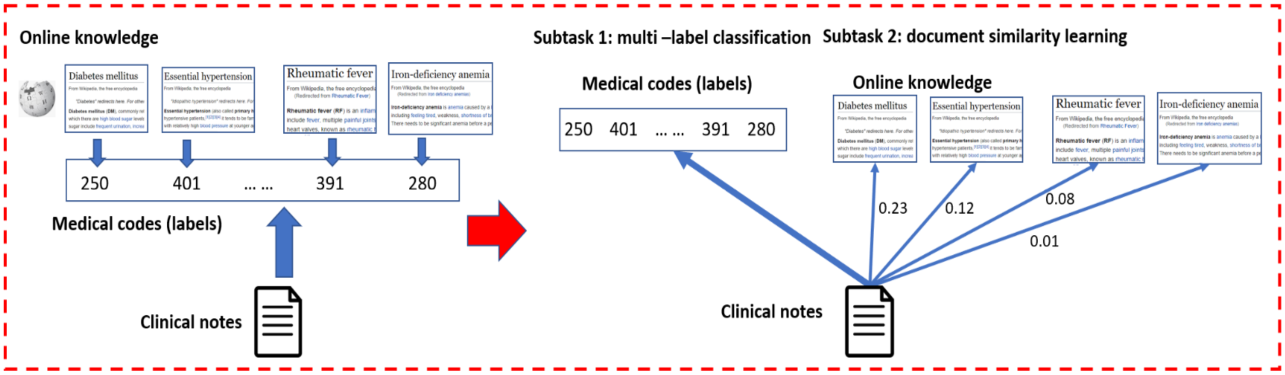

Figure 1: Our main goal is to predict medical codes from clinical notes by taking advantage of online knowledge sources. We decompose the task into two parallel subtasks: the first subtask is traditional multi-label classification and the second subtask is document similarity learning.

Table 1: Examples of 3-digit ICD-9 codes and their descriptions (249-259).

ICD-9 code	Description
249	Secondary diabetes mellitus
250	Diabetes mellitus
251	Other disorders of pancreatic internal secretion
252	Disorders of parathyroid gland
253	Disorders of the pituitary gland and its hypothalamic control
254	Diseases of thymus gland
255	Disorders of adrenal glands
256	Ovarian dysfunction
257	Testicular dysfunction
258	Polyglandular dysfunction and related disorders
259	Other endocrine disorders

Table 2: Wikipedia webpage for ICD-9 code “250" (partially shown).

ICD-9 code: 250 (Diabetes mellitus)
Diabetes mellitus
Diabetes mellitus (DM), commonly referred to as diabetes, is a group of metabolic disorders in which there are high blood sugar levels over a prolonged period. Symptoms of high blood sugar include frequent urination, increased thirst, and increased hunger. If left untreated, diabetes can cause many complications. Acute complications can include diabetic ketoacidosis, hyperosmolar hyperglycemic state, or death. Serious long-term complications include cardiovascular disease, stroke, chronic kidney disease, foot ulcers, and damage to the eyes.

codes account for 26% of all code occurrences and the least common 437 codes account for only 1% of the occurrences. Thus, there is highly limited information available for training a neural network to recognize such rare codes. The rare codes problem is particularly severe considering the length of clinical notes: each MIMIC-III discharge note contains an average of 1,596 words. Even with the attention mechanisms, it would be very difficult to learn to identify relevant portions of text from only a few examples.

In this paper we consider exploiting online knowledge sources to improve accuracy of medical code prediction from clinical notes. There are rich disease-related online knowledge sources such as Mayo Clinic, Wikipedia, and Medscape, which contain detailed information such as signs, symptoms, and epidemiology useful for differentiating different diseases in the clinical notes. For example, as shown in Table 2, from a Wikipedia article introducing diabetes, we learn that the typical signs of diabetes include frequent urination, increased thirst, and increased hunger. If a clinical note also mentions in a passage that a patient is having these symptoms, then the passage could be used as an evidence for diabetes. In this paper we use Wikipedia, because it is an easily accessible online resource with loose license constraints that has already been widely studied in the research community. In order to take advantage of the existing state of the art neural networks for medical code prediction, instead of building a completely new model, we develop a framework called **K**nowledge **S**ource **I**ntegration (KSI), which is able to enhance the state of the art neural networks with information from the external knowledge sources. As illustrated in Figure 1, the proposed architecture consists of two components. The first component is directly predicting codes from a note, and it could be any of the state of the art neural networks. Our main contribution is the second component, which is a neural network whose objective is to learn similarity scores between a note and Wikipedia articles about various diseases. Each similarity score measures how closely a clinical note is associated with a corresponding medical code. Finally, the proposed architecture combines the similarity scores with the output of the first subtask and learns to optimize both components to minimize the loss between the predictions and the true labels.

Using Wikipedia as an external knowledge source for analyzing clinical notes poses a couple of challenges. First, there are significant differences between content of Wikipedia articles and clinical notes. Wikipedia articles are mostly carefully edited and the terminology used is optimized toward the general audience. Since Wikipedia can be edited by anyone with Internet access, there is a certain risk that the provided information is incorrect. On the contrary, clinical notes are written by medical professionals in a hurry and contain a lot of non-standard abbreviations, misspellings and specialized medical terminology. Second, the coverage of Wikipedia and clinical notes are very different. As a knowledge base, Wikipedia elaborates different aspects of diseases including pathophysiology, prevention, history, and social impact, which are unlikely to appear in clinical notes. On the other hand, clinical notes contain patient conditions, including medications on admission and allergies, which are unlikely to occur in Wikipedia articles. In summary, these two types of documents have very different data distributions, which makes unsupervised document similarity measures such as Word Mover's Distance [21] unsuitable, as we need to carefully deal with noise and misleading information in the documents and focus on aspects important for medical code prediction.

In order to properly handle the above challenges, in this paper we develop a supervised document similarity learning model, which is designed to measure the similarity of two types of documents with very different data distributions. To accomplish this, we first take intersection of a clinical note and a Wikipedia document. The intersection operation discards disparate parts of the text and focuses on the common aspects of the two documents. Then, we project the intersection into a low dimensional vector space to reduce dimensionality. Since terms remaining in the intersection might have different importance for medical code prediction, next we apply an attention mechanism to weight the terms. Finally, the weighted intersection vector is transformed into a matching score, which measures the similarity between the two documents. All matching scores are combined with the output of the baseline neural network model and trained to predict medical codes.

In our experiments we selected a variety of baseline models, from simple logistic regression to a recently proposed attention based deep neural network, which achieves the state-of-the-art performance. We show that our KSI framework of incorporating external knowledge sources consistently improves the baseline models. Moreover, thorough analysis shows that KSI significantly improves accuracy of rare code prediction. Finally, by analyzing attention values and weights of the model, our framework reveals which words in Wikipedia documents are the most influential to the prediction, which is important for understanding the reasoning behind the prediction.

Our work makes the following contributions:

- We propose a novel end-to-end framework to integrate external knowledge sources with any end-to-end baseline neural network model for the task of medical code prediction from clinical notes.
- We empirically show that our framework improves a variety of baseline models, including the state of the art attention based neural networks, and that the improvement is particularly large on rare codes.
- We demonstrate that our framework allows interpretability via analyzing attention values and weights of the model, which sheds light on the connection between the external knowledge source and clinical text.

We organize the rest of the paper as follows: in Section 2 we discuss the related previous work. Section 3 presents our KSI framework. In Section 4 we explain the experiment design. The results are shown and discussed in Section 5.

2 RELATED WORK

2.1 Prediction of Medical Codes from Clinical Notes

Automatic assignment of medical codes is a long-standing research area dating back to 1990s [11]. [11] represents clinical notes and ICD-9 codes as two TF-IDF vectors and calculates cosine similarity in the joint vector space. Later, a variety of methods were applied to this area, ranging from rule-based methods [10, 13] to machine learning methods such as K-nearest-neighbor, relevance feedback, Bayesian independence classifiers [22] and support vector machines [25]. In [30, 31] the authors develop methods which exploit the hierarchical tree structure of ICD-9 ontology. More recent models combine neural networks with attention mechanism. In [6], the authors apply hierarchical attention networks [41], which calculate both word-level attention and sentence-level attention while predicting medical codes. In [35], the authors apply character-aware neural network to generate hidden representations of ICD-9 codes and diagnosis section in clinical notes and use the attention mechanism to match these two hidden representations. Their ICD-9 descriptions are quite short (typically, less than 10 words), while in our case a Wikipedia document contains thousands of words providing multiple views of a disease. In [37] the authors propose a modified GRU, where each dimension in the hidden vector corresponds to a specific label, and use their model on medical code prediction. In [29], the authors combine the attention mechanism with a convolutional neural network [19] and report state-of-the-art accuracy on a medical prediction task. In [39], the authors project both words and labels into the same low-dimensional vector space and calculate the compatibility/attention between them. Then, they combine the attention values with word embeddings to get the final document representation. None of the above approaches incorporate external knowledge sources on the task of medical code prediction.

In [32], the authors use memory network to infer clinical diagnosis from clinical notes. While they also use Wikipedia as knowledge source, their work is different from ours in several ways: first, their focus is on a different task, which uses a much smaller diagnosis label set directly derived from clinical notes, while we use accurate medical codes as labels. Second, their memory network is highly specialized and hard to generalize, while our framework can incorporate knowledge into any end-to-end neural network model developed in this field. Lastly, their model is hard to interpret, while our attention based model sheds light on how knowledge source and notes are connected.

2.2 Attention Mechanism

The attention mechanism was first proposed in machine translation [2, 26] to select the most relevant words in a source sentence while predicting the next word in the target sentence. Typically, a neural network is applied to the context vector, generated by either Recurrent Neural Network (RNN) or Convolutional Neural Network (CNN), and attention values are expected to capture the importance of the context vectors. Final sentence or document representations are simply the sum of context vectors weighted by the attention values. The attention mechanism is widely used in natural language processing [2, 6, 29, 35, 38, 39, 41] and sequence modeling in healthcare domain [5, 9, 27, 33]. Our variable-level attention mechanism is the most similar to [9] because we compute attention on different vector elements. Our work is different in a way that we focus on learning attention of clinical words, instead of medical codes.

2.3 Document Similarity Learning

Document similarity computation transforms two documents into a score depicting how similar the two documents are. Documents are typically transformed into vectors, with each dimension corresponding to the statistic of a word (count, occurrence or TF-IDF [16]) in the vocabulary. Then, similarity between two documents is calculated as the cosine similarity between the two vectors. Other methods transform documents into low-dimensional vectors which capture semantics of documents, such as singular vector decomposition [12] or latent dirichlet allocation [7]. More recently, doc2vec [23] uses architecture similar to word2vec [28] in order to learn document emebddings. Word Mover's Distance [21] exploits high quality word embeddings and treats the document distance as an instance of Earth Mover's Distance. The above approaches are unsupervised. Supervised approaches such as supervised Word Mover's Distance [17] and neighborhood components analysis [14] consider label information while learning similarity between two vectors. However, there is no explicit way to handle the situation where each instance has multiple labels. Moreover, as mentioned before, Wikipedia and clinical notes have very different distributions, thus making direct measurement of the similarity unsuitable. On the contrary, our approach first calculates the overlap between two documents and then uses the attention mechanism to select the most informative parts for medical code prediction.

3 METHODOLOGY

In this section we describe the proposed KSI framework. We first introduce the problem setup and our goal. Then, we describe the details of the proposed approach. Finally, we show how to interpret the learned model via analyzing attention values and weights of neural network associated with words in clinical notes and Wikipedia documents.

3.1 Basic Problem Setup

Assume our dataset is a collection of clinical notes. Each note N consists of a sequence of words. N is accompanied with a set of medical codes M corresponding to the note. We denote the size of word vocabulary $W = \left\{w_1, w_2, ..., w_{|W|}\right\}$ as $|W|$ and the size of medical code vocabulary $C = \left\{c_1, c_2, ..., c_{|C|}\right\}$ as $|C|$. In addition, we have an external knowledge source K which contains a set of documents $K = \left\{k_1, k_2, ..., k_{|C|}\right\}$ in which each document k_i is a sequence of words. Each unique medical code c_i in the medical code vocabulary C has a corresponding document k_i in K. Given a clinical note N, our goal is to predict associated medical codes M when knowledge source K is available. This is naturally a multi-label text classification problem.

3.2 KSI Framework

As shown in Figure 1, when each label (*i.e.*, medical code) c_i has a document k_i in knowledge source K, we can calculate the likelihood of label c_i as the similarity between document k_i and note N. Therefore, we decompose the prediction task into two parallel subtasks. As shown in Figure 2, the first subtask is to directly predict codes from notes, which can be achieved by any end-to-end multi-label classification model in this field. We denote the baseline model as f, which is a mapping from notes N to the label space. The last layer of f is a vector $o_1 \in \mathbb{R}^{|C|}$

$$o_1 = f(N). \tag{1}$$

Since the task is multi-label classification, a sigmoid activation function is applied to o_1 and the i-th dimension in the resulting vector represents the probability that code c_i is the true label. Loss values are calculated between the output and the true labels. In KSI framework o_1 is combined with the output of subtask 2 to get the final output. We select a variety of baseline models and describe the details in the next section. Our main focus is the second subtask, *i.e.*, to learn similarity scores between Wikipedia documents and a note.

First, we represent note N as binary vector $q \in \{0, 1\}^{|W|}$ and Wikipedia document k_i as binary vector $x_i \in \{0, 1\}^{|W|}$. Binary vectors encode the presence of words. If word occurs in the document, then the corresponding element has value 1, otherwise it is 0. By encoding documents as binary vectors we lose the frequency information. However, as mentioned before, the frequency does not necessarily reflect the importance of words for code prediction since these two types of documents have very different data distributions, *e.g.*, in the Wikipedia document about "influenza", word "birds" has high frequency because there is one section in the document talking about "bird flu". As we formulate the similarity learning in a supervised framework, we expect the network to be able to assign large weights to important words for the task.

After we represent both N and k_i as binary vectors, we calculate their element-wise multiplication as

$$z_i = x_i \odot q, \tag{2}$$

where $z_i \in \{0, 1\}^{|W|}$ represents the intersection of two documents. Value of words that appear in both documents equals 1, otherwise it is 0. By taking the intersection operation, we discard irrelevant parts in two documents and focus on the common content between them.

Then, we use an embedding matrix $W_e \in \mathbb{R}^{m \times |W|}$ to transform the high-dimensional sparse vector z_i into a low-dimensional vector space

$$e_i = W_e z_i, \tag{3}$$

where $e_i \in \mathbb{R}^m$ denotes the embedding of z_i and m is the dimensionality of the embedding space. e_i is expected to capture the semantic information about the intersection set. In e_i, each element has its own latent meaning and represents a hidden topic. Since not all topics are relevant to the prediction task, we use a variable-level attention mechanism to weight variables in the vector. We use weight matrix $W_a \in \mathbb{R}^{m \times m}$ to calculate the attention values as

$$\alpha_i = sigmoid(W_a e_i) \tag{4}$$

$$v_i = \alpha_i \odot e_i, \tag{5}$$

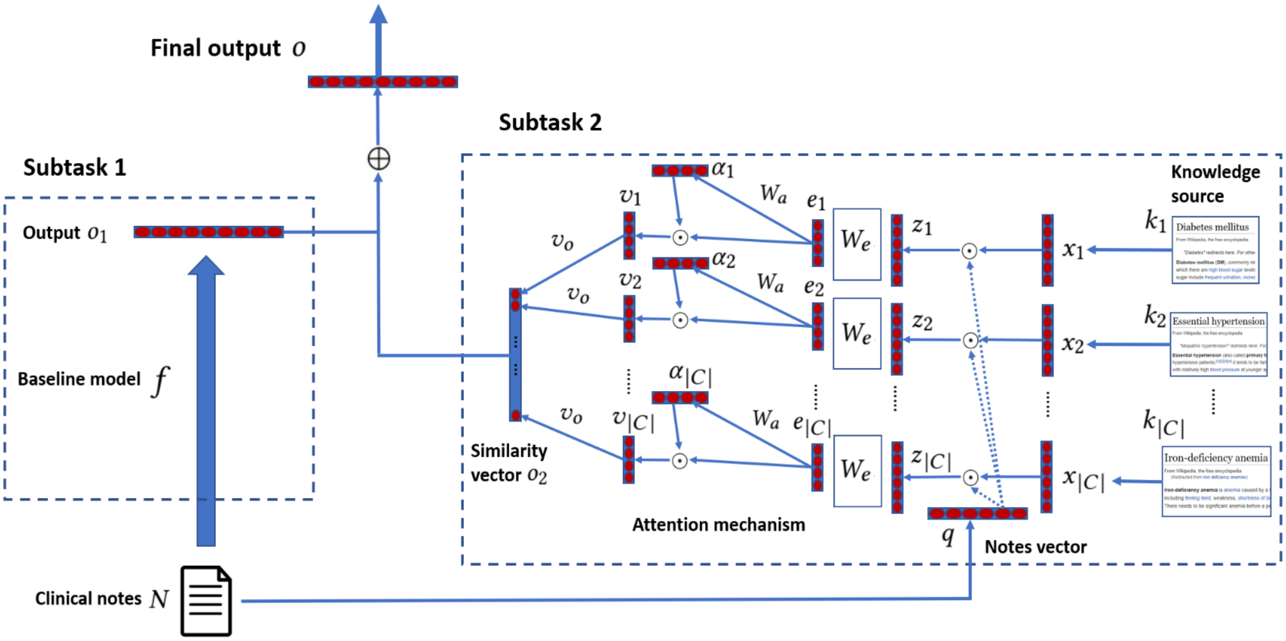

Figure 2: The illustration of our KSI framework. First, it represents document k_i and clinical note N as binary vectors x_i and q. Then, it performs element-wise multiplication of x_i and q to obtain the intersection vector z_i. Next, it projects z_i into a low-dimensional vector e_i. Next, the attention mechanism is used to transform e_i into a weighted vector v_i. Finally, it projects v_i into a similarity score and the concatenation of all scores is combined with the output of baseline model to get the final output.

where $\alpha_i \in \mathbb{R}^m$ is the attention weight vector and $sigmoid()$ is the sigmoid activation function. $v_i \in \mathbb{R}^m$ is the weighted embedding which is the element-wise multiplication of α_i and e_i.

Next, we use a vector $v_o \in \mathbb{R}^m$ to transform v_i into a similarity score s_i as

$$s_i = v_o^T v_i + b_o, \tag{6}$$

where s_i depicts how similar Wikipedia document k_i and note N are and b_o is a scalar bias term. We concatenate the similarity scores for all $k_i \in K$ to get the similarity vector $o_2 \in \mathbb{R}^{|C|}$,

$$o_2 = [s_1, s_2, ..., s_{|C|}]. \tag{7}$$

Finally, the similarity vector o_2 is combined with the output of the baseline model o_1 to get the final output vector $o \in \mathbb{R}^{|C|}$ as

$$o = o_1 + o_2 \tag{8}$$

$$\hat{y} = sigmoid(o), \tag{9}$$

where $\hat{y} \in \mathbb{R}^{|C|}$ is the predicted label vector. True labels are also represented as a binary vector $y \in \{0,1\}^{|C|}$ and the training procedure minimizes the binary cross-entropy loss between the predicted values and the true values,

$$L_{BCE}(N, y) = -(\sum_{i=1}^{|C|} y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - \hat{y_i})), \tag{10}$$

where we sum the binary cross entropy errors over all dimensions of $\hat{y}$. Note that equation (10) is the loss for one note. In our implementation, we calculate the average loss for multiple notes. We use Adam optimizer [20] to minimize the above loss values. For deep learning based models (CNN and RNN based models), it typically takes a long time to converge. Therefore, in our training, we first deactivate the document similarity learning part of the model to let the baseline model converge. We activate the document similarity learning part and disable parameters updating in the baseline model when the performance of baseline model stops increasing on the validation set as the early stopping criteria.

3.3 Interpreting KSI framework

Interpretability is a model's ability to provide rationale behind its behavior. In KSI framework, the larger the s_i is, the more evidence Wikidedia knowledge source provides for predicting code c_i. To understand what evidence Wikipedia knowledge source provides, we need to calculate the contribution of each variable in x_i to the final similarity score. By combining equations (6) and (5), we get

$$s_i = v_o^T v_i + b_o = v_o^T(\alpha_i \odot e_i) + b_o. \tag{11}$$

Since α_i are weights of e_i, we keep them fixed and decompose e_i. As in equation (3), e_i is the sum of the columns of W_e weighted by each variable in z_i, and we can rewrite equation (11) as

$$\begin{aligned} s_i =& v_o^T(\alpha_i \odot (\sum_{j=1}^{|W|} z_{i,j} W_e[:, j])) + b_o \\ =& \sum_{j=1}^{|W|} v_o^T(\alpha_i \odot W_e[:, j]) z_{i,j} + b_o. \end{aligned} \tag{12}$$

Using equation (2), we can further decompose $z_{i,j}$ as the product of $x_{i,j}$ and q_j,

$$s_i = \sum_{j=1}^{|W|} v_o^T(\alpha_i \odot W_e[:,j]q_j)x_{i,j} + b_o. \tag{13}$$

Since $x_{i,j}$ is a binary value indicating the presence of word w_j in document k_i, $\lambda(x_{i,j}) = v_o^T(\alpha_i \odot W_e[:,j]q_j)$ is the coefficient for $x_{i,j}$ and measures how much word w_j in k_i contributes to final similarity s_i.

4 EXPERIMENTAL SETUP

4.1 DataSets

In our experiments we use clinical notes in MIMIC-III dataset and Wikipedia pages of ICD-9 diagnosis codes. Both datasets are publicly available. The source code of KSI is available at https://github.com/tiantiantu/KSI.

Clinical notes dataset: MIMIC-III Critical Care Database [18] is the largest publicly available electronic health records dataset which contains de-identified health records of 46,518 patients who stayed in the Beth Israel Deaconess Medical Center's Intensive Units from 2001 to 2012. We use all discharge summary notes and the accompanying ICD-9 diagnosis codes. The total number of discharge summary notes is 59,652. For each patient visit, we aggregated all discharge summary notes. The number of aggregated discharge summary notes is 52,722. On average, in each aggregated note there are 1,596 words. During preprocessing we lowercased all tokens and removed punctuations, stop words, words containing only digits, and words whose frequency is less than 10. The final word vocabulary contains 47,965 unique words. We extracted all listed ICD-9 diagnosis codes for each visit and grouped them by their first three digits. On average, each visit has 11 medical codes. The code vocabulary contains 942 codes. Of those codes, we selected a subset of 344 codes for which we found the corresponding Wikipedia document and used those codes in our experiments.

Wikipedia knowledge dataset: Wikipedia maintains a web page including all ICD-9 diagnosis codes and links to their web pages if available[1]. Out of the 389 Wikipedia pages available for the first three digits ICD-9 diagnosis codes, we found 344 of them in MIMIC-III medical code vocabulary. We extracted Wikipedia documents for these 344 medical codes and transformed them into pure text. Then we removed punctuations and lowercased all tokens. On average each processed document has 1,058 words. The size of the word vocabulary of Wikipedia documents is 60,968, out of which only 12,173 are also in the word vocabulary of MIMIC-III clinical notes, showing that the two types of documents have very different word distributions.

4.2 Baseline model selection

We selected 5 baseline models f for this study. In our experiments, we use them stand-alone and as a component of the KSI framework that incorporates Wikipedia knowledge in order to evaluate the improvement the KSI framework brings. We briefly introduce the selected baseline models below.

Logistic regression (LR). First, we represent note N as binary bag-of-words vector $n \in \{0,1\}^{|W|}$. Then, we use a weight matrix $W_{LR} \in \mathbb{R}^{|C|\times|W|}$ to map feature space to label space as

$$o_1 = W_{LR}n. \tag{14}$$

For all baseline models, when combined with the KSI framework we use equations (8) and (9) to get the predicted values. When used as a stand-alone model, we apply sigmoid function to its output,

$$\hat{y} = sigmoid(o_1). \tag{15}$$

Recurrent neural network (RNN). We first project each word in note N to a low-dimensional vector $emb \in \mathbb{R}^m$. Then we transform the sequence of words into a sequence of word embeddings and feed it to a recurrent neural network,

$$h_1, h_2, ..., h_{|N|} = RNN(emb_1, emb_2, ..., emb_{|N|}), \tag{16}$$

in which hidden vector $h_i \in \mathbb{R}^{d_o}$ captures the context information of emb_i. Then, max-pooling is used to compute the sequence vector $g \in \mathbb{R}^{d_o}$ and transform it into output o_1 as

$$g(i) = \max_k h_{k,i} \tag{17}$$

$$o_1 = W_{RNN} \cdot g + b_{RNN}, \tag{18}$$

where $W_{RNN} \in \mathbb{R}^{|C|\times d_o} \in$ and $b_{RNN} \in \mathbb{R}^{|C|}$. In our experiments we use the long short-term memory network [15] as the specific RNN selection.

RNN with attention (RNNatt). Instead of using max-pooling of hidden vectors as the sequence vector, we can apply label-specific attentions to them. Let $H = [h_1, h_2, ..., h_{|N|}]$. For each label $c_j \in C$ we use $u_j \in \mathbb{R}^{d_o}$ to calculate the attention as

$$\alpha_j = softmax(H^T u_j), \tag{19}$$

where $softmax = \frac{\exp(x)}{\sum_i \exp(x_i)}$ and $\exp(x)$ is the element-wise exponentiation of x. Then we calculate vector representation for label c_j as

$$p_j = \sum_{k=1}^{|N|} \alpha_{j,k} h_k, \tag{20}$$

and the final likelihood for label c_j in o_1 is

$$o_1(j) = \beta_j^T p_j + b_j, \tag{21}$$

in which $\beta_j \in \mathbb{R}^{d_o}$ and b_j is a scalar bias.

Convolutional neural network (CNN) [19]. As in the RNN case, we first project each word in note N to a low-dimensional vector $emb \in \mathbb{R}^m$ and transform the sequence of words into a sequence of word embeddings. Next, we use a convolutional filter $W_{CNN1} \in \mathbb{R}^{k\times m\times d_o}$ to aggregate context information, where k is the filter width and d_o is the size of filter output. Then for each word embedding emb_n, we use W_{CNN1} to capture its context as

$$h_n = ReLU(W_{CNN1} * emb_{n:n+k-1} + b_{CNN1}), \tag{22}$$

where $*$ is the convolution operator, $ReLU$ is the rectified linear unit and $b_{CNN1} \in \mathbb{R}^{d_o}$ is the bias. We pad both sides of the input to make $H = [h_1, h_2, ..., h_{|N|}] \in \mathbb{R}^{d_o\times|N|}$. Here, as in the RNN case, H is the matrix capturing context information of each word. In [19], max-pooling is used to compute the sequence vector as

$$g(i) = \max_k h_{k,i}. \tag{23}$$

[1] https://en.wikipedia.org/wiki/List_of_ICD-9_codes

Table 3: Label frequency distribution

Frequency range	Number of unique codes	Percentage of code occurrences
1-10	80	0.1%
11-50	73	0.6%
51-100	25	0.6%
101-500	82	6.7%
>500	84	92.0%

Finally, $g \in \mathbb{R}^{d_o}$ is mapped to the label space:

$$o_1 = W_{CNN2} \cdot g + b_{CNN2}, \tag{24}$$

where $W_{CNN2} \in \mathbb{R}^{|C| \times d_o}$ and $b_{CNN2} \in \mathbb{R}^{|C|}$.

Convolutional Attention (CAML) [29]: CAML combines the attention mechanism with CNN and achieves the state-of-the-art performance on medical code prediction. After calculating context matrix H in equation (22), instead of applying max-pooling in equation (23), CAML applies label-specific attentions following equations (19) (20) (21). For each label $c_j \in C$, we use $u_j \in \mathbb{R}^{d_o}$ to calculate the attention as

$$\alpha_j = softmax(H^T u_j). \tag{25}$$

Then we can calculate vector representation for label c_j as

$$p_j = \sum_{k=1}^{|N|} \alpha_{j,k} h_k, \tag{26}$$

and the final likelihood for label c_j in o_1 as

$$o_1(j) = \beta_j^T p_j + b_j, \tag{27}$$

where $\beta_j \in \mathbb{R}^{d_o}$ and b_j is a scalar bias.

4.3 Evaluation metric

In order to properly evaluate different models, we report six different metrics: average loss value, top-10 recall, macro-averaged AUC, micro-averaged AUC, macro-averaged F1 and micro-averaged F1. The average loss value is the average of binary cross-entropy defined in equation (10) over all test notes. Top-10 recall is defined as the number of correct medical codes ranked in the top max(10, $|M|$) in $\hat{y}$ divided by $|M|$, where $|M|$ is the number of medical codes for the note, and averaged over all test notes. This metric is motivated by the potential use case in computer-aided coding where the system recommends several codes for human experts to review. AUC (area under ROC curve) measures the probability that a classifier ranks a randomly chosen positive instance higher than a randomly chosen negative instance. F1 score is defined as the harmonic mean between the precision and the recall. Micro-averaged values are calculated by aggregating instances of all classes and calculating the average metric over all instances. Therefore, this metric will be dominated by the most frequent medical codes. Medical codes in our dataset (3-digit ICD-9 codes which have Wikipedia webpage) are highly imbalanced. As shown in Table 3, the most common 84 codes account for 92% of all code occurrences. On the other hand, the macro-averaged metric first calculates the value for each medical code separately and then takes the average over all the codes. Since a frequent class is weighted the same as a rare class, the macro-averaged metrics puts a higher emphasis on rare medical code prediction.

Implementation details. All models are implemented with Pytorch. We randomly divide the dataset into training, validation and test sets in a 0.7, 0.1, 0.2 ratio. Early stopping on the validation set is used to determine when to stop training. Training terminates if top-10 recall does not improve in the next 5 epochs and the model with the highest top-10 recall is used on the test set. Each batch contains 32 notes. We use Adam optimizer with the learning rate of 0.001 to minimize the loss function. The dimensionality of word embedding m and hidden vector of RNN and CNN d_o are set to 100 and 200, respectively.

5 EXPERIMENTAL RESULTS

5.1 Predictive results

We show the accuracies of the baseline models alone and with the KSI framework in Table 4. From the table we can see that regardless of the baseline model, KSI framework consistently improves the accuracy. Although different metrics measure different aspect of model performance, the KSI framework improves most of them. Overall, the micro-averaged score is much higher than the macro-averaged score, as expected because the macro-averaged score emphasizes performance on the rare medical codes on which the number of available training examples is too small for accurate learning. On the other hand, in the micro-averaged scores medical codes contribute proportionally to their frequency and the frequent medical codes such as diabetes and essential hypertension provide more examples for accurate learning.

Overall, deep learning models outperform the logistic regression model, and KSI framework improves these complicated models further, showing the effectiveness of exploiting external online knowledge sources. The attention mechanism based models (RNNatt and CAML) outperform other models while CAML achieves the best overall performance compared to the other baseline models. The success of CAML and RNNatt could be attributed to their multi-label attention mechanism. For each label, both models use a label-specific weight matrix to generate label specific attentions over all words in the text. Therefore, they are able to pay attention to locations of text segments which refer to a specific disease, which is important for medical notes since evidence about different diseases can occur in different parts of clinical notes.

It is worth noting that KSI framework improves all macro F1 and macro AUC of all models by a large margin. For macro AUC: 4.8% for LR, 3.1% for RNN, 3.0% for RNNatt, 3.4% for CNN and 3.6% for CAML. As macro AUC is the unweighted average AUC over all labels we hypothesize that the KSI framework is able to improve AUC for labels which are not so frequent in the dataset. In particular, since the frequent medical codes have a larger number of examples for accurate learning, they stand to benefit less from the external sources. On the contrary, the rare codes lack examples and are benefiting greatly from the external sources.

To validate this, we divide medical codes into 5 groups based on their frequencies in the dataset: [1, 10], [11, 50], [51, 100], [101, 500] and [500, +∞). Then, we calculate macro-averaged AUC of each medical code group for RNN, RNNatt, CNN, and CAML baselines

Table 4: The predictive performance of baseline models alone and with KSI framework.

Method	Macro AUC	Micro AUC	Macro F1	Micro F1	Test loss value	Top-10 recall
LR	0.716	0.956	0.167	0.554	**0.048**	0.724
KSI+LR	**0.764**	**0.961**	**0.196**	**0.557**	0.055	**0.738**
RNN	0.854	0.972	0.204	0.653	0.032	0.772
KSI+RNN	**0.885**	**0.978**	**0.244**	**0.662**	**0.030**	**0.798**
RNNatt	0.850	0.973	0.264	0.647	0.035	0.791
KSI+RNNatt	**0.880**	**0.976**	**0.300**	**0.659**	**0.033**	**0.802**
CNN	0.825	0.968	0.214	0.626	0.040	0.753
KSI+CNN	**0.859**	**0.973**	**0.237**	**0.637**	**0.039**	**0.775**
CAML	0.855	0.978	0.257	0.656	0.032	0.806
KSI+CAML	**0.891**	**0.980**	**0.285**	**0.659**	0.032	**0.814**

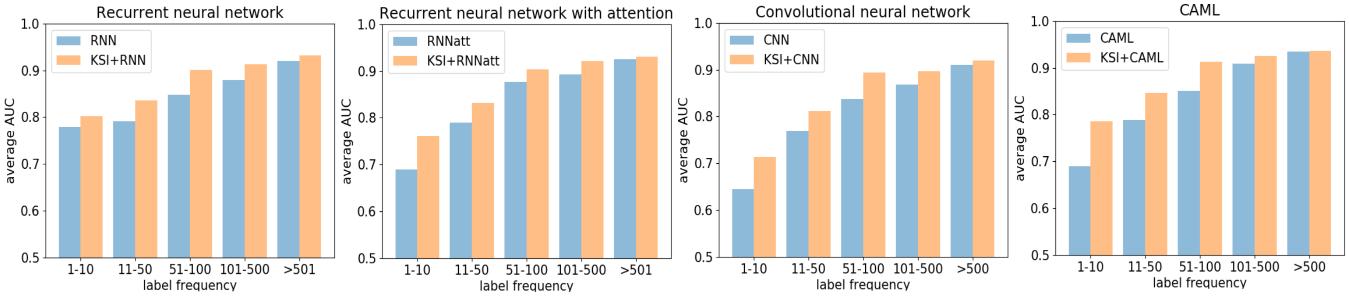

Figure 3: Macro-averaged AUC by label frequency group for RNN, RNNatt, CNN and CAML. x-axis denotes the label frequency group and y-axis denotes the macro-averaged AUC for each group.

and their counterparts under the KSI framework. The results are summarized in Figure 3.

As shown in the figure, different methods perform differently on each medical code group. Overall, the average AUC of medical code prediction increases with the code frequency. For the least frequent group (occurring less than 10 times), the average AUC of medical codes is less than 0.8. For the most common group of medical codes (occurring more than 500 times), all methods achieve an averaged AUC of more than 0.9. The figure also shows the trend in which the KSI improvement decreases with the frequency of medical codes in the dataset. For example, for CAML model, KSI framework brings 9.6% improvement of AUC on the least common [1-10] group. On the [11-50] and [51-100] groups, KSI brings 5.8% and 6.3% improvement, respectively. On the more common groups [101-500] and [500,+∞), the KSI framework only increases AUC by 1.5% and 0.1% respectively. The benefit of the KSI framework on rare medical codes can intuitively be explained in the following way. An intersection of a note and a Wikipedia document can be viewed as a feature selection; it significantly narrows down the feature space and makes it easier for the attention mechanism to find the evidence for the rare medical codes. To validate this, in the following paragraph we take a look at the contribution coefficient we derived in section 3.3.

5.2 Informative evidence extraction

The success of medical code prediction depends on the ability to extract evidence for inferring the specific codes. In equation (13), we can calculate coefficient $\lambda(x_{i,j})$ of each word w_j for document k_i. $\lambda(x_{i,j})$ depicts the contribution of each word to predicting code c_i. We calculate the coefficient for each word in the document and sort the words by their values. We study 10 words with the highest coefficients in order to gain insight into the performance of the KSI framework. In Table 5 we show the top 10 words according to the document similarity learning sub-network (subtask 2) and we also show the top 10 substrings extracted by the baseline sub-network (subtask 1). For the table, we used CAML as our baseline sub-network since it outperformed other baseline models. In equation (25), CAML calculates the attentions α_j for all words in a clinical note. These attentions are specific to medical code $c_j \in C$, the higher the attentions, the more the underlying words contribute to predicting c_j. Since in equation (22) CAML uses a convolutional layer to aggregate context information for a given word, the attentions are generated on top of the n-grams, where n equals to the filter size. Since in our experiments we set the filter size n to 3, the table is showing the top 10 trigrams with the highest attentions.

In Table 5 we show results for three representative diseases, both frequent and rare ones in order to better understand how KSI framework creates its two sub-networks.

As shown in the table, the first medical code is “250" (diabetes mellitus), which is one of the most frequent codes in our dataset. Since CAML sub-network alone is able to predict this disease with AUC as high as 0.97, the similarity sub-network cannot improve the performance any further. By checking the trigrams with the highest coefficients, we find that CAML sub-network finds highly meaningful substrings for inferring diabetes such as “failure diabetes mellitus" and “significant diabetes mellitus". By contrast, the similarity sub-network assigns the largest weight to word “name" rather than to some words that are more directly related to diabetes.

Table 5: Words/trigrams with highest contribution by document similarity learning (subtask 2) and baseline model alone (subtask 1) for medical codes 250, 138 and 501.

Code: 250 (Diabetes mellitus) Count: 13,804			
Subtask 2 **Final AUC: 0.97**		**Subtask 1 (CAML)** **Final AUC: 0.97**	
Word	Weight	Trigram	Weight
name	0.69	failure diabetes mellitus	0.29
mellitus	0.44	significant diabetes mellitus	0.19
coronary	0.19	diabetes mellitus treated	0.15
different	0.15	diabetes mellitus chronic	0.11
date	0.14	dictated bylast name	0.05
showed	0.13	hematocrit percent past	0.05
oral	0.08	fraction percent hypokinesis	0.04
number	0.07	md number1 dictated	0.02
diabetes	0.07	heart failure diabetes	0.01
white	0.07	percent hypokinesis right	0.01
Code: 138 (Polio, late effects) Count: 68			
Subtask 2 **Final AUC: 0.98**		**Subtask 1 (CAML)** **Final AUC: 0.74**	
Word	Weight	Trigram	Weight
polio	2.48	angioseal device deployed	0.15
puncture	0.33	anxiety per pt	0.10
proximal	0.22	postcath restarted oa	0.06
illness	0.18	qd gi cocktail	0.06
femoral	0.14	hypotension pt remained	0.06
constipation	0.12	hypotension postcath restarted	0.04
hospital	0.08	ptca comments french	0.03
surgery	0.08	stricture web although	0.03
continued	0.08	discharge date service	0.03
condition	0.08	protonix qd gi	0.02
Code: 501 (Asbestosis) Count: 115			
Subtask 2 **Final AUC: 0.96**		**Subtask 1 (CAML)** **Final AUC: 0.76**	
Word	Weight	Trigram	Weight
asbestosis	5.21	asbestos exposure hyperlipidemia	0.95
transferred	0.68	medical history cad	0.01
inhalation	0.53	discharge diagnosis cad	0.01
illness	0.45	history cad sp	0.004
condition	0.30	one day five	0.003
two	0.17	cad sp pci	0.002
breath	0.16	cad sp pci	0.002
family	0.10	disp50 tablets refills	0.002
shortness	0.08	date date birth	0.002
current	0.01	homecare discharge diagnosis	0.002

This can be explained as follows: since there are many notes in the training dataset to allow the CAML sub-network to achieve high accuracy, errors back-propagated into the similarity sub-network are small and the impact of code "250" on any of its weights is small. Next, we take a look at two less common diseases on which CAML sub-networks achieves smaller accuracy.

The second medical code shown in the table is "138" (polio, late effects). This code occurs only in 68 notes. AUC of the CAML sub-network is only 0.74. The similarity sub-network is very helpful and it results in increasing the AUC to 0.98. Among the top 10 trigrams of the CAML sub-network, we see that only "hypotension" is vaguely related to poliomyelitis. On the other hand, the similarity sub-network identifies highly important terms such as "polio", "puncture" and "constipation", which are all related to poliomyelitis problems. These words are mentioned both in Wikipedia document about "poliomyelitis" and in the clinical notes and are assigned large coefficients.

Table 6: Intersection of a clinical note and Wikipedia document for medical code "214" (Lipoma; count: 33) and final coefficient for each word.

After taking intersection (containing only 35 words):
known surgical present cm last first prior one medical family physical exam soft results blood disease possible identified multiple greater diagnosis condition internal years size tissue obese include removed wound common tumors minor lipomas
Final weights for words in intersection set:
obese: 1.42 lipomas: 0.70 tumors: 0.48 minor: 0.30 soft: 0.29 last: 0.24 wound: 0.22 tissue: 0.13 ... ...

Finally, we look at another rare code "501" (asbestosis). CAML sub-network recognizes important word "asbestos". The similarity sub-network identifies related words "asbestosis", "inhalation", "breath" and "shortness", which are all symptoms of asbestosis. The similarity sub-network improves the AUC for this disease from 0.76 to 0.96.

The strengths of the KSI framework in finding the important evidence can be explained by the way it works: the main obstacle to finding evidence in clinical notes for rare medical codes is that they do not have enough occurrences for models to select useful features from a large feature space. For example, an average note contains 1,596 words and CAML calculates an attention value for each word in the document. Thus, it is difficult for CAML to learn the attention values among more than a thousand words when medical codes are rare. On the other hand, external knowledge sources such as Wikipedia contain important words (signs, symptoms, typical treatments) for each disease, which can be used to narrow down the feature space. On average, each clinical note contains 407 unique words and each Wikipedia document contains 517 unique words. After taking their intersection, the resulting intersection set only contains an average of 64 words, which is much smaller than the original feature space. Then, the KSI framework learns weight for each word in the intersection set, which is much easier compared to learning weights from the original feature space. An example intersection set and final weights are shown in Table 6. Even for very rare labels such as "lipoma" the KSI framework is able to learn large weights for relevant words. We will also show the effectiveness of the intersection process in the next subsection.

5.3 Ablation study

To study contribution of each component in the KSI framework to learning the document similarity score, we compare our method to several variations of the framework. To focus on accuracy of different variations, in this subsection we fix the baseline model to be CAML and we study the effect of changing components of document similarity learning part on accuracy. We consider the following variations:

Table 7: The predictive performance of KSI and its variations.

Method	Macro AUC	Micro AUC	Macro F1	Micro F1	Test loss value	Top-10 recall
KSI-attention	0.871	0.978	0.255	0.654	0.032	0.807
KSI-intersection	0.864	0.979	0.263	**0.662**	0.032	0.809
KSI	**0.891**	**0.980**	**0.285**	0.659	0.032	**0.814**

KSI-attention. In order to study the effect of variable-level attention mechanism, we remove the attention vector in equation (5) and directly use e_i as input to the next layer.

KSI-intersection. In section 5.2 we analyzed the advantage of taking intersection of clinical note vector q and Wikipedia document vector k_i. In this variation we swap the order of element-wise multiplication and embedding. We first embed both q and k_i into low-dimensional vector space as

$$e_q = W_e q \tag{28}$$

$$e_{x_i} = W_e x_i, \tag{29}$$

where $e_q \in \mathbb{R}^m$ and $e_{x_i} \in \mathbb{R}^m$. Both e_q and e_{k_i} contain information about all words in the respective documents. Then we calculate element-wise multiplication of e_q and e_{x_i} as

$$e_i = e_q \odot e_{x_i}, \tag{30}$$

where e_i is used as in equations (4) and (5).

The results are shown in Table 7. From the table we can see that KSI outperforms other variations. Since the baseline model is fixed to CAML, which performs well on frequent labels, all accuracies are similar on micro-averaged metric. However, for macro-averaged metric, we can observe sizable differences. KSI significantly outperforms other methods on macro AUC and macro F1. KSI-intersection performs the worst on macro AUC, showing the importance of the intersection on inferring rare medical codes. KSI-attention also underperforms KSI by 2.0% and 3.0% on macro AUC and macro F1, respectively, showing the effectiveness of attention mechanism on selecting important features for prediction.

5.4 Qualitative insights from coefficients of KSI

Recent research about medical codes prediction [29, 35] observed improved performance by incorporating words in ICD-9 code descriptions into prediction model. These ICD-9 code descriptions are typically very short, with less than 10 words (e.g., “secondary diabetes mellitus with hyperosmolarity"). Nonetheless, it would be very promising to design a small knowledge base specialized for providing evidence for medical code prediction. To design such a knowledge base, it is important to understand which evidence is the most helpful for the task.

Besides serving as a framework that incorporates knowledge into other end-to-end models, the KSI could provide an insight into how Wikipedia helps clinical code prediction. As we showed by calculating coefficient λ, we are able to see which words in the intersection set are influencing the final prediction. We calculated the average coefficient of each word for a given Wikipedia document. Words with high average coefficients are influential and should be considered for future specialized knowledge base construction. We show words with the highest average coefficients for four Wikipedia documents in Table 8.

Table 8: Words with highest average coefficients for four disease Wikipedia documents.

Wiki documents	Words with highest coefficients
Episodic mood disorders	lithium bipolar affective dysregulation mdd manic anxiety psychiatrist hypomanic behavior
Lipoma	lipoma lipomatosis lipomas malignant tumor wound repeat tumors subcutaneous obese
Mild intellectual disabilities	epilepsy retardation service supervision abuse benzodiazepines psychiatry cerebral threatened
Kaposi’s sarcoma	sarcoma aids tumor haart kaposi ks antiretroviral chemotherapy meical hiv viral

As shown in the table, the influential words include representative symptoms for a disease such as “dysregulation", “anxiety", “manic" and “retardation". The full name and abbreviation of diseases are other types of highly informative words such as “lipoma", “ks" and “mdd". Besides, risk factors such as “hiv" and “obese" are also important evidence for inferring medical codes. These results shed light on the connection between the knowledge source and clinical notes, which provides guideline for future specialized knowledge-based design.

6 CONCLUSION

Medical code prediction from clinical notes is an active research area. The success of the task depends on finding evidence to infer diagnosis. However, it is challenging to learn what is the corresponding evidence in the notes when medical codes are rare. In this paper we propose a framework called KSI which is able to incorporate an external online knowledge source into any neural network multi-label predictive model. The key component of the KSI framework is to take intersection of a clinical note and external knowledge document, then use an attention mechanism to select important features in the intersection set. Finally, it calculates a matching score between the note and each knowledge document. KSI framework consistently improves the baseline model, especially when predicting rare codes. KSI framework also provides an insight into the evidence from an external source that was useful for the prediction. While our work uses medical codes which have Wikipedia web pages as labels, there is still a substantial number of medical codes without Wikipedia documentation. Our results suggest that it would be worthwhile for the Wikipedia community to create pages for all major ICD-9 and ICD-10 codes to not only enhance the health knowledge of public, but also to bring benefits to predictive modeling of clinical notes.

REFERENCES

[1] Anand Avati, Kenneth Jung, Stephanie Harman, Lance Downing, Andrew Ng, and Nigam H Shah. 2017. Improving palliative care with deep learning. In *Bioinformatics and Biomedicine (BIBM), 2017 IEEE International Conference on.* IEEE, 311–316.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[3] Tian Bai, Ashis Kumar Chanda, Brian L Egleston, and Slobodan Vucetic. 2017. Joint learning of representations of medical concepts and words from ehr data. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 764–769.

[4] Tian Bai, Ashis Kumar Chanda, Brian L Egleston, and Slobodan Vucetic. 2018. EHR phenotyping via jointly embedding medical concepts and words into a unified vector space. *BMC medical informatics and decision making* 18, 4 (2018), 123.

[5] Tian Bai, Shanshan Zhang, Brian L Egleston, and Slobodan Vucetic. 2018. Interpretable Representation Learning for Healthcare via Capturing Disease Progression through Time. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 43–51.

[6] Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noemie Elhadad. 2017. Multi-label classification of patient notes a case study on ICD code assignment. *arXiv preprint arXiv:1709.09587* (2017).

[7] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[8] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. 2016. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*. 301–318.

[9] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*. 3504–3512.

[10] Koby Crammer, Mark Dredze, Kuzman Ganchev, Partha Pratim Talukdar, and Steven Carroll. 2007. Automatic code assignment to medical text. In *Proceedings of the workshop on bionlp 2007: Biological, translational, and clinical language processing*. Association for Computational Linguistics, 129–136.

[11] Luciano RS de Lima, Alberto HF Laender, and Berthier A Ribeiro-Neto. 1998. A hierarchical approach to the automatic categorization of medical documents. In *Proceedings of the seventh international conference on Information and knowledge management*. ACM, 132–139.

[12] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science* 41, 6 (1990), 391–407.

[13] Richárd Farkas and György Szarvas. 2008. Automatic construction of rule-based ICD-9-CM coding systems. In *BMC bioinformatics*, Vol. 9. BioMed Central, S10.

[14] Jacob Goldberger, Geoffrey E Hinton, Sam T Roweis, and Ruslan R Salakhutdinov. 2005. Neighbourhood components analysis. In *Advances in neural information processing systems*. 513–520.

[15] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[16] Karla L Hoffman and Ted K Ralphs. 2013. Integer and combinatorial optimization. In *Encyclopedia of Operations Research and Management Science*. Springer, 771–783.

[17] Gao Huang, Chuan Guo, Matt J Kusner, Yu Sun, Fei Sha, and Kilian Q Weinberger. 2016. Supervised word mover's distance. In *Advances in Neural Information Processing Systems*. 4862–4870.

[18] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (2016), 160035.

[19] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).

[20] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[21] Matt Kusner, Yu Sun, Nicholas Kolkin, and Kilian Weinberger. 2015. From word embeddings to document distances. In *International Conference on Machine Learning*. 957–966.

[22] Leah S Larkey and W Bruce Croft. 1995. *Automatic assignment of icd9 codes to discharge summaries*. Technical Report. Technical report, University of Massachusetts at Amherst, Amherst, MA.

[23] Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*. 1188–1196.

[24] Nathan Levitan, A Dowlati, SC Remick, HI Tahsildar, LD Sivinski, R Beyth, and AA Rimm. 1999. Rates of initial and recurrent thromboembolic disease among patients with malignancy versus those without malignancy. *Risk analysis using Medicare claims data. Medicine (Baltimore)* 78, 5 (1999), 285–91.

[25] Lucian Vlad Lita, Shipeng Yu, Stefan Niculescu, and Jinbo Bi. 2008. Large scale diagnostic code classification for medical patient records. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.

[26] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* (2015).

[27] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1903–1911.

[28] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*. 3111–3119.

[29] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable Prediction of Medical Codes from Clinical Text. *arXiv preprint arXiv:1802.05695* (2018).

[30] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. 2013. Diagnosis code assignment: models and evaluation metrics. *Journal of the American Medical Informatics Association* 21, 2 (2013), 231–237.

[31] Adler J Perotte, Frank Wood, Noemie Elhadad, and Nicholas Bartlett. 2011. Hierarchically supervised latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*. 2609–2617.

[32] Aaditya Prakash, Siyuan Zhao, Sadid A Hasan, Vivek V Datla, Kathy Lee, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2017. Condensed Memory Networks for Clinical Diagnostic Inferencing.. In *AAAI*. 3274–3280.

[33] Alvin Rajkomar, Eyal Oren, Kai Chen, Andrew M Dai, Nissan Hajaj, Peter J Liu, Xiaobing Liu, Mimi Sun, Patrik Sundberg, Hector Yee, et al. 2018. Scalable and accurate deep learning for electronic health records. *arXiv preprint arXiv:1801.07860* (2018).

[34] Narges Razavian, Saul Blecker, Ann Marie Schmidt, Aaron Smith-McLallen, Somesh Nigam, and David Sontag. 2015. Population-level prediction of type 2 diabetes from claims data and analysis of risk factors. *Big Data* 3, 4 (2015), 277–287.

[35] Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P Xing. 2017. Towards Automated ICD Coding Using Deep Learning. *arXiv preprint arXiv:1711.04075* (2017).

[36] Donald H Taylor Jr, Truls Østbye, Kenneth M Langa, David Weir, and Brenda L Plassman. 2009. The accuracy of Medicare claims as an epidemiological tool: the case of dementia revisited. *Journal of Alzheimer's Disease* 17, 4 (2009), 807–815.

[37] Ankit Vani, Yacine Jernite, and David Sontag. 2017. Grounded Recurrent Neural Networks. *arXiv preprint arXiv:1705.08557* (2017).

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 6000–6010.

[39] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint Embedding of Words and Labels for Text Classification. *arXiv preprint arXiv:1805.04174* (2018).

[40] Wolfgang C Winkelmayer, Sebastian Schneeweiss, Helen Mogun, Amanda R Patrick, Jerry Avorn, and Daniel H Solomon. 2005. Identification of individuals with CKD from Medicare claims data: a validation study. *American Journal of Kidney Diseases* 46, 2 (2005), 225–232.

[41] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. 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*. 1480–1489.