IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 20, 2019, accepted October 7, 2019, date of publication October 11, 2019, date of current version October 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2946622

Regular Expression Based Medical Text Classification Using Constructive Heuristic Approach

MENGLIN CUI[1], RUIBIN BAI[1], ZHENG LU[1], XIANG LI[1], UWE AICKELIN[2], AND PEIMING GE[3]
[1]School of Computer Science, University of Nottingham, Ningbo 315100, China
[2]School of Computing and Information Systems, The University of Melbourne, Melbourne, VIC 3010, Australia
[3]Technology Department, Ping An Health Cloud, Shanghai 200030, China

Corresponding author: Ruibin Bai (ruibin.bai@nottingham.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 71471092 and Grant 61806129, in part by the Natural Science Foundation of Zhejiang Province under Grant LR17G010001, in part by the Ningbo Municipal Bureau of Science and Technology under Grant 2017D10034 and Grant 2014A35006, and in part by the China Postdoctoral Science Foundation under Grant 2018M640820 and Grant 2019T120751.

ABSTRACT Medical text classification assigns medical related text into different categories such as topics or disease types. Machine learning based techniques have been widely used to perform such tasks despite the obvious drawback in such ''black box'' approach, leaving no easy way to fine-tune the resultant model for better performance. We propose a novel constructive heuristic approach to generate a set of regular expressions that can be used as effective text classifiers. The main innovation of our approach is that we develop a novel regular expression based text classifier with both satisfactory classification performance and excellent interpretability. We evaluate our framework on real-world medical data provided by our collaborator, one of the largest online healthcare providers in the market, and observe the high performance and consistency of this approach. Experimental results show that the machine-generated regular expressions can be effectively used in conjunction with machine learning techniques to perform medical text classification tasks. The proposed methodology improves the performance of baseline methods (Naive Bayes and Support Vector Machines) by 9% in precision and 4.5% in recall. We also evaluate the performance of modified regular expressions by human experts and demonstrate the potential of practical applications using the proposed method.

INDEX TERMS Regular expressions, text classification, constructive heuristic method.

I. INTRODUCTION

Despite the popularity of Electronic Medical Record System, there are still a large amount of unstructured text data in medical domain. Classifying such data into useful categories such as topics or disease types by computer can significantly reduce human efforts and provide useful information for hospitals and medical services, especially the popular online healthcare services. However, classification in the medical domain is usually challenging because a great amount of domain knowledge is required to solve a even seemingly simple problem [1]. In the past decades, many techniques and algorithms have been developed for medical data mining and classification tasks. The prevailing approaches are machine learning algorithms such as Support Vector Machines (SVM) [2] and Latent Dirichlet Allocation (LDA) [3]. The application of Artificial Neural Network (ANN) to medical fields has rapidly gained popularity [4] after the first appearance in early 1990s [5]. While producing promising results, the models or solutions from these techniques are usually not interpretable by human. Human experts cannot directly fine-tune the models when the solution does not satisfy the high precision requirement of medical decision making.

Unlike dealing with other types of texts, medical text processing is unique in itself given the fact that reliability verification by domain experts is often required. Health experts are inclined to verify the evidence that supports decision making

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Afzal.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

and do not trust systems that act as black boxes. A regular expression based approach can be used to tackle such problem for its great interpretability. The regular expression, also known as regexp or regex, is a classical means for string pattern matching. Regular expressions have long been regarded as efficient tools in a wide range of application domains such as information extraction and text mining. While manually constructing regular expressions is obviously time-consuming, error-prone, and experience-dependent, there has been little work indicating that automatically generated regular expressions are able to give performances comparable to manual work. One of the main challenges in learning regular expressions by computer is the huge search space due to the large number of candidate words and their combinations through different operators. In addition, most of the previous work is designed without considering the capability of the model to be fine-tuned by human experts.

In online medical guidance, the narrative clinical texts written by patients often contain typos, misspellings, abbreviations, non-standard jargons, as well as incomplete sentences [6]. The oral expression of medical terms is therefore difficult to be processed by natural language processing (NLP) tools developed for ordinary text [7]. To address these issues, we investigate an automated regular expression generation method to classify medical texts in order to provide informative and comprehensive human-like medical guidance. Our framework is based on a constructive heuristic procedure carefully tailored to meet the specific demands for generating regular expression in the real-world application. The model has been proved to be capable of addressing medical text classification task with realistic data set.

In our opinion, medical text classification approaches should aim to achieve better performance (in terms of precision and recall, for example) and at the same time allow human experts to modify the solutions for even better results. In this research, we call a solution interpretable if and only if the solution can be comprehended and further enhanced by human. Our regular expression based system is transparent and interpretable for domain experts to make further modifications, whereas a system that is using sophisticated and not easy-to-understand machine learning techniques may require additional efforts to achieve this goal.

The main contributions of this work are summarized as follows:

1) A regular expression based text classifier that alleviates the ''black box'' problem which prevails in machine learning algorithms. The introduced regular expression structure makes it easy for human experts to understand and modify the solution for better results.

2) A novel constructive heuristic method that considers both the classification performance and the interpretability of regular expressions. The automated construction of regular expressions achieves the classification performance comparable to manual approach and reduces the labor and time costs. The proposed method is able to be used in conjunction with advanced machine learning methods for better overall performances.

3) A fully working system whose performance has been evaluated with massive amounts of real-world medical data. This work is an innovative attempt to produce interpretable medical decision support with much practical value.

The remainder of this paper is organized as follows. Section II discusses the related work with regards to clinical text classification and regular expressions implementations. The problem scenario is described in Section III. The proposed constructive heuristic method to generate regular expressions is presented in Section IV. In Section V, experimental results are provided to demonstrate the performance of the proposed approach. The paper is concluded in Section VI.

II. RELATED WORK

Text classification has been studied in recent years with the efforts focusing on learning based methods. Problems such as classifying cancer [8], patient record notes [9], intensive care unit (ICU) procedures and diagnosis [10], and other text documents [11] are solved by prevailing machine learning approaches with satisfactory results. With the development of deep learning techniques, many neural network models are designed for text classification tasks. Convolutional Neural Networks (CNNs) that learn high-level features have shown competitive results in sentence modeling [12]–[14], sentiment [15] and semantic [16] classification, email classification [17], online medical guidance [18], and other domains [19]–[21]. Recurrent Neural Networks (RNNs) also achieve state-of-the-art performances on a number of clinical data mining tasks [22]–[25]. Learning based methods often produce very promising results, but the drawback is also obvious because their solutions are not interpretable, leaving no easy way for human to fine-tune the solutions with their own domain knowledge.

Despite the promising performance that learning based methods show, regular expressions are often used in the situation where interpretable results are needed. Regular expressions are applied to perform several NLP tasks such as text classification, information extraction, and automatic summarizing. To reduce human efforts, a wealth of research efforts have been paid to automatically synthesize regular expressions for interpretable solutions [26], [27]. However, these researches often assume that the target regular expression is small and compact thereby allowing the learning algorithm to exploit the information efficiently. In addition, most of these works consider theoretical problems that are not inspired by any real-world applications [28] and the applicability of the corresponding methods is still largely unexplored. Attempts at learning regular expressions over real text were later introduced to detect Hyper Text Markup Language (HTML) lines [29] and spam emails [30], [31]. In medical domain, many regular expression based approaches have been adopted in various tasks including symptom classification [32] and extraction of medical information such as

blood pressure [33], ejection fraction [34], and bodyweight values [35] from clinical notes. Although producing promising results, these applications aim to deal with very specific and often simpler problems, and hence have trouble to cope with a much longer sequence of symbols from a much larger alphabet. In contrast, our study aims to present a more generalizable approach for automated regular expression learning to tackle text classification problems. The proposed framework inspired by real-world practice is not restricted by any aforementioned limitations in previous attempt. The sheer amount of data collected from online medical services dramatically increases the alphabet size and we exploit syntactic constructs (such as synonyms recognition and correlation analysis) accordingly to enable this scale-up.

Combining machine learning with explicit rules has been studied for decades and the rule-based solutions have demonstrated competitive performances [36], [37]. In general, two types of hybrid systems are developed to combine rule-based algorithms with machine learning techniques. One approach utilizes rules to verify the machine learning output [38], [39], while a more prevailing approach leverages on rule-based algorithms to precisely identify the desired features to feed machine learning models. In the medical domain, such methods have been explored to solve NLP tasks such as clinical text classification [40], [41], entity extraction [42], [43], and relation detection [44]–[47]. While the performance of machine learning, especially deep learning models is bound by the number and quality of annotated data, the rule-based feature extraction effectively exploits the available training data and gives a clear boost to machine learning models. In the recent literature, Zhang *et al.* [48] utilized simple regular expressions collected online and composed manually to produce weak labels for the entity mentions over a large document corpus, and a neural network was trained based on the regex-generated weak labels. Instead of massive labeling, human experts only need to label a small set of documents to fine-tune the neural network. Luo *et al.* [49] incorporated knowledge of regular expressions into the training of neural networks to solve typical spoken language understanding (SLU) tasks. Experiments demonstrate that the learning performance can be significantly improved by the implicit knowledge encoded within regular expressions. In clinical text classification, Wang *et al.* [40] presented a classification paradigm using weak supervision and deep representation to reduce human efforts. Weak supervision is achieved by a rule-based NLP algorithm to automatically generate labels, and then the pre-trained word embeddings are used as deep representation features for training machine learning models. Similarly, Yao *et al.* [41] proposed a clinical text classification method that combines rule-based features to identify trigger phrases and knowledge-guided CNN for disease classification. Experimental evaluations have validated the possibility of combining rule-based algorithms and machine learning techniques to achieve impressive accuracy while requiring modest human effort. The rules applied by these works are, however, relatively simple, restricted to specific domains, and ineffective for complex multi-classification tasks [40]. Deep learning techniques are used to leverage imperfect rules for higher accuracy. Our work remedies the limitation by developing compact and easily interpretable regular expressions which can be utilized as independent text classifiers with satisfying performance.

The contribution of our work is that we focus on automating the construction of a complete, informative, and well-generalized rule-based algorithm. The regular expression based classifier not only achieves promising results on its own, but also serves as a valid secondary verification to correct the instances misclassified by traditional machine learning and deep learning approaches.

III. SCENARIO

In this paper, we are concerned with generating regular expression based classifiers to solve text classification tasks with sample data, *i.e.*, strings annotated with their desired classes. The problem statement along with the notation used hereafter is thoroughly defined in this section.

A. PROBLEM DESCRIPTION

Formally the problem can be defined as follows: given a set of predefined classes C and a set of text inquires Q, the task is to classify each inquiry $q \in Q$ to one of the classes $c \in C$. The set of text inquiries belong to the same class c is denoted by Q_c. This is a typical text classification problem often being solved by supervised machine learning approaches.

Each solution to the problem (a regular expression based classifier for one class) is encoded as a vector of concatenated regular expressions

$$< R_1, R_2, \ldots, R_i, \ldots > . \tag{1}$$

To check whether a text inquiry belongs to a particular class, the regular expressions in the vector are matched sequentially in the same order of the vector with the text inquiry under consideration. The text inquiry is classified to the particular class if it is matched by any of the regular expressions in the classifier. Therefore, this task is treated as a binary classification.

B. REGULAR EXPRESSION DEFINITION

Each regular expression R_i is derived via a combination of functions and terminals defined in Table 1 and follows a global structure of two parts P_i and N_i concatenated by the `NOT` function denoted as #_#. That is, each regular expression R_i has the following format:

$$R_i = P_i \cdot (\#\_\#(N_i)), \tag{2}$$

where P_i tries to match all positive text inquiries and N_i is used to filter out the potential text inquiries wrongly matched by P_i. Note that under the circumstances when the positive part P_i alone is precise enough for correct classification (*i.e.*, the precision of P_i exceeds a predefined threshold), N_i is not needed. In this case, the structure of R_i is simplified to:

$$R_i = P_i. \tag{3}$$

TABLE 1. Functions and terminals used in the model.

Name	Label	Description
words	w	List of keywords extracted from the target text set using n-gram heuristic.
expression	e	An expression or term obtained through a combination of words and functions.
AND	$\cdot$	function to test logic `and` of two expressions.
OR	$\|$	function to test logic `or` of two expressions.
distance	$\{a, b\}$	function to test whether the distance between two words w_1 and w_2 are in the range $[a, b]$.
NOT	#_#	function to negate a given expression.

Let us define the expression e_i as a collection of words combined with `OR` function, which can be expressed as:

$$e_i = (w_{i1}|w_{i2}|\ldots|w_{in}), \tag{4}$$

where n is the number of words in the expression. The positive part P_i is either a single expression e_i^k consisting of keywords w_i^k, or a concatenation of e_i^k and e_i^r (a collection of related words w_i^r) with *distance* function. The negative part N_i is a single expression e_i^n consisting of negative words w_i^n. We will describe how to select e_i^k, e_i^r, and e_i^n in detail in Section IV-C. Formally, the two parts can be expressed as follows:

$$P_i = e_i^k \text{ or } e_i^k\{a, b\}e_i^r, \tag{5}$$

$$N_i = e_i^n. \tag{6}$$

C. GENERALITY OF A REGULAR EXPRESSION

Overfitting is one of the common issues in classification problems, where the classifier performs well with the training data but poorly with the testing data. In order to address this issue, we measure the distinctiveness of the words used in the expression compared with words present in negative samples. We define average word frequency f_c^w as the number of times a given word $w \in W$ present in all text inquiries in Q_c, divided by the total number of sentences of all text inquiries in Q_c. That is,

$$f_c^w = \frac{\sum\limits_{q \in Q_c} f_{c,q}^w}{\sum\limits_{q \in Q_c} |Q_c|}, \tag{7}$$

where $|Q_c|$ is the length (number of sentences) of Q_c.

In a larger sense, a generalizable classifier should be able to learn features directly from data and independent of seed patterns as explained in former parts. Therefore, our model is designed by employing a bottom-up constructive approach which starts from learning similarities and finding patterns rather than modifying seed patterns that are supplied by domain experts.

D. INTERPRETABILITY

In addition to the fitness measurement, we introduce the term `interpretability` of the solution to the problem in this paper. In our setting, the interpretability of a solution is highly domain specific or medical related. Specifically, the solution should contain keywords and demonstrate the relationship and interaction between keywords. With the help of domain experts, we develop the co-occurrence matrix and apply special operators to generate regular expressions R_i for better interpretability.

1) CO-OCCURRENCE MATRIX

A regular expression with good interpretability should be able to identify the hidden pattern of texts in a given class and the words in the expression should be related to each other. To achieve this, we build a co-occurrence matrix to suggest word correlations and determine the distance between phrases. Co-occurrence here is referred to as the frequency of two words occurring together in a certain order in every text inquiry of the input data. Apart from the frequency count, word distance information is also kept in the matrix. Let p be the size of vocabulary in the whole corpus. A matrix with a size of $p \times p \times 2$ will be produced. Specifically, we define the co-occurrence matrix $\boldsymbol{M}$ whose elements are calculated as follows:

$$\boldsymbol{M}(i, j, 1) = \sum_{q \in Q} \begin{cases} 1, & \text{if } pos_q(i) < pos_q(j) \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where i and j indicate the i-th and j-th word w_i and w_j respectively, $0 \le i, j \le p$, and $pos_q(*)$ represents the index (position) of a given word in text inquiry q.

Word distance is calculated by averaging the index difference, *i.e.* the value of $pos_q(j) - pos_q(i)$, over the inquires in which $pos_q(i) < pos_q(j)$. Explicitly, let Q' be the set of inquiries that contain both the i-th and j-th words and $pos_q(i) < pos_q(j)$, that is,

$$Q' = \{q \mid w_i, w_j \in q \text{ and } pos_q(i) < pos_q(j)\}. \tag{9}$$

The distance between the i-th and j-th word can then be counted as:

$$\boldsymbol{M}(i, j, 2) = \frac{\sum\limits_{q \in Q'} (pos_q(j) - pos_q(i))}{|Q'|}, \tag{10}$$

where $|Q'|$ is the number of inquires in Q'. The co-occurrence matrix $\boldsymbol{M}$ will be used in later stages to generate regular expressions.

2) SPECIAL OPERATORS

The use of some special regular expression operators during the generation process can lead to a shorter solution, hence increasing the interpretability. In particular, special

operators "?!" (zero-width negative look-ahead assertion) and "?<!" (zero-width negative look-behind assertion) are used for negation apart from N_i. Negative look-ahead and look-behind assertions are indispensable when we want to match something not followed or led by something else. For example, if we intend to match a c not followed by a b, negative look-ahead provides the solution: $c(?!b)$. Similarly, $(?<!b)c$ matches a c that is not preceded by a b using negative look-behind. These special negation operators are often used to eliminate undesired words or short phrases instead of long patterns, making the regular expression easier to read, especially when it comes to identifying complex patterns.

IV. REGULAR EXPRESSIONS GENERATION

In this section, we present the overall framework of the automatic generation of regular expressions using a constructive heuristic approach. Different from local search heuristics that improve a complete solution locally, a constructive heuristic method starts from an empty solution and then iteratively expands the current solution until a complete solution is constructed. Our method constructs a set of regular expressions directly from input data based on an iterative process and aims to find solutions with desired precision and recall as well as good interpretability.

In our setting, the input training data to our framework is a set of medical text inquires labeled with $|C|$ classes. The solution, a set of regular expressions, is constructed for every class, transforming the multi-class classification task into $|C|$ binary classification tasks. It is acknowledged that a key challenge in learning regular expressions is the huge search space of candidates since factors such as semantic similarity, word order, and distance between phrases should all be taken into consideration. To reduce the search space and build valid expressions, two methods have been introduced: the co-occurrence matrix is built to demonstrate the word correlations both grammatically and semantically; parallel labels C are clustered by their similarities to produce hierarchical labels.

For a given class c, the training data is divided into positive and negative sets based on labels. We then calculate the comparative frequency of each word in the two sets and select feature words accordingly. Regular expressions are generated based on the calculation of similarities and co-occurrence between words by predefined filtering mechanisms. The iterative process stops when predefined evaluation metrics are satisfied, *i.e.*, with fitness score above a certain threshold or no more additions to the existing solutions can be found. The proposed framework is illustrated in Fig. 1.

A. PRE-PROCESSING

During the preprocessing step, we apply the state-of-the-art Chinese word segmentation method *jieba* to every inquiry text in the data set. Duplicated strings are merged to prevent redundant processing. Parallel labels C are clustered to several subclasses according to their similarities. The set C can

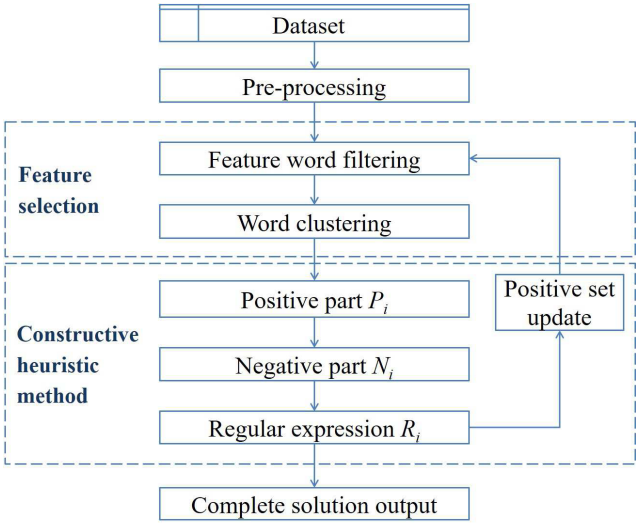

FIGURE 1. Diagrammatic demonstration of the regular expression generation process.

be expressed as a collection of subsets:

$$C = \{C_1, C_2, \ldots, C_i, \ldots\} \quad \text{with } C_i = \{c_i^1, c_i^2, \ldots, c_i^j, \ldots\}. \tag{11}$$

Given a class $c = c_i^j$, all text inquiries $q \in Q_c$ are treated as positive samples while the rest of the inquiries $q \in Q_{\bar{c}}$ are treated as negative samples, where $\bar{c}$ is defined as the complement of c_i^j given the complete set C_i. Stop words, symbols, punctuation, *etc*. are removed from the tokenized data to obtain the positive word set W_c and the negative word set $W_{\bar{c}}$.

B. FEATURE SELECTION

We term `feature words` as a set of words that are related to the class topic and contain domain knowledge. The selection of feature words is based on the relevance of a given word to the selected class. Empirically, the more often a word occurs in a class, the more relevant it is to the class; the more often a word occurs throughout all inquiries, the more poorly it discriminates between classes. Therefore, average word frequency f_c^w and $f_{\bar{c}}^w$ are calculated for every word w in W_c and $W_{\bar{c}}$. $\forall w \in W_c$, if $w \notin W_{\bar{c}}$ or $f_c^w / f_{\bar{c}}^w \geq \lambda_f$, where λ_f is a preliminarily defined threshold, w is regarded as a feature word and kept in W_c, otherwise it is removed from W_c. Word set $W_{\bar{c}}$ is filtered similarly. Keywords w^k and related words w^r are all chosen from W_c, while negative words w^n are chosen from $W_{\bar{c}}$.

Synonyms grouped together in the regular expression is a good indicator for interpretability. We measure the correlation among words is by their semantic similarity. Word embeddings assign such a low-dimensional vector representation to each word that semantically similar words are close to each other in the vector space [50]. The semantic correlation between two words is therefore quantified by

the cosine similarity measure between their corresponding vector representations. Word2vec can be trained over a large-scale unannotated corpus efficiently and encode meaningful linguistic relationships between words into learned word embeddings. We train our word2vec model on tens of millions of text records in the medical domain to produce effective word embedding. We search the embedding dimension from {50, 100, 200, 300, 400} and find 100 dimensional word embedding gives good performance while requiring modest computation time.

To improve the interpretability of regular expressions, we cluster feature words in every word set W to several groups G according to their similarity, where every group G contains synonyms or similar words. Similarity scores between words are measured by word2vec distributed representation. Word pairs with the similarity above a given threshold λ_s and with common words are clustered to one group G. Therefore, word set W_c and $W_{\bar{c}}$ can be expressed as:

$$W_c = \{G_1^c, G_2^c, \ldots, G_j^c, \ldots, G_{k_1}^c\}, \tag{12}$$

$$W_{\bar{c}} = \{G_1^{\bar{c}}, G_2^{\bar{c}}, \ldots, G_j^{\bar{c}}, \ldots, G_{k_2}^{\bar{c}}\}, \tag{13}$$

where k_1 and k_2 is the total number of groups in W_c and $W_{\bar{c}}$ respectively, and every word group G is expressed in the format defined in (4), which could be regarded as a valid regular expression. The maximum number of word groups in both W_c and $W_{\bar{c}}$ is restricted to 50 (*i.e.*, $k_1, k_2 \leq 50$) for more efficient calculation. Word groups that contain the least number of words are eliminated.

C. CONSTRUCTIVE HEURISTIC METHOD

Positive part P_i (line 2 to line 16 in Algorithm 1) As mentioned in (5), the positive part P_i is expressed in either of the two formats: e_i^k or $e_i^k\{a, b\}e_i^r$. The word group in W_c with the highest recall is selected as e_i^k to form the positive part P_i. Two parameters λ_{p1} and λ_{p2} are set to evaluate the precision of e_i^k, where $\lambda_{p1} > \lambda_{p2}$. The choice between the two formats depends on the precision of e_i^k with regards to the threshold λ_{p2}. We define the function $f_p(e)$ as the precision of expression e and $f_r(e)$ as the recall for clearer formulation.

- If the precision of e_i^k is higher than λ_{p2}, the expression e_i^k alone is considered to be specific enough to define topic-related knowledge for the given class, thus P_i is expressed as e_i^k.
- If the precision of e_i^k is not higher than λ_{p2}, another expression e_i^r is needed to further specify the pattern, thus P_i is expressed as $e_i^k\{a, b\}e_i^r$. The expression e_i^r is selected based on the co-occurrence matrix $\boldsymbol{M}$.

Distance control is realized by `distance` operator, which is denoted by $\{a, b\}$ in a regular expression. It restricts the number of tokens in between phrases, where a is the minimum token length required, and b is the maximum token length allowed. Compared with concatenating phrases with `and` operator which posts no distance restriction between phrases, regular expressions with distance control is less likely to match false positive snippets. In our implementation, a is set to zero, and the value of b is the average word distance calculated in the co-occurrence matrix $\boldsymbol{M}$ introduced in Section III-E.

Algorithm 1 Generation of a Single Regular Expression R_i

```
1:  procedure RegEx(W_c, W_c̄)
2:      calculate f_r(G_j^c) for every word group G_j^c in W_c
3:      sort W_c by f_r(G_j^c) in descending order
4:      for each G_j^c in W_c do
5:          if f_p(G_j^c) ≥ λ_p2 then
6:              P_i ← G_j^c
7:              break
8:          else
9:              select G_j̄^c in W_c with the highest
            co-occurrence rate with G_j^c
10:             compute average word distance b by
            co-occurrence matrix M
11:             P_i ← G_j^c{0, b}G_j̄^c
12:             if f_p(P_i) > λ_p2 then
13:                 break
14:             end if
15:         end if
16:     end for

17:     if f_p(P_i) ≥ λ_p1 then
18:         R_i ← P_i
19:     else
20:         set maxfit = 0
21:         for each G_j^c̄ in W_c̄ do
22:             R_candidate ← P_i #_# G_c̄
23:             if fitness score of R_candidate > maxfit then
24:                 maxfit ← fitness score of R_candidate
25:                 R_i ← R_candidate
26:             end if
27:         end for
28:     end if
29:     return R_i
30: end procedure
```

Negative part N_i (line 17 to line 28 in Algorithm 1) Negative part N_i is needed when P_i matches too many text inquires $q \notin Q_c$. In other words, when the precision of the positive part P_i is lower than the threshold λ_{p1}. Negative expression e_i^n is constructed by word groups from $W_{\bar{c}}$. We traverse all word groups in $W_{\bar{c}}$ and output the regular expression R_i with the highest fitness score.

Regular expression R_i and iteration (Algorithm 2) Regular expression R_i is generated in an iterative fashion. Whenever a new regular expression is added to the classifier, the positive set is updated by eliminating the instances matched by the regular expression in order for a quicker and more targeted generation of the next regular expression. The above-mentioned process starting from word selection is iterated to construct new regular expressions to match positive instances that are not matched by previous ones until

Algorithm 2 Iterative Process for the Complete Solution

Require: positive inquiry set Q_c, positive word list W_c, negative word list $W_{\bar{c}}$, recall threshold λ_r, and the maximum number of iterations n_{stop}
Ensure: a complete regex classifier C

```
set i = 0
repeat
    call procedure RegEx(W_c, W_c̄)
    add R_i to the classifier C
    update Q_c by deleting q ∈ Q_c matched by R_i
    update W_c according to new positive set Q_c
    i ← i + 1
until f_r(C) ≥ λ_r or i > n_stop
```

termination criteria are met, *i.e.*, the recall of current solution has already exceeded the threshold λ_r, or the number of iteration reaches n_{stop}.

Algorithm 1 presents the pseudo code of the generation of the regular expression R_i and Algorithm 2 demonstrates the iterative process for constructing the complete solution. Complexity analysis is specified to evaluate the efficiency of our algorithm. The time complexity of the algorithm to generate regular expression R_i is $O(m + n^2)$, where n is the vocabulary size of the dictionary and m is the total number of inquiries. Specifically, the calculation of precision, recall, and fitness of a single regular expression holds the complexity of $O(m)$. The selection of related part e_i^r in P_i and the calculation of distance between word clusters have the complexity of $O(n^2)$. In general, our algorithm is relatively efficient with large inputs, because the vocabulary size grows much slower as we increase the number of input text inquiries.

V. EXPERIMENTS

We carried out a comprehensive experimental evaluation to test the performance of the proposed regular expression based text classifier. Specifically, we try to address the interpretability of our proposed approach and the effectiveness of our method in improving the performance of existing machine learning algorithms. Additional experiments on a publicly available data set are conducted to evaluate the generalizability of our proposed method. We report the accuracy, precision, recall, macro- and micro-$F_{0.5}$ (as we focus on the precision parameter more than the recall) as our evaluation metrics.

A. DATA AND PARAMETER SETTINGS

We use online consultation data provided by our collaborator, a major online healthcare provider in the Chinese market, for both training and testing of our system. A collection of patient inquiries labeled by medical categories is treated as our input to perform the text classification task. The categories are manually labeled by a team of medical experts in our collaborating company. With the help of medical experts, we cluster these medical categories and give them more general labels (we call them clinical departments) to produce hierarchical labels. For example, common clinical departments may include pediatrics department, orthopedics department, gynecology department, *etc*. Within the gynecology department, medical categories such as vaginitis, menstrual disorder, and uterine myomas can be found. One can consider medical categories as a logical grouping of similar text inquiries and may not equal to the set of similar diseases as those in International Classification of Diseases Ninth Revision (ICD-9). Table 2 gives an intuition of what the data set looks like. Carefully going through the last three examples, we find the three inquiries with limited distinctions belong to totally different classes. Similar sentence structures and feature words in common would greatly obscure the boundary and confuse traditional classification models, whereas our regular expression based classifiers become comparably more effective because of their ability to precisely identify and exclude undesired patterns in the negative part of the regex solution.

The goal of our experiment is to classify each inquiry q to a class (or medical category in the context of our application). For a class c, positive instances used for constructing the positive part of a regex are inquiries that are labeled as c,

TABLE 2. Data set demonstration.

Text inquiries	Clinical Department	Medical Category
I feel difficult to fall asleep every day and I always dream during the night.	General medicine department	Insomnia
Always sleep talking. Feel stressed.	General medicine department	Insomnia
Will I get pneumonia if I occasionally get choked while eating?	General medicine department	Pneumonia
Baby has a fever and coughs. I'm worried if it's pneumonia.	Pediatrics department	Children cough
My son gets a cold and often sneezes.	Pediatrics department	Acute upper respiratory infection
5-year-old child burps out loud in the morning and it gets worse before sleep.	Pediatrics department	Children indigestion
Hiccup, uncomfortable throat.	Gastroenterology department	Adult indigestion
I feel full by eating just a little and could not digest properly.	Gastroenterology department	Adult indigestion
My right knee feels painful when I go upstairs. It doesn't hurt when I walk.	Orthopedics department	Knee pain
My husband smokes heavily. Does it matter if we want to have a child?	Gynecology department	Pregnancy preparation
Can I smoke after abortion surgery?	Gynecology department	Induced abortion
My period hasn't come this month. It sometimes comes late and the bleeding is scanty.	Gynecology department	Irregular menstruation
My period hasn't come this month, but I see a little blood on my underwear.	Gynecology department	Abnormal vaginal bleeding
I had sex last night and saw blood on my underwear this morning.	Gynecology department	Postcoital vaginal bleeding
...	...	...

while negative instances used for constructing the negative part are inquiries that belong to the same clinical department but different medical categories.

In our setting, 13 clinical departments with 776 medical categories are predefined by experts. A total of 4, 634, 742 effective records are collected from a 2-week online operational stream. The cumulative percentage of inquires with the number of categories is demonstrated in Fig. 2. Statistics reveal that the top 100 most common categories take up 80% of the total inquiries. We therefore choose to train regular expression based classifiers only for the top 100 categories which are within 7 major clinical departments in our experiment, because the quality of the regular expressions cannot be guaranteed with too little training data.

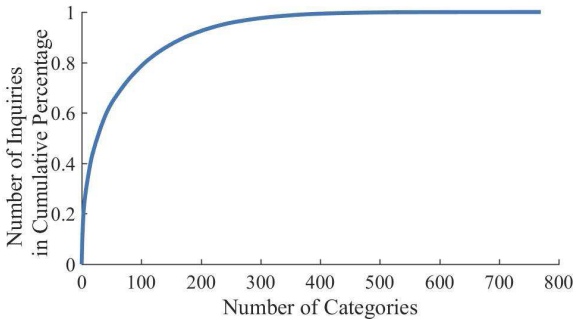

FIGURE 2. Number of categories and inquiries in cumulative percentage.

To evaluate the performance of our method under different sizes of training data, we utilize three training sets with different sizes. A validation set consisting of 100, 000 instances is utilized to set up the hyper-parameters, and a testing set of size 500, 000 is exclusively prepared. The distribution of records in the 2, 000, 000 training set categories is illustrated in Fig. 3. We tune the value for the parameters λ_f, λ_s, λ_{p1}, λ_{p2}, λ_r, n_{stop} on validation set after exploratory experimentation. The results were quite robust to the choice of hyper-parameters within specific ranges. According to our

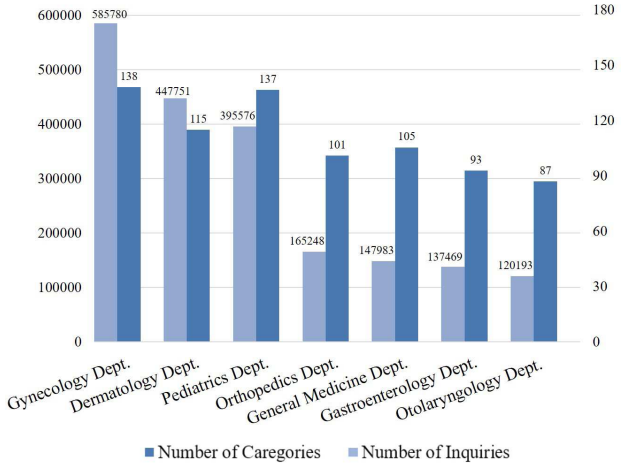

FIGURE 3. Data set categories and inquiries distribution.

experiments, we set $\lambda_f = 5$, $\lambda_s = 0.7$, $\lambda_{p1} = 0.8$, $\lambda_{p2} = 0.6$, $\lambda_r = 0.9$, and $n_{stop} = 10$ to get the best results.

B. REGULAR EXPRESSION BASED TEXT CLASSIFIER

Given a set of regular expressions, if the input text contains the pattern defined by any one of the regular expressions, the text is classified as a positive instance; otherwise, the input text is regarded as negative. The classification is performed on a class-by-class basis. On average, our method generates 7 unique regular expressions for each class. The solution is able to recognize 57% (recall) of the text inquiries on average and yields 89% precision using 2, 000, 000 training samples. We observe that patterns with higher fitness score from the training data tend to provide correct classification on the testing data as well. Therefore, the overfitting problem is effectively avoided by our proposed approach. We compare the performance of our method using different training sizes (see Table 3). Results demonstrate that more training data contributes to the better overall performance of the solution, as expected.

TABLE 3. Classification performance on different sample sizes.

Training size	200,000 (2:1:5)	500,000 (5:1:5)	2,000,000 (20:1:5) [a]
Precision	0.76 ± 0.10	0.83 ± 0.09	0.89 ± 0.07
Recall	0.45 ± 0.14	0.51 ± 0.11	0.57 ± 0.08

[a] Ratio of training, validation, and test set.

Note that the performance of the generated solution varies from category to category due to the different nature with various difficulties in finding feature words and synonyms, and discovering relations between phrases. For example, compared with 92% precision and 73% recall of category *diarrhea*, the algorithm only achieves the precision of 74% and recall of 28% for category *orthopedic pain*. When we carefully analyze the reason for this seemingly unstable performance, we find that symptoms for *orthopedic pain* can be very diverse, that is to say, feature words are hard to be extracted based on word frequency ratio. For instance, pain in body parts such as arm, leg, thigh, chest, finger, toe, *etc*. is generally covered by this category. This means text inquiries belong to *orthopedic pain* share less common features. The detailed information of precision and recall distribution of the selected 100 categories is illustrated in Fig. 4.

The current practice of our collaborator is to manually generate regular expressions by human experts for the classification task. This process is labor intensive and does not scale when the online data accumulate. Our method is able to automate the generation of regular expressions without human intervention even with large data sets. Figure 4 demonstrates that the precision of all solutions generated by our proposed approach lies in the range of [0.7, 1], while we observe four out of one hundred manually composed classifiers only yield a precision lower than 50%. This suggests that our

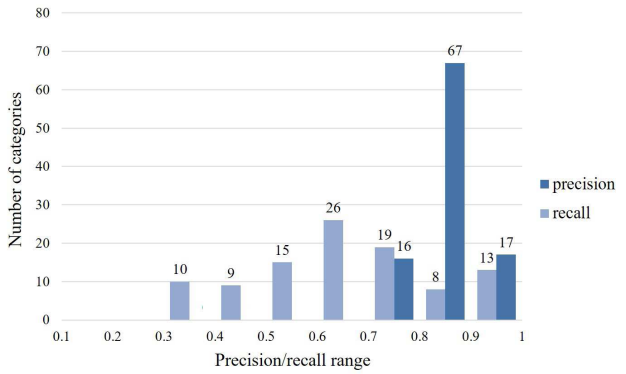

FIGURE 4. Precision and recall distribution of regex classifiers.

method is more robust and less overfitting than the manual approach. With such promising results, we still want to stress that the goal of our system is not to compete with machine learning methods or manual authorship of regular expressions and completely replace human efforts in the loop. In fact, the interpretability of our solution makes it a useful initial solution for human experts to modify. As a result, the overall labor and time required for human is significantly reduced.

C. MACHINE LEARNING + REGEX BASED CLASSIFIER

Note that our regular expression based text classification model is not proposed to compete with other classification approaches such as machine learning based approach. In fact, we design our method in this way so that both types of approaches can be used together for better performance. The regular expression based classifier is utilized to perform secondary verification on prediction results given by machine learning models. Experimental results demonstrate that we can achieve better performance by combining our method with baseline methods. Naive Bayes (NB), Support Vector Machines, and prevailing deep learning techniques for text classification such as Convolutional Neural Networks and Recurrent Neural Networks are utilized as baseline classifiers. In [18], a CNN model with named entity features achieved state-of-the-art performance in online medical guidance. We reproduced this model as one of our baseline methods. The baseline classifiers are trained on the same data set as we train the regular expression based classifiers. Due to the computational cost demanded by machine learning models, we perform the evaluation on the training data set with 500, 000 samples. Note that the absolute performances of the baseline models are not so important as we are interested in the improvement of combining the existing text classification models with our approach.

Regular expression based classifiers are combined with baselines by performing secondary verification on baseline model predictions. If the classification confidence is less than 0.6, we apply our regular expression based classifiers to the top 5 predictions with the highest confidence score sequentially. For example, for the inquiry "*My son is 7 years old. He has big trouble concentrating in class, and his hands and feet sometimes tremble while sleeping.*", the SVM model gives the top 5 predictions *hand tremble*, *Attention Deficit Hyperactivity Disorder (ADHD)*, *pediatric convulsions*, *sleep problems*, and *insomnia* with confidence scores 0.56, 0.47, 0.20, 0.05, 0.02 respectively. Regular expression classifiers of these 5 predictions are executed successively to verify the result until the inquiry is matched by a particular regular expression. For the example provided above, the inquiry text is matched by a regular expression within *ADHD* category. The predicted label of this instance is thus *ADHD* instead of *hand tremble*, by which means correcting the uncertain prediction originally given by SVM.

Furthermore, one prominent innovation of our proposal is that we develop regular expression based text classifiers that are fully interpretable to human and highly flexible for modification. Compared with manually composed classifiers which are usually labor intensive and time consuming, our approach is adequate to extract key features and identify text patterns within only a small amount of time. Human experts are able to enrich the solution with their prior experience, domain knowledge, and empirical facts to achieve human-machine collaborative intelligence. A list of examples is provided in Table 4 to illustrate how human modification is completed based on the regular expressions generated by our approach. Examples reveal that feature words and sentence structures can be precisely collected by our method. Medical experts may add uncommon synonyms, remove meaningless phrases, cluster similar expressions, and resolve overbroad solutions on this basis. Four medical experts collaboratively review and modify regular expressions which are generated on the same training set as we train the four baseline models.

Experimental results of baseline models, the combination of baselines and regular expressions, and the combination of baselines and human modified regular expressions are demonstrated in Table 5. The machine generated regular expressions improve the precision of Naive Bayes and SVM results by 9% and the recall by 4.5% on average. Macro and micro F-scores also reflect the multi-classification performance. The regex-based classifier narrows the gap between macro and micro $F_{0.5}$ given by NB and SVM models, indicating that the regular expressions elevate the performance of the classes with fewer samples, with which machine learning models do not perform well in general.

Since the performance of deep learning techniques outperforms traditional machine learning models by a great margin, directly combining the above-generated regex classifier with CNNs and RNNs does not markedly enhance the system performance. Nevertheless, the regex solution can be further enriched because of its full interpretability, and we introduce human-in-the-loop to amend the solution. The effectiveness of combining human modified regular expressions with DNNs is validated by experimental results shown in Table 5. A hybrid system which applies deep learning techniques as the foundation and regular expressions as secondary verification reaches a classification precision above 90% with

TABLE 4. Human modification of regular expressions for medical category "Diarrhea".

Regular expressions before human modification	**Precision: 92%**	**Recall: 73%**
.*(?<!(not|no)(diarrhea|diarrhoea|enterorrhea|scour|trot).{0,22}(yellow|watery|cramp)).*#_#.*(constipation|astriction|colonitis|colitis).*		
.*(antidiarrheal|lomotil|imodium|kaopectate).*#_#.*(vomit|dizzy|(stomach|belly).{0,7}(ache|pain)).*		
...		
Regular expressions after human modification	**Precision: 95%**	**Recall: 82%**
.*(?<!(not|no|like|want to)(diarrhea|diarrhoea|enterorrhea|dysentery|toilet.{0,2}[3456789] times)).*#_#.*(constipation|astriction|?<!(not|no|had)(colonitis|colitis|colonic ulcers|uicerative colitis)).*		
.*((scour|trot|flux|stool).{0,12}(yellow|watery|water-like|loose)|(yellow|watery|water-like|loose).{0,8}(scour|trot|flux|stool)).*#_#.*(constipation|astriction|?<!(not|no|had)(colonitis|colitis|colonic ulcers|uicerative colitis)).*		
.*(antidiarrheal|lomotil|imodium|kaopectate|loperamide).*#_#.*(vomit|dizzy|nausea|constipation|(stomach|belly).{0,3}(ache|pain|bulge|swell)).*		
...		

TABLE 5. Performances of Baseline, Baseline + Regex, and Baseline + Human Modified Regex methods.

	Accuracy	**Precision**	**Recall**	**Macro $F_{0.5}$**	**Micro $F_{0.5}$**
Naive Bayes					
Baseline Method	0.78 ± 0.08	0.71 ± 0.11	0.63 ± 0.15	0.66	0.73
Baseline + Regex	0.85 ± 0.05	0.81 ± 0.08	0.69 ± 0.12	0.75	0.79
Baseline + Modified Regex	0.89 ± 0.06	0.88 ± 0.07	0.77 ± 0.09	0.83	0.86
Support Vector Machines					
Baseline Method	0.83 ± 0.07	0.77 ± 0.10	0.75 ± 0.13	0.72	0.77
Baseline + Regex	0.86 ± 0.05	0.85 ± 0.07	0.78 ± 0.10	0.81	0.84
Baseline + Modified Regex	0.88 ± 0.05	0.89 ± 0.06	0.81 ± 0.09	0.86	0.87
Recurrent Neural Network					
Baseline Method	0.90 ± 0.03	0.88 ± 0.06	0.79 ± 0.08	0.85	0.88
Baseline + Regex	0.90 ± 0.03	0.89 ± 0.06	0.78 ± 0.08	0.85	0.89
Baseline + Modified Regex	0.93 ± 0.03	0.93 ± 0.05	0.85 ± 0.06	0.90	0.91
Convolutional Neural Network [18]					
Baseline Method	0.94 ± 0.03	0.91 ± 0.04	0.82 ± 0.05	0.88	0.90
Baseline + Regex	0.94 ± 0.03	0.91 ± 0.04	0.81 ± 0.05	0.88	0.90
Baseline + Modified Regex	0.95 ± 0.03	0.94 ± 0.04	0.87 ± 0.04	0.92	0.94

relatively high recall as well. The performances of traditional machine learning methods obtain even greater enhancement with the help of modified regular expressions. Therefore, we conclude that human-machine collaborative intelligence can be achieved with the help of interpretable regular expressions to tackle the "black box" issue in machine learning, especially deep learning models.

D. GENERALIZATION CAPABILITY

To demonstrate the generalizability of our method, we conduct additional experiments on a publicly available data set[1] released in [51] which was originally utilized to develop Chinese word segmentation tools. The data is collected from an online medical forum Good Doctor Online,[2] a Chinese forum for medical consultant. The number of instances in training, validation, and test sets are 4863, 1412, and 1474 respectively.

The annotation was done by the same team of medical experts following the same annotation rule in our previous experiment. Due to the limitation of data volume, the introduction of an excessive number of classes will cause training bias and inaccuracies, we perform the classification task based on the clinical department, a more general label than the medical category as mentioned in Section V-A. The number of instances in each class is provided in Fig. 5.

Four baseline classification models, namely NB, SVM, RNN, and CNN, are trained and combined with regex-based classifiers. Experimental results in Fig. 6 show that our hybrid system can obtain a 6.5% improvement in precision and a 5.6% improvement in recall on average. Further modifications by human experts are not conducted on this data set due to time and practical concern, but we are confident to anticipate that the performance of the system can be further enhanced if we introduce human-in-the-loop to fine-tune the

[1]The dataset has been kindly provided at https://github.com/adapt-sjtu/AMTTL/tree/master/medical_data

[2]http://www.haodf.com

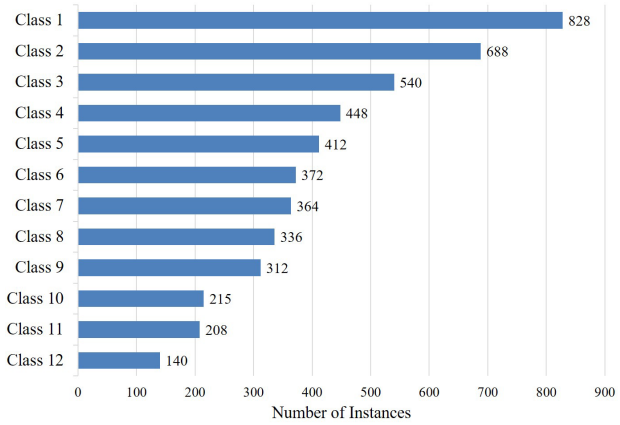

FIGURE 5. Data distribution of the forum data set.

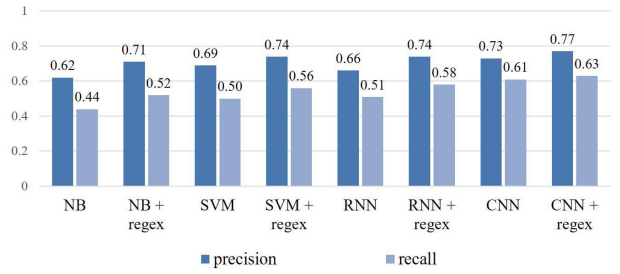

FIGURE 6. Performances of baseline and baseline + regex methods.

interpretable regular expressions with specialized domain knowledge.

VI. CONCLUSIONS AND EXTENSIONS

Regular expressions have long been used for text processing because of their expressiveness and flexibility. However, the generation of fully interpretable regular expressions is not trivial and often requires significant investment in manual work in qualified personnel. We have proposed a novel constructive heuristic algorithm towards constructing regular expression based classifiers for medical text classification tasks. The approach only requires a set of labeled examples and has no limitation on the size of the alphabet. Experimental results on real-world online medical inquiry demonstrate the high performance and consistency of this approach. Although there are many developed models for text or sentence classification, most suffer from ''black box'' problems and are incapable of future modification. Compared with conventional machine learning methods, the regular expression based classifiers not only further improve their performances by correctly recognizing many of the misclassified instances, but also avoid black boxes and instead generate solutions fully interpretable by humans. We believe that using regular expressions to tap into the sequential relationships among salient words is a promising approach to improve text classification performance and support accurate decision-making.

In the future, we will extend our framework to perform entity extraction tasks by inducing general patterns of medical entities in healthcare. Information extraction tasks may be addressed by regular expressions, because in many practical cases the relevant entities follow an underlying syntactical pattern that may be described by regular expressions [52]. With the extraction of high quality information and medical concepts as a basis, an automated process of knowledge graph construction will be achieved to learn high quality knowledge bases linking diseases and symptoms directly from electronic medical records.

REFERENCES

[1] A. Hall and G. Walton, ''Information overload within the health care system: A literature review,'' *Health Inf. Libraries J.*, vol. 21, no. 2, pp. 102–108, 2004.
[2] J. Thorsten, ''Learning to classify text using support vector machines: Methods, theory and algorithms,'' *Comput. Linguist*, vol. 29, no. 4, pp. 655–661, 2002.
[3] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, ''Learning to classify short and sparse text & Web with hidden topics from large-scale data collections,'' in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 91–100.
[4] P. J. G. Lisboa, ''A review of evidence of health benefit from artificial neural networks in medical intervention,'' *Neural Netw.*, vol. 15, no. 1, pp. 11–39, 2002.
[5] W. G. Baxt, ''Use of an artificial neural network for the diagnosis of myocardial infarction,'' *Ann. Internal Med.*, vol. 115, no. 11, pp. 843–848, 1991.
[6] H. Dalianis, *Clinical Text Mining: Secondary Use of Electronic Patient Records*. New York, NY, USA: Springer, 2018, pp. 1–4.
[7] K. Smith, B. Megyesi, S. Velupillai, and M. Kvist, ''Professional language in swedish clinical text: Linguistic characterization and comparative studies,'' *Nordic J. Linguistics*, vol. 37, no. 2, pp. 297–323, 2014.
[8] L. Wang, F. Chu, and W. Xie, ''Accurate cancer classification using expressions of very few genes,'' *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 4, no. 1, pp. 40–53, Jan. 2007.
[9] R. Cohen, I. Aviram, M. Elhadad, and N. Elhadad, ''Redundancy-aware topic modeling for patient record notes,'' *PloS One*, vol. 9, no. 2, 2014, Art. no. e87555.
[10] B. J. Marafino, J. M. Davies, N. S. Bardach, M. L. Dean, and R. A. Dudley, ''N-Gram support vector machines for scalable procedure and diagnosis classification, with applications to clinical free text data from the intensive care unit,'' *J. Amer. Med. Inform. Assoc.*, vol. 21, no. 5, pp. 871–875, 2014.
[11] A. Tripathy, A. Anand, and S. K. Rath, ''Document-level sentiment classification using hybrid machine learning approach,'' *Knowl. Inf. Syst.*, vol. 53, no. 3, pp. 805–831, 2017.
[12] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, ''A convolutional neural network for modelling sentences,'' 2014, *arXiv:1404.2188*. [Online]. Available: https://arxiv.org/abs/1404.2188
[13] A. Hassan and A. Mahmood, ''Convolutional recurrent deep learning model for sentence classification,'' *IEEE Access*, vol. 6, pp. 13949–13957, 2018.
[14] M. Hughes, I. Li, S. Kotoulas, and T. Suzumura, ''Medical text classification using convolutional neural networks,'' *Stud Health Technol. Inf.*, vol. 235, pp. 246–250, May 2017.
[15] J. Du, L. Gui, Y. He, R. Xu, and X. Wang, ''Convolution-based neural attention with applications to sentiment classification,'' *IEEE Access*, vol. 7, pp. 27983–27992, 2019.
[16] D. Zhang, L. Tian, M. Hong, F. Han, Y. Ren, and Y. Chen, ''Combining convolution neural network and bidirectional gated recurrent unit for sentence semantic classification,'' *IEEE Access*, vol. 6, pp. 73750–73759, 2018.
[17] G. Mujtaba, L. Shuib, R. G. Raj, N. Majeed, and M. A. Al-Garadi, ''Email classification research trends: Review and open issues,'' *IEEE Access*, vol. 5, pp. 9044–9064, 2017.
[18] C. Yao, Y. Qu, B. Jin, L. Guo, C. Li, W. Cui, and L. Feng, ''A convolutional neural network model for online medical guidance,'' *IEEE Access*, vol. 4, pp. 4094–4103, 2016.

[19] M. Cui and Y. Zhang, "Memristive synaptic circuits for deep convolutional neural networks," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.
[20] Y. Zhang, M. Cui, L. Shen, and Z. Zeng, "Memristive quantized neural networks: A novel approach to accelerate deep learning on-chip," *IEEE Trans. Cybern.*, to be published.
[21] Y. Zhang, M. Cui, Y. Liu, and L. Shen, "Hybrid CMOS-memristive convolutional computation for on-chip learning," *Neurocomputing*, vol. 355, pp. 48–56, Aug. 2019.
[22] Ö. Uzuner, B. R. South, S. Shen, and S. L. DuVall, "2010 i2b2/VA challenge on concepts, assertions, and relations in clinical text," *J. Amer. Med. Informat. Assoc.*, vol. 18, no. 5, pp. 552–556, 2011.
[23] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with LSTM recurrent neural networks," 2015, *arXiv:1511.03677*. [Online]. Available: https://arxiv.org/abs/1511.03677
[24] A. N. Jagannatha and H. Yu, "Structured prediction models for RNN based sequence labeling in clinical text," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, p. 856.
[25] A. N. Jagannatha and H. Yu, "Bidirectional RNN for medical event detection in electronic health records," in *Proc. Conf. Assoc. Comput. Linguistics North Amer. Meeting*, 2016, pp. 473–482.
[26] F. Denis, "Learning regular languages from simple positive examples," *Mach. Learn.*, vol. 44, nos. 1–2, pp. 37–66, 2001.
[27] H. Fernau, "Algorithms for learning regular expressions from positive data," *Inf. Comput.*, vol. 207, no. 4, pp. 521–541, 2009.
[28] O. Cicchello and S. C. Kremer, "Inducing grammars from sparse data sets: A survey of algorithms and results," *J. Mach. Learn. Res.*, vol. 4, pp. 603–632, Oct. 2003.
[29] E. Kinber, "Learning regular expressions from representative examples and membership queries," in *Proc. Int. Colloq. Grammatical Inference*. Berlin, Germany: Springer, 2010, pp. 94–108.
[30] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten, and I. Osipkov, "Spamming botnets: Signatures and characteristics," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 4, pp. 171–182, 2008.
[31] P. Prasse, C. Sawade, N. Landwehr, and T. Scheffer, "Learning to identify regular expressions that describe email campaigns," 2012, *arXiv:1206.4637*. [Online]. Available: https://arxiv.org/abs/1206.4637
[32] D. D. A. Bui and Q. Zeng-Treitler, "Learning regular expressions for clinical text classification," *J. Amer. Med. Inform. Assoc.*, vol. 21, no. 5, pp. 850–857, 2014.
[33] A. Turchin, N. S. Kolatkar, R. W. Grant, E. C. Makhni, M. L. Pendergrass, and J. S. Einbinder, "Using regular expressions to abstract blood pressure and treatment intensification information from the text of physician notes," *J. Amer. Med. Inform. Assoc.*, vol. 13, no. 6, pp. 691–695, 2006.
[34] J. H. Garvin, S. L. DuVall, B. R. South, B. E. Bray, D. Bolton, J. Heavirland, S. Pickard, P. Heidenreich, S. Shen, and C. Weir, "Automated extraction of ejection fraction for quality measurement using regular expressions in unstructured information management architecture (UIMA) for heart failure," *J. Amer. Med. Inform. Assoc.*, vol. 19, no. 5, pp. 859–866, 2012.
[35] M. A. Murtaugh, B. S. Gibson, D. Redd, and Q. Zeng-Treitler, "Regular expression-based learning to extract bodyweight values from clinical notes," *J. Biomed. Inform.*, vol. 54, pp. 186–190, Apr. 2015.
[36] P. Langley and H. A. Simon, "Applications of machine learning and rule induction," *Commun. ACM*, vol. 38, no. 11, pp. 54–64, 1995.
[37] S. M. Weiss and N. Indurkhya, "Rule-based machine learning methods for functional prediction," *J. Artif. Intell. Res.*, vol. 3, pp. 383–403, Dec. 1995.
[38] J. Y. Lee, F. Dernoncourt, and P. Szolovits, "MIT at SemEval-2017 task 10: Relation extraction with convolutional neural networks," 2017, *arXiv:1704.01523*. [Online]. Available: https://arxiv.org/abs/1704.01523
[39] C. Li, Z. Rao, Q. Zheng, and X. Zhang, "A set of domain rules and a deep network for protein coreference resolution," *Database*, vol. 2018, pp. 1–10, Jan. 2018.
[40] Y. Wang, S. Sohn, S. Liu, F. Shen, L. Wang, E. J. Atkinson, S. Amin, and H. Liu, "A clinical text classification paradigm using weak supervision and deep representation," *BMC Med. Inform. Decis. Making*, vol. 19, no. 1, 2019, Art. no. 1.
[41] L. Yao, C. Mao, and Y. Luo, "Clinical text classification with rule-based features and knowledge-guided convolutional neural networks," *BMC Med. Informat. Decis. Making*, vol. 19, no. 3, 2019, Art. no. 71.
[42] G. K. Savova, J. J. Masanz, P. V. Ogren, J. Zheng, S. Sohn, K. C. Kipper-Schuler, and C. G. Chute, "Mayo clinical text analysis and knowledge extraction system (cTAKES): Architecture, component evaluation and applications," *J. Amer. Med. Informat. Assoc.*, vol. 17, no. 5, pp. 507–513, Sep. 2010.
[43] Y. Xu, K. Hong, J. Tsujii, and E. I.-C. Chang, "Feature engineering combined with machine learning and rule-based methods for structured information extraction from narrative clinical discharge summaries," *J. Amer. Med. Informat. Assoc.*, vol. 19, no. 5, pp. 824–832, 2012.
[44] Y.-C. Chang, H.-J. Dai, J. C.-Y. Wu, J.-M. Chen, R. T.-H. Tsai, and W.-L. Hsu, "TEMPTING system: A hybrid method of rule and machine learning for temporal relation extraction in patient discharge summaries," *J. Biomed. Informat.*, vol. 46, pp. S54–S62, Dec. 2013.
[45] A. Kovačević, A. Dehghan, M. Filannino, J. A. Keane, and G. Nenadic, "Combining rules and machine learning for extraction of temporal expressions and events from clinical narratives," *J. Amer. Med. Informat. Assoc.*, vol. 20, no. 5, pp. 859–866, 2013.
[46] Y.-L. Yang, P.-T. Lai, and R. T.-H. Tsai, "A hybrid system for temporal relation extraction from discharge summaries," in *Proc. Int. Conf. Technol. Appl. Artif. Intell.* Cham, Switzerland: Springer, 2014, pp. 379–386.
[47] A. Bravo, T. S. Li, A. I. Su, B. M. Good, and L. I. Furlong, "Combining machine learning, crowdsourcing and expert knowledge to detect schemical-induced diseases in text," *Database*, vol. 2016, pp. 1–11, Jun. 2016.
[48] S. Zhang, L. He, S. Vucetic, and E. Dragut, "Regular expression guided entity mention mining from noisy Web data," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 1991–2000.
[49] B. Luo, Y. Feng, Z. Wang, S. Huang, R. Yan, and D. Zhao, "Marrying up regular expressions with neural networks: A case study for spoken language understanding," 2018, *arXiv:1805.05588*. [Online]. Available: https://arxiv.org/abs/1805.05588
[50] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2177–2185.
[51] J. Xing, K. Zhu, and S. Zhang, "Adaptive multi-task transfer learning for Chinese word segmentation in medical text," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 3619–3630.
[52] A. Bartoli, A. De Lorenzo, E. Medvet, and F. Tarlao, "Regex-based entity extraction with active learning and genetic programming," *ACM SIGAPP Appl. Comput. Rev.*, vol. 16, no. 2, pp. 7–15, 2016.

MENGLIN CUI received the B.S. degree in economics from the Beijing Institute of Technology, Beijing, China, in 2016, and the M.S. degree in business analytics from the University of Rochester, Rochester, NY, USA, in 2017. She is currently pursuing the Ph.D. degree with the School of Computer Science, University of Nottingham, U.K. From 2014 to 2015, she was an Exchange Student with the University of Bayreuth, Germany. Her current research interests include natural language processing, data mining, and deep learning.

RUIBIN BAI received the B.Sc. and M.Sc. degrees from Northwestern Polytechnic University, China, and the Ph.D. degree from the University of Nottingham, U.K. He is currently a Professor with the School of Computer Science, University of Nottingham, Ningbo, China, and leads the Artificial Intelligence and Optimization (AIOP) Group. His current research interests include computational intelligence, reinforcement learning, operations research, modeling, scheduling and optimization with a special focus on transportation systems, and digital healthcare. He is an Associate Editor of *Networks* and an ISI indexed journal.

ZHENG LU received the B.S. and Ph.D. degrees in computer science from the National University of Singapore, in 2004 and 2011, respectively. He was with the Microsoft Research Asia, from 2009 to 2010. He was a Postdoctoral Research Fellow with The University of Texas at Austin, in 2012. He is currently an Assistant Professor with the University of Nottingham, Ningbo, China. His current research interests include computer vision, image processing, and machine learning.

XIANG LI received the B.S. degree in automotive engineering from the Wuhan University of Science and Technology, Wuhan, China, in 2010, the M.S. degree in mechanical engineering from Loughborough University, U.K., in 2012, and the Ph.D. degree in computer science from the University of Nottingham, U.K., in 2017. He is currently a Research Fellow with the School of Computer Science, University of Nottingham, Ningbo, China. His current research interests include data mining, machine learning, natural language processing, and recommender system.

UWE AICKELIN received the Ph.D. degree from the University of Wales, U.K. He is currently a Professor and the Head of the School of Computing and Information Systems, The University of Melbourne. His current research interests include artificial intelligence (modeling and simulation), data mining and machine learning (robustness and uncertainty), decision support and optimization (medicine and digital economy), and health informatics (electronic healthcare records). He is an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION.

PEIMING GE received the B.S. and Ph.D. degrees in computational intelligence from Southwest Jiaotong University, Chengdu, China, in 2001 and 2006, respectively. He was a Postdoctoral Research Fellow with Tongji University, Shanghai, China, from 2006 to 2008. He is currently with the Ping An Health Cloud and leading the artificial intelligence Team. His current research interests include machine learning, data visualization, and knowledge graph.