IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 5, 2020, accepted January 23, 2020, date of publication February 6, 2020, date of current version February 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2972205

CREGEX: A Biomedical Text Classifier Based on Automatically Generated Regular Expressions

CHRISTOPHER A. FLORES[1], ROSA L. FIGUEROA[1], JORGE E. PEZOA[1], (Member, IEEE), AND QING ZENG-TREITLER[2]
[1]Electrical Engineering Department, Universidad de Concepción, Concepción 4070409, Chile
[2]Biomedical Informatics Center, George Washington University, Washington, DC 20037, USA

Corresponding author: Rosa L. Figueroa (rosa.figueroa@biomedica.udec.cl)

This work was supported in part by the CONICYT-PFCHA/Doctorado Nacional under Grant 2017-21172062 Scholarship, in part by the UCO 1866 Project, in part by the Universidad de Concepción VRID-Enlace Project under Grant 217.092.052-1.0, in part by the FONDEF IDeA I+D under Grant ID19I10120, and in part by the CORFO Project under Grant 16CTTS66390.

ABSTRACT High accuracy text classifiers are used nowadays in organizing large amounts of biomedical information and supporting clinical decision-making processes. In medical informatics, regular expression-based classifiers have emerged as an alternative to traditional, discriminative classification algorithms due to their ability to model sequential patterns. This article presents CREGEX (Classifier Regular Expression), a biomedical text classifier based on an automatically generated regular-expressions-based feature space. We conceived an algorithm for automatically constructing an informative and discriminative regular-expressions-based feature space, suitable for binary and multiclass discrimination problems. Regular expressions are automatically generated from training texts using a coarse-to-fine text aligning method, which trades off the lexical variants of words, in terms of gender and grammatical number, and the generation of a feature space containing a large number of noisy features. CREGEX carries out feature selection by filtering keywords and also computes a confidence metric to classify test texts. Three de-identified datasets in Spanish, with information on smoking habits, obesity, and obesity types, were used here to assess the performance of CREGEX. For comparison, Support Vector Machine (SVM) and Naïve Bayes (NB) supervised classifiers were also trained with consecutive sequences of tokens (n-grams) as features. Results show that, in all the datasets used for evaluation, CREGEX not only outperformed both the SVM and NB classifiers in terms of accuracy and F-measure (p-value<0.05) but also used a fewer amount of training examples to achieve the same performance. Such a superior performance is attributed to the regular expressions' ability to represent complex text patterns.

INDEX TERMS Biomedical informatics, regular expressions, sequence alignment, text classification.

I. INTRODUCTION

Continuous technological progress has made it possible to generate a large amount of information in digital formats. It is estimated that by 2025 there will be 175 zettabytes of digital information, much of it presented in unstructured form or free text [1], [2]. In health care, this growing rate of accumulation of digital information is reflected in the electronic medical record data, necessitating the development of technologies that make it possible to organize and discover relevant knowledge automatically from such sources in support of decision making [3]. One of the most widely used techniques to organize a large amount of digital information is text classification [4].

Text classification, or categorization, is a supervised learning method that automatically assigns labels to free texts based on their content [5]. By employing supervised learning algorithms trained on labeled texts, a classification model is created to predict the labels of unlabeled texts [6].

The NB and the SVMs are two of the most commonly used automatic learning algorithms for text classification because of their simplicity and accurate label predictions [7]. Although these algorithms often perform well, there is room for improvement. Researchers have looked into regular expressions as an alternative and have achieved comparable performance to traditional approaches [8], [9]. Algorithm that requires regular expressions to be manually created by domain experts is less desirable. Thus, automatic generation of regular expressions is of particularly interest to researchers [10].

The associate editor coordinating the review of this manuscript and approving it for publication was Victor Hugo Albuquerque.

This paper aims to address the following research questions:

- For a given biomedical text classification problem, can an algorithm based on regular expressions model the lexical variants of the representative terms of each class of the problem?
- For a given biomedical text classification problem, can an algorithm based on regular expressions outperform traditional classification methods?

The first contribution of this paper is an algorithm for automatically constructing a feature space from biomedical texts. The feature space is composed of a set of regular expressions, which are automatically generated from labeled training texts. Our essential idea relies on employing a coarse-to-fine text aligning scheme for generating informative and discriminating regular expression features. Thus, the set of automatically generated regular expressions trades off the lexical variants of words, in terms of gender and grammatical number, and the dimension of the feature space, in terms of generating a proper number of relevant features and a reduced number of noisy features. We remark that the feature space construction method can be used in both binary and multiclass discrimination problems.

The second contribution of this paper is a biomedical text classifier, which we have termed as CREGEX (Classifier Regular Expression). CREGEX exploits the structure provided by the regular-expression–based feature space to define a simple decision function that uses the labels associated with the regular expressions and, in case of ambiguity, uses as additional discrimination information local similarity scores previously computed during the text alignment step. Three biomedical datasets in the Spanish language were used to evaluate the effectiveness of CREGEX for text classification. CREGEX's performance was compared to the performance achieved by SVM and NB classifiers. Features used to build the SVM, and NB classifiers were n-grams and represented using the Bag of Words (BoW) model. Results indicate that, in all cases, CREGEX outperformed both SVM and NB in terms of the accuracy (ACC) and F-measure ($p\text{-value}<0.05$) metrics. Also, CREGEX used fewer training examples than the SVM and NB classifiers to achieve the same performance.

The rest of this paper is organized as follows: Section II presents a review of the research body related to the automatic generation of regular expressions and some applications in text mining. Section III describes the biomedical texts used in this paper and explains how CREGEX works. Section IV presents the performance results of CREGEX in terms of ACC, F-value (F1), learning curves, and classification errors. Section V presents an analysis of the results achieved by CREGEX and outlines the future work to be carried out by our group.

II. RELATED WORK

Regular expressions are defined as a sequence of characters used in programming that make it possible to define search patterns in the texts [10], [11]. The simplicity of creating regular expressions manually in different domains has led to the extensive use of this method in the validation of forms, information extraction, spam detection, tokenization, and negation detection in texts, among others [11], [12]. However, in biomedical text classification there is limited use of regular expressions, most of which focusing on information extraction tasks [13]–[17].

On the other hand, the automatic generation of regular expressions from examples is a current research topic [18]. One type of methods generate regular expressions by transforming an input regular expression to improve its performance in a specific task [19]–[23]. Therefore, the performance of these methods is influenced by the initial example. Some of the current methods rely on domain experts to provide a good initial example. For example, Li et al. proposed a method called ReLIE for information extraction tasks [19]. ReLIE performs multiple transformations to an input regular expression using metacharacters (e.g., lookahead operators, quantifiers, and disjunctions) until the performance is optimized. The results indicate that ReLIE performed better than the Conditional Random Field (CRF) method and can also improve its performance by training with features extracted by ReLIE.

Some other methods do not require an initial regular expression but require training examples with the text segments of interest to be labeled [15], [24], [25]. For example, Murtaugh et al. propose a method called REDEx to extract numerical values from biomedical texts [15]. REDEx builds a pattern for the target value by converting the preceding and succeeding text segments into regular expressions (e.g., replacing punctuation, digits, and whitespaces). Then these patterns are progressively added to the target value until false positives are obtained in the training set. The REDEx obtained a performance of over 98% in terms of the ACC and F-measure metrics.

Yet another group of methods employed genetic programming or dynamic programming, differing from the previous work on both the evaluation of regular expressions in the training set and the extraction of common tokens from the texts [8], [26]–[28]. For instance, Bartoli et al. used genetic programming to generate a population of regular expressions from an input-labeled example [27]. These regular expressions are iteratively modified using genetic operators such as mutation and cross-over until a maximum number of generations, or a maximum performance has been achieved according to a fitness function that considers the size of the regular expression generated and a measure of distance (Levenshtein), with respect to the labeled segment of interest. Bui and Zeng-Treitler proposed a biomedical text classifier called RED, which uses dynamic programming to generate regular expressions automatically [8]. RED generates regular expressions by combining sequences of tokens (phrases), which are obtained after a process of local text alignment. To do so, RED uses the Smith-Waterman (SW) algorithm and metacharacters for controlling the number of tokens that can be inserted between the phrases (whitespaces, whitespace negations, and quantifiers) and the normalization of numbers. Regular expressions are then filtered by a performance threshold, measured in terms of precision, and used to classify texts. Results indicate that RED achieved over 80% of classification accuracy and F-measure, which are superior to the performance of a SVM.

As in [8], the CREGEX algorithm automatically extracts tokens from biomedical texts to form regular expressions; however, the main differences between CREGEX and such work are the following. First, before the extraction of tokens, CREGEX added a pre-processing stage for the biomedical texts using hierarchical clustering and the Needleman-Wunsch (NW) algorithm to represent common groups of words through a common pattern. In this stage, CREGEX also replaces numbers with patterns that represent numerical intervals. Secondly, CREGEX filters regular expressions using keywords according to the classification problem domain of knowledge. Finally, CREGEX uses different strategies to classify the test texts according to the number of regular expressions that match in them.

III. MATERIALS AND METHODS

A. DATASETS DESCRIPTION AND PREPROCESSING

To carry out this study, the Guillermo Grant Benavente Hospital (HGGB) in Concepción provided three de-identified datasets for this study. These datasets have been authorized to be used here by the ethics committee of this health care establishment. Besides, datasets have also been used in previous research works. These datasets were retrieved from the HGGB's electronic medical record system and then manually annotated by a group of Biomedical Engineering students from the Universidad de Concepción, obtaining an almost perfect agreement between annotators in all datasets [29]–[31]. The first dataset, called "SMOKING STATUS," contains labeled texts with the classes "SMOKE" and "DOES NOT SMOKE." The second dataset, called "OBESITY STATUS," contains texts with the classes "OBESE" and "NON OBESE." Finally, the third dataset, "OBESITY TYPES," contains multi-class information on the obesity types: "MODERATE OBESITY," "SEVERE OBESITY," "MORBID OBESITY," and "SUPER MORBID OBESITY." A brief description of the datasets is shown in Table 1.

The biomedical texts of each dataset were pre-processed to facilitate the implementation of all classifiers. Each text was converted to lower-case and then tokenized, considering whitespaces as token boundaries (words, numbers, and symbols). Whitespaces were added between non-alphanumeric characters to extract more fine-grained tokens. As a result, it is possible to extract the tokens "asthmatic", "patient", "(", "ex", "smoker" and ")" from the text "asthmatic patient (ex smoker)".

TABLE 1. Description of the datasets. The keywords "imc," "peso," and "sobrepeso" mean "body mass index (bmi)," "weight," and "óverweight" in Spanish, respectively. The keywords "obes*," "tab*," "fum*," "cig*," and "caj*" are the roots of the keywords "obesity," "tobacco," "smoker," "cigarette," and "cigarette box" in Spanish, respectively.

Dataset	Keywords	Number of classes	Number of examples	Kappa index
OBESITY STATUS	obes*, imc, peso, sobrepeso	2	1111	0.98^a
OBESITY TYPES	obes*, imc, peso, sobrepeso	4	851	0.97^a
SMOKING STATUS	tab*, fum*, cig*, caj*	2	1027	0.86^b

[a] Cohen's Kappa index. [b] Fleiss' Kappa index.
*Keyword root.

B. ALGORITHM FOR AUTOMATICALLY CONSTRUCTING A FEATURE SPACE

1) DEFINITIONS

Regular expressions are defined as a sequence of characters defining search patterns in texts [10], [11]. For example, the regular expression "Obesity" contains seven literal characters and will match any text that contains the word "Obesity" starting with a capital letter. In contrast, the regular expression "^Obesity" contains the metacharacter "^" that will match any text containing the word "obesity" starting with "O" at the beginning of the text. A regular expression, r, can be defined mathematically as a set of strings over a finite alphabet.

Consider now the set $X = \{x_1, x_2, \cdots x_n\}$, which represents n biomedical texts. Consider also that every text is labeled with one and only one class, out of m possible options. Let us denote as y_i the class of the ith text, with $y_i \in Y = [1, m]$, $m \geq 2$. Consider next the collection R that represents all the regular expressions that can be generated from the biomedical texts in X. We define R as the regular-expressions–based feature space for the classification problem induced by X and Y. Note that the classification problem can be either binary or multiclass.

The proposed algorithm for automatically constructing a feature space introduces the mapping $\Phi(x_i, y_i) : (X, Y) \rightarrow R_i \subseteq R$ that automatically generates a collection of n_i regular expressions, $R_i = (r_1^i(y_i), \ldots, r_{n_i}^i(y_i))$, associated with the ith training text and the class label y_i. Thus, after applying the map to the entire sets X and Y, our algorithm automatically generates the collection $R = \cup_{i=1}^n R_i$ of regular expressions, for each one of the m problem classes. The mapping $\Phi(\cdot, \cdot)$ is defined in terms of two text aligning algorithms: the global NW algorithm, which aligns groups of words in a text at a coarse-grain level, and the local SW algorithm, which aligns texts at a fine-grain level. Details are provided in the upcoming sections.

Figure 1 shows a functional scheme containing all stages of the proposed algorithm for automatically constructing the feature space. First, the pre-processed input texts are aligned globally and locally to extract common patterns. Second, the extracted patterns are used to automatically generate

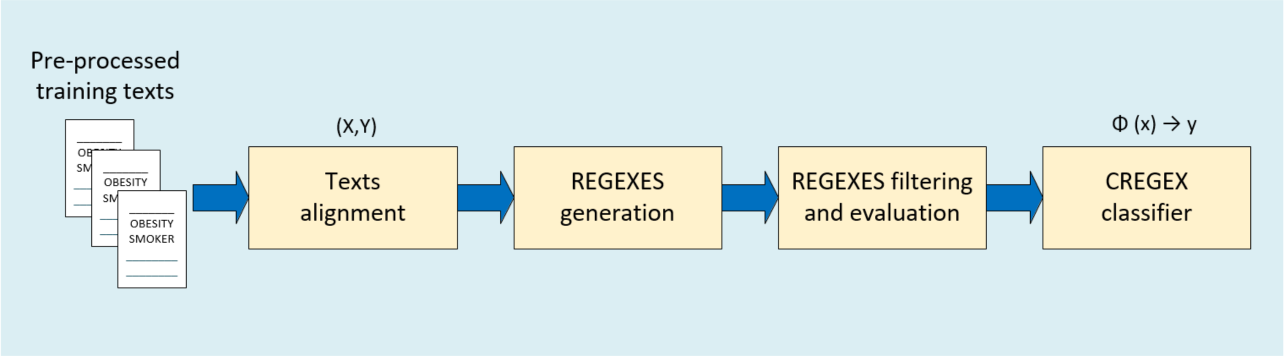

FIGURE 1. General functional scheme of our algorithm for automatically constructing a regular-expressions–based feature space and its relationship with the CREGEX biomedical text classifier.

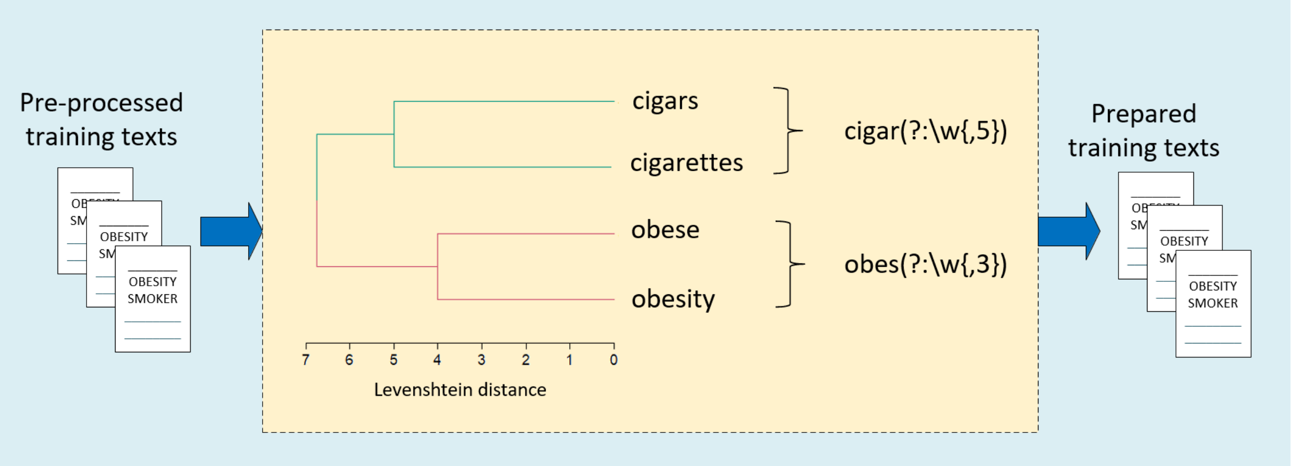

FIGURE 2. Hierarchical clustering and the NW algorithm to find common patterns among words.

regular expressions (REGEXES). The generated REGEXES are then filtered by keywords and evaluated to measure their performance in the training set. Finally, the CREGEX classifier assigns a y_i class to a x_i test text by using all the regular expressions generated in the previous stages.

2) COARSE-GRAIN TEXT ALIGNING: THE GLOBAL NW algorithm

To facilitate the extraction of tokens for the generation of regular expressions, our algorithm groups the similar words of the training set and replaces them with a common pattern. Thus, it attempts to capture the lexical variants of words in terms of gender and grammatical number, including typos. Similar words are grouped by hierarchical clustering using the metric Levenshtein distance [32]. A cut-off point equal to four was used to form the groups from the dendrogram, which was determined based on an exploratory analysis of the data. Since verbs contain important temporal information about diseases or habits of patients, they were excluded from grouping by using a list of Spanish verbs, including words in the infinitive tense.

Once word groups have been formed using hierarchical clustering, the global alignment NW algorithm is applied in each of these groups. The NW algorithm allows finding global similarity regions between two similar sequences, assigning positive scores for matching regions and negative scores for insertions, eliminations, and non-matching regions, which are represented in an alignment matrix [33]. Then, in a backtracking stage, NW traces the route of the aligned sequences. The proposed algorithm for automatically constructing the feature space uses the NW algorithm to represent groups of words through a common pattern in order to capture the diversity of the texts in terms of grammatical gender, grammatical number, and spelling errors. To do this, NW aligns the common letters of the group of words, considering as the base of the alignment the most frequent word according to the training set. Subsequently, our algorithm incorporates the metacharacter "{, max}" between the aligned letters, where "max" corresponds to the maximum number of different letters between the aligned elements of the group. An example of hierarchical clustering and posterior global alignment is shown in Fig. 2.

3) FINE-GRAIN TEXT ALIGNING: THE LOCAL SW algorithm

After the global alignment of the word groups, each pair of texts in the training set is aligned using the SW method.

Our algorithm for automatically constructing the feature space uses the SW method to find regions of local similarity between biomedical texts belonging to the same class to extract sequences of representative tokens. If the texts contain numbers, they are replaced by a token containing metacharacters to represent numerical intervals considering a range equal to five (see Fig. 3). This numerical range was chosen because it is used to represent the different levels of obesity [30]. For example, the token "24.3" is replaced by the pattern "2[0-4]{1}(?:[\.\,]\d+)?", while the token "25.4" is replaced by the pattern "2[5-9]{1}(?:[\.\,]\d+)?", where the numbers between the square brackets represent the unit digit and "(?:[\.\,]\d+)?" represents the decimal part of the number written using either a comma or a dot.

As in the NW method, SW assigns positive scores for matching regions and negative scores to insertions, eliminations, and non-matching regions, and represents them in a matrix to later find the sequences aligned in the backtracking stage [33]. The SW and NW methods differ mainly in the form in which the alignment matrix is initialized (zeros and negative values, respectively) [34].

4) FEATURE EXTRACTION AND SELECTION: AUTOMATIC GENERATION AND FILTERING OF REGULAR EXPRESSIONS

Once tokens have been extracted from the training set, our algorithm automatically generates the regular expressions, which form the feature space, by replacing the whitespaces with the metacharacter "\s*" (zero or more whitespaces) and adding a backslash ("\") to the non-alphanumeric characters (escape character). Finally, the algorithm assigns to each regular expression the class of text where the token was aligned.

Each regular expression is then filtered using the keywords listed in Table 1. The aim is to reduce the number of regular expressions that can be generated from the training texts, keeping only those expressions that are directly related to the thematic of the processed dataset. An example of the regular expression generation process for the positive class of the OBESITY STATUS dataset is shown in Fig. 3. The regular expression "the\s*patient" was filtered because it does not include a keyword for the dataset. Afterwards, each regular expression is evaluated in all the training texts to obtain the respective confusion matrices.

C. THE CREGEX BIOMEDICAL TEXT CLASSIFIER

For classification purposes, all regular expressions are applied to each instance of text under evaluation to assign a class to the instance. Depending on the number of regular expressions matching each instance, three possible scenarios may arise: (i) one or more regular expressions of the same class match a test text; (ii) no regular expression matches a test text; (iii) one or more regular expressions from different classes match a test text. In the first case, the CREGEX classifier assigns the only possible class to the test text. In the second case, CREGEX assigns to the test text the class associated with the highest SW similarity score computed according to the decision function:

$$class(x_i) = class(\operatorname*{argmax}_{x_j \in X} sw\_sim(x_i, x_j)). \tag{1}$$

Finally, in the third case, CREGEX assigns the class of the regular expression with the highest precision during the training step. More precisely, assuming that n_i regular expressions match the x_i test text, CREGEX classifies this example using the following decision function:

$$class(x_i) = class(\operatorname*{argmax}_{r \in [1, n_i]} P_r), \tag{2}$$

where P_r is the precision of the rth regular expression matching the test text x_i and was already computed during the training stage as the ratio between the number of correct matches and the total number of matches.

IV. RESULTS

A. PERFORMANCE EVALUATION

For comparison, SVM-based and NB-based classifiers were implemented to evaluate the performance of the CREGEX biomedical text classifier. In the case of SVM, a linear kernel was chosen, keeping the rest of the parameters by default, whereas for the NB, a multinomial model was chosen [35], [36]. For both classifiers, the BoW approach was used to represent the features in the form of a sequence of n-tokens, where n represents the number of tokens in the sequence (a sequence of n-tokens is also referred to as n-grams, where 1-grams are called unigrams, 2-grams are bigrams, etc.) [37]. The BoW counts the frequency of the features used regardless of the order in which they occur in the texts [38].

For the training and evaluation of the classifiers, 10-fold cross validation was implemented, executing the experiments 10 times to average the results of the equation metrics (3) to (4) [39], [40]. In the case of SVM, for the OBESITY TYPES dataset, the One-vs-All strategy was used and the metrics of the equations (3) to (4) were averaged considering the amount of test examples [35]:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}, \tag{3}$$

$$F1 = \frac{2TP}{2TP + FP + FN}, \tag{4}$$

where TP and TN correspond to the correct classifications (true positives and negatives), while FP and FN correspond to the incorrect classifications (false positives and negatives). Additionally, t-student paired tests were performed between the performance of CREGEX and each other classifier with an $\alpha = 0.05$. Finally, learning curves were constructed to analyze the performance of the classifiers according to the number of training examples used, 50 examples being selected each time to complete the total (passive or random sampling) [8].

Finally, to analyze the classification error of CREGEX during the training and testing, the Zero-One-Loss metric (L) was used, which assigns a one to the classification errors and

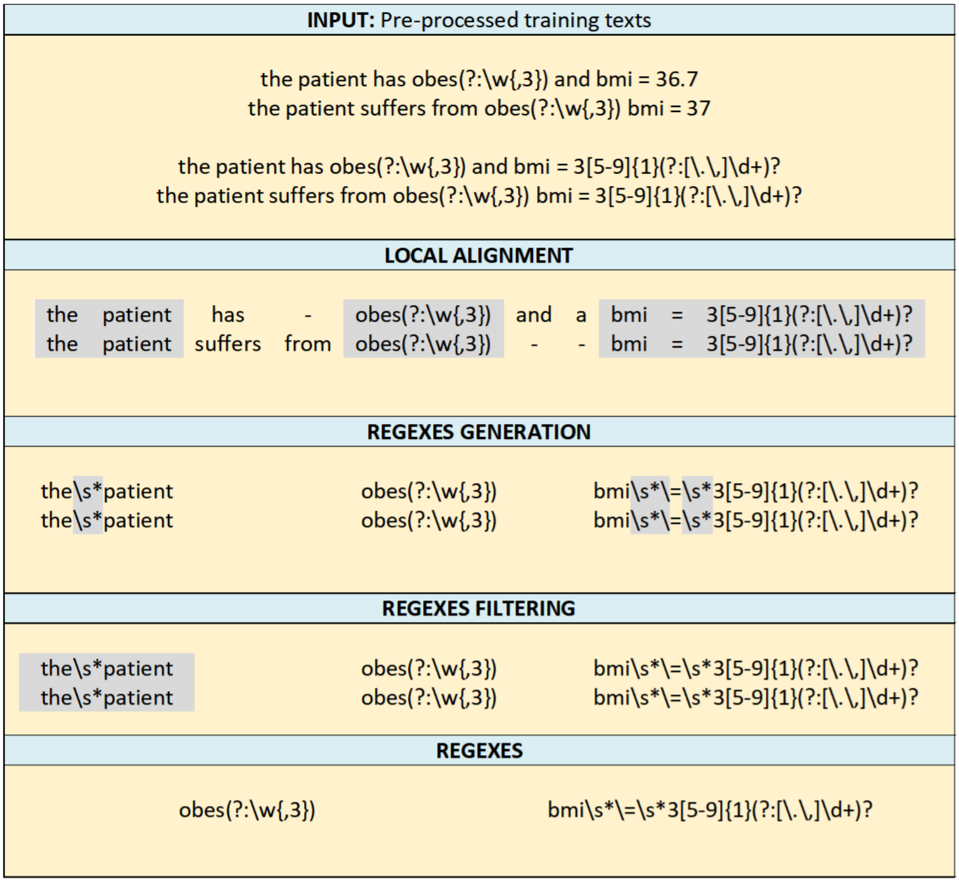
INPUT: Pre-processed training texts
the patient has obes(?:\w{,3}) and bmi = 36.7
the patient suffers from obes(?:\w{,3}) bmi = 37
the patient has obes(?:\w{,3}) and bmi = 3[5-9]{1}(?:[\.\,]\d+)?
the patient suffers from obes(?:\w{,3}) bmi = 3[5-9]{1}(?:[\.\,]\d+)?
LOCAL ALIGNMENT
the patient has - obes(?:\w{,3}) and a bmi = 3[5-9]{1}(?:[\.\,]\d+)?
the patient suffers from obes(?:\w{,3}) - - bmi = 3[5-9]{1}(?:[\.\,]\d+)?
REGEXES GENERATION
the\s*patient obes(?:\w{,3}) bmi\s*\=\s*3[5-9]{1}(?:[\.\,]\d+)?
the\s*patient obes(?:\w{,3}) bmi\s*\=\s*3[5-9]{1}(?:[\.\,]\d+)?
REGEXES FILTERING
the\s*patient obes(?:\w{,3}) bmi\s*\=\s*3[5-9]{1}(?:[\.\,]\d+)?
the\s*patient obes(?:\w{,3}) bmi\s*\=\s*3[5-9]{1}(?:[\.\,]\d+)?
REGEXES
obes(?:\w{,3}) bmi\s*\=\s*3[5-9]{1}(?:[\.\,]\d+)?

FIGURE 3. Example of regular expressions generation for the positive class of the OBESITY STATUS dataset.

TABLE 2. Classification results.

Classifier	OBESITY STATUS		OBESITY TYPES		SMOKING STATUS	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
CREGEX	97.68	98.45	91.20	90.51	88.77	90.08
SVM-N1	97.32	98.22	86.22	86.06	86.39	87.45
SVM-N2	96.27	97.57	89.64	89.27	87.50	88.86
SVM-N3	91.80	94.83	88.24	87.47	82.58	85.57
NB-N1	84.96	90.14	70.48	65.96	75.42	77.23
NB-N2	87.85	92.15	81.32	79.12	80.52	81.87
NB-N3	88.04	92.37	84.23	82.68	82.68	84.05

a zero to the correct ones [41]:

$$L(y_i, y_i') = \begin{cases} 1 & y_i \neq y_i' \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

where y_i and y_i' represent the predicted and the true class labels, respectively. Table 2 lists the classification results of CREGEX, NB, and SVM for all datasets in terms of the performance metrics ACC (%) and F1 (%). In the case of SVM and NB, unigrams (N1), bigrams (N2), and trigrams (N3) were used as features for the training of the classifiers. In all cases the performance of CREGEX was better than SVM and NB with statistically significant differences (p-value<0.05).

Fig. 4 shows the learning curves of the classifiers according to the number of training examples and the performance obtained in terms of ACC and F1. In all cases, with the exception of a small portion in the OBESITY STATUS dataset, the performance of CREGEX was better than SVM and NB, with the most significant differences encountered in the OBESITY TYPES dataset. It is also observed that the performance of SVM was better than NB, and that the lowest performances were obtained by NB-N1.

Tables 3, 4, and 5 show the minimum number of training examples to achieve a given performance in terms of ACC and F1 according to what is observed in Fig. 4. In all cases the best performance of CREGEX was superior to the best performance of SVM and NB. On the other hand, CREGEX used a smaller number of training examples to achieve the highest performance in all cases.

Fig. 5 shows the error curves of CREGEX in terms of Zero-One Loss during training and testing (normalized). In all cases the training error is less than the test error. The smallest errors were found in the OBESITY STATUS dataset. In all cases the curve of test error is kept relatively stable after decreasing until completing the total number of training examples.

Fig. 6 shows the percentage distribution of CREGEX classification errors according to the three possible cases: regular expressions belonging to the same class (first case), no regular expression (second case) and regular expressions

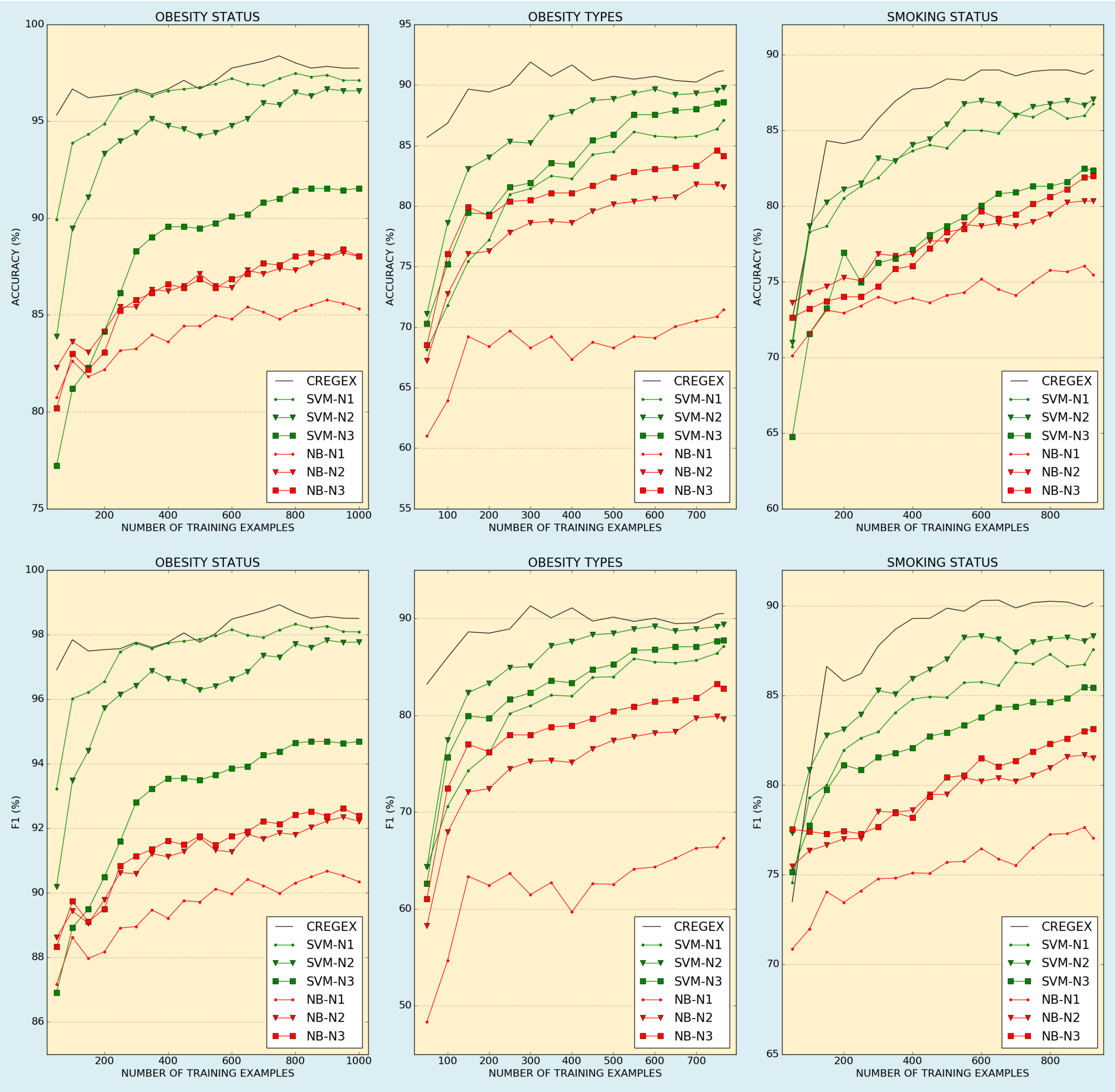

FIGURE 4. Learning curves of the classifiers. Top: ACC, Bottom: F1.

TABLE 3. Results of the learning curves of the OBESITY TYPES dataset.

Metric (%) ≥	Minimum training sample size													
	CREGEX		SVM-N1		SVM-N2		SVM-N3		NB-N1		NB-N2		NB-N3	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
65	50	50	50	100	50	100	50	100	150	650	50	100	50	100
75	50	50	150	200	100	100	100	100	-	-	150	300	100	150
85	50	100	550	550	250	300	450	500	-	-	-	-	-	-
87	150	150	766	766	350	350	550	650	-	-	-	-	-	-
88	150	150	-	-	450	450	700	-	-	-	-	-	-	-
89	150	300	-	-	550	600	-	-	-	-	-	-	-	-
90	250	300	-	-	-	-	-	-	-	-	-	-	-	-

- Indicates that the model did not reach the performance indicated in the corresponding row

TABLE 4. Results of the learning curves of the OBESITY STATUS dataset.

Metric (%) ≥	Minimum training sample size													
	CREGEX		SVM-N1		SVM-N2		SVM-N3		NB-N1		NB-N2		NB-N3	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
85	50	50	50	50	100	50	250	50	650	50	250	50	250	50
87	50	50	50	50	100	50	300	100	-	50	500	50	650	50
88	50	50	50	50	100	50	300	100	-	100	900	50	800	50
89	50	50	50	50	100	50	350	150	-	350	-	100	-	100
90	50	50	100	50	150	50	600	200	-	550	-	250	-	250
92	50	50	100	50	200	100	-	300	-	-	-	850	-	700
94	50	50	150	100	300	150	-	700	-	-	-	-	-	-
96	100	50	250	100	800	250	-	-	-	-	-	-	-	-
97	450	100	600	250	-	700	-	-	-	-	-	-	-	-
98	700	450	-	600	-	-	-	-	-	-	-	-	-	-

- Indicates that the model did not reach the performance indicated in the corresponding row

TABLE 5. Results of the learning curves of the SMOKING SATTUS dataset.

Metric (%) ≥	Minimum training sample size													
	CREGEX		SVM-N1		SVM-N2		SVM-N3		NB-N1		NB-N2		NB-N3	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
65	50	50	50	50	50	50	100	50	50	50	50	50	50	50
75	100	100	100	100	100	50	200	50	600	400	200	50	350	50
85	300	150	550	550	500	300	-	900	-	-	-	-	-	-
87	400	300	-	800	925	500	-	-	-	-	-	-	-	-
88	500	350	-	-	-	550	-	-	-	-	-	-	-	-
89	-	400	-	-	-	-	-	-	-	-	-	-	-	-
90	-	600	-	-	-	-	-	-	-	-	-	-	-	-

- Indicates that the model did not reach the performance indicated in the corresponding row

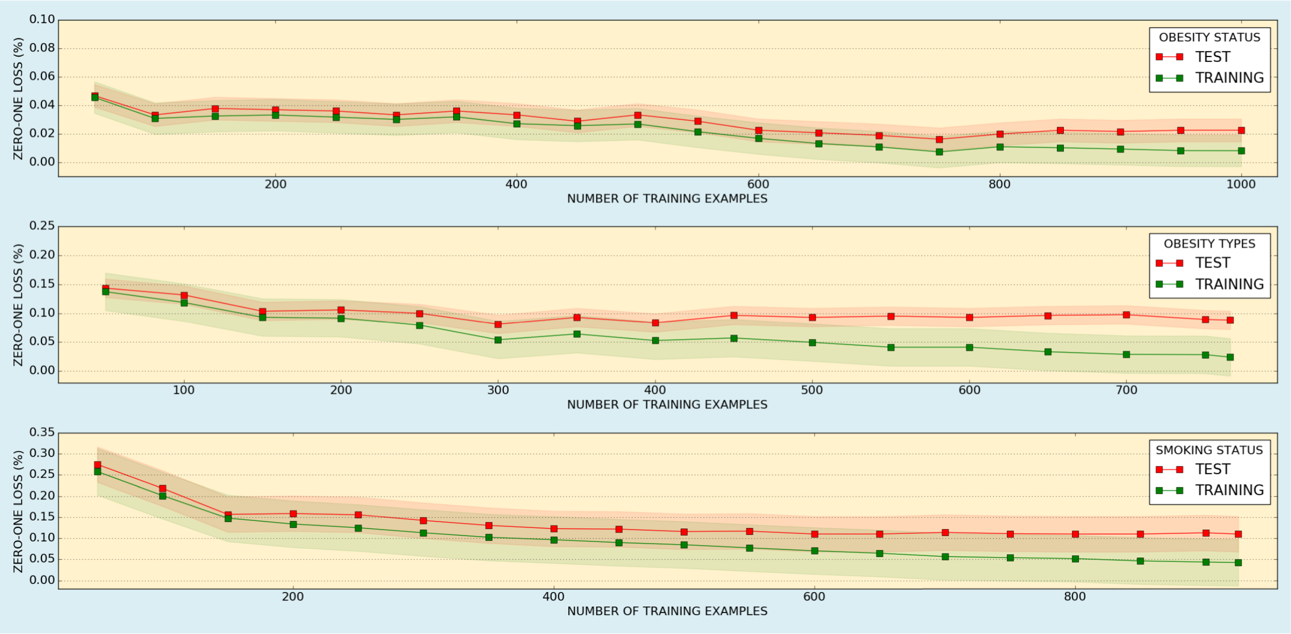

FIGURE 5. CREGEX error curves (Zero-One Loss) during training and testing.

of different classes (third case). Taking into account all the datasets, the smallest number of errors was obtained in the second possible case with CREGEX. The largest number of classification errors in the OBESITY STATUS dataset were found in the third case, while in the OBESITY TYPES and SMOKING STATUS datasets they were observed in the first case (unambiguous case).

V. CONCLUSION AND FUTURE WORK

This article has presented a biomedical text classifier based on regular expressions called CREGEX, which is a more flexible classifier in comparison to the traditional classification algorithms that require a matrix representation of the data, due to its ability to consider words sequences and represent the lexical variance of the texts through regular expressions.

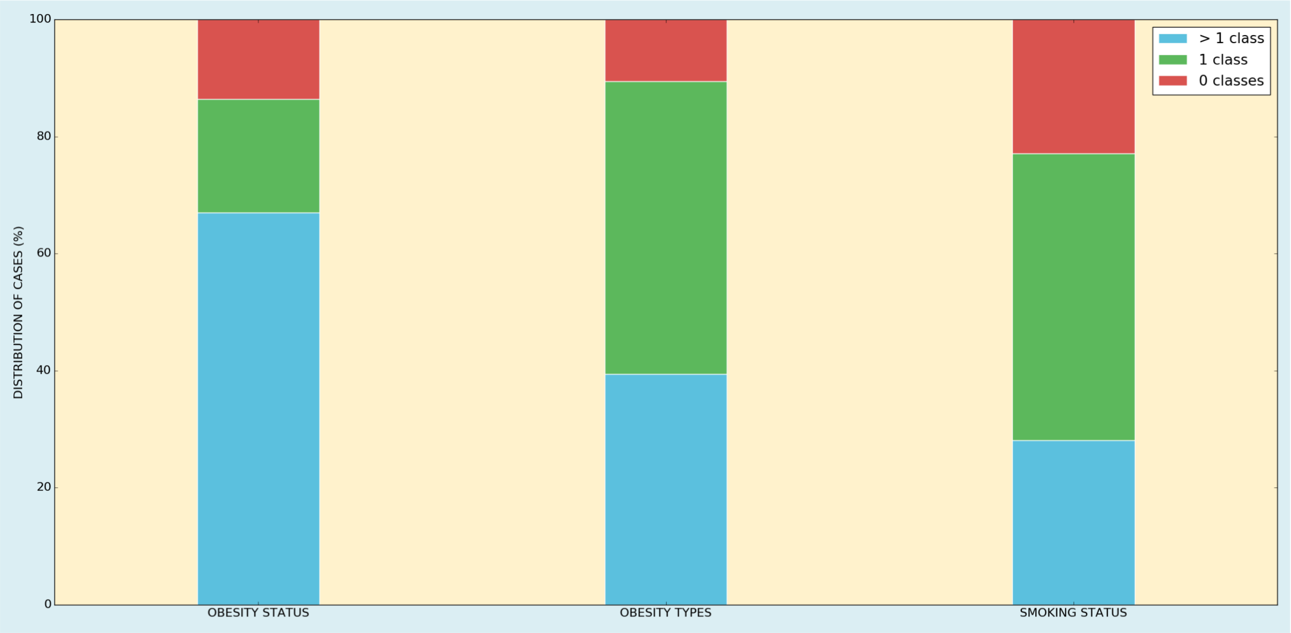

FIGURE 6. CREGEX classification errors in each dataset.

Although CREGEX was applied in biomedical texts written in Spanish, it could be extended to other languages, taking into consideration the grouping of similar words (exclusion of verbs) and the filtering of regular expressions (keywords).

The performance of CREGEX was better than SVM and NB with statistically significant differences (p-value<0.05), as shown in Table 2. On the other hand, Fig. 6 shows the percentage distribution of the classification errors in the three cases of CREGEX in the tests set: regular expressions belonging to the same classes (first type), no regular expression (second type), and regular expressions of different classes (third type). The smallest number of classification errors occurred in the second case, indicating that CREGEX allowed the generation of enough regular expressions to capture the lexical diversity of the texts for all datasets and that the use of SW similarity made it possible to assign correct classes. The largest number of classification errors were distributed over the rest of the cases. For example, in the OBESITY TYPES and SMOKING STATUS datasets, the largest number of classification errors occurred in the unambiguous cases (first type). The above results could be explained by the fact that both health conditions can be described using terms that include many numerical and temporal elements (negations). Although CREGEX was able to generate enough regular expressions for the most representative terms of each classification problem using the common tokens (aligned tokens) extracted by the Smith-Waterman algorithm, it could work on the incorporation of non-aligned tokens to better capture the lexical variance of biomedical texts. With regard to the OBESITY STATUS dataset, the number of errors relating to the unambiguous cases (first type) was lower than for the rest of the datasets. This can be explained by the fact that this dataset has a more limited vocabulary to refer to the presence or absence of obesity so that CREGEX was able to better capture the lexical diversity of representative terms. Furthermore, according to the learning curves of the classifiers in Fig. 4, it can be observed that in general CREGEX performed better than SVM and NB in all datasets. In addition, CREGEX reached the peak performance with fewer training examples (see Tables 3, 4 and 5). This is significant because the annotation process is a labor intensive and slow.

The results shown by the error curves (see Fig. 5) indicate that CREGEX is able to learn correctly from the training examples, avoiding underfitting and overfitting problems. This is demonstrated by the fact that the error of the test set curve remains relatively stable once it decreases along with the training error curve.

As regards future research, we intend to focus on improving specific elements of CREGEX, particularly in relation to the grouping stages of similar words and the filtering of regular expressions by automatic methods. Additionally, we will work on an active learning function that would make it possible to identify the most informative training examples for CREGEX. It is expected that CREGEX will be able to achieve its highest performance using a smaller number of training examples compared to passive learning (see Fig. 4) [42]. Finally, we will also compare CREGEX's performance with classifiers trained on word embeddings models such as BERT, Word2vec, GloVe, and fastText, which are considered an improvement over the BoW-based classifiers.

ACKNOWLEDGMENT

The authors would like to thank Informatics Unit with the Guillermo Grant Benavente Hospital in Concepción, Chile, for providing datasets.

REFERENCES

[1] S. Kaisler, F. Armour, J. A. Espinosa, and W. Money, "Big data: Issues and challenges moving forward," in *Proc. 46th Hawaii Int. Conf. Syst. Sci.*, Jan. 2013, pp. 995–1004.

[2] M. Liu, L. Pan, and S. Liu, "To transfer or not: An online cost optimization algorithm for using two-tier storage-as-a-service clouds," *IEEE Access*, vol. 7, pp. 94263–94275, 2019.

[3] A. Shachak and S. Reis, "The impact of electronic medical records on patient–doctor communication during consultation: A narrative literature review," *J. Eval. Clin. Pract.*, vol. 15, no. 4, pp. 641–649, 2008.

[4] J. Adeva, J. Atxa, M. Carrillo, and E. Zengotitabengoa, "Automatic text classification to support systematic reviews in medicine," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1498–1508, 2014.

[5] R. Sinoara, J. Camacho-Collados, R. Rossi, R. Navigli, and S. Rezende, "Knowledge-enhanced document embeddings for text classification," *Knowl.-Based Syst.*, vol. 163, pp. 955–971, Jan. 2018.

[6] X. Deng, Y. Li, J. Weng, and J. Zhang, "Feature selection for text classification: A review," *Multimedia Tools Appl.*, vol. 78, no. 3, pp. 3797–3816, 2019.

[7] S. Hassan, M. Rafi, and M. Shaikh, "Comparing SVM and Naïve Bayes classifiers for text categorization with wikitology as knowledge enrichment," in *Proc. IEEE 14th Int. Multitopic Conf.*, Dec. 2011, pp. 31–34.

[8] D. D. A. Bui and Q. Zeng-Treitler, "Learning regular expressions for clinical text classification," *J. Amer. Med. Informat. Assoc.*, vol. 21, no. 5, pp. 850–857, 2015.

[9] M. Cui, R. Bai, Z. Lu, X. Li, U. Aickelin, and P. Ge, "Regular expression based medical text classification using constructive heuristic approach," *IEEE Access*, vol. 7, pp. 147892–147904, 2019.

[10] Z. Zhong, J. Guo, W. Yang, T. Xie, J.-G. Lou, T. Liu, and D. Zhang, "Generating regular expressions from natural language specifications: Are we there yet?" in *Proc. Workshops 32nd AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, 2018, pp. 791–794.

[11] D. Denis, "High-performance regular expression matching with Parabix and LLVM," M.S. thesis, School Comput. Sci., Simon Fraser Univ., Burnaby, BC, Canada, 2014.

[12] V. Cotik, V. Stricker, J. Vivaldi, and H. R. Hontoria, "Syntactic methods for negation detection in radiology reports in Spanish," in *Proc. 15th Workshop Biomed. Natural Lang. Process. (BioNLP)*, Berlin, Germany, Aug. 2016, pp. 156–165.

[13] A. Rosier, A. Burgun, and P. Mabo, "Using regular expressions to extract information on pacemaker implantation procedures from clinical reports," in *Proc. AMIA Annu. Symp.*, 2008, p. 81.

[14] Y. Kang and M. Kayaalp, "Extracting laboratory test information from biomedical text," *J. Pathol. Inf.*, vol. 4, no. 1, p. 23, 2013.

[15] 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. Informat.*, vol. 54, pp. 186–190, Apr. 2015.

[16] D. D. A. Bui, G. Del Fiol, J. F. Hurdle, and S. Jonnalagadda, "Extractive text summarization system to aid data extraction from full text in systematic review development," *J. Biomed. Informat.*, vol. 64, pp. 265–272, Dec. 2016.

[17] N. Milosevic, C. Gregson, R. Hernandez, and G. Nenadic, "A framework for information extraction from tables in biomedical literature," *Int. J. Document Anal. Recognit.*, vol. 22, no. 1, pp. 55–78, Mar. 2019.

[18] A. Bartoli, G. Davanzo, A. D. Lorenzo, and E. S. E. Medvet, "Automatic synthesis of regular expressions from examples," *IEEE Comput. Soc.*, vol. 47, no. 12, pp. 72–80, Dec. 2013.

[19] Y. Li, R. Krishnamurthy, S. Raghavan, and S. Vaithyanathan, "Regular expression learning for information extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, Honolulu, HI, USA, Oct. 2008, pp. 21–30.

[20] R. Babbar and N. Singh, "Clustering based approach to learning regular expressions over large alphabet for noisy unstructured text," in *Proc. 4th Workshop Analytics Noisy Unstructured Text Data (AND)*, New York, NY, USA, 2010, pp. 43–50, doi: 10.1145/1871840.1871848.

[21] K. Murthy, P. Deepak, and P. M. Deshpande, "Improving recall of regular expressions for information extraction," in *Proc. Int. Conf. Web Inf. Syst. Eng.* Paphos, Cyprus: Springer, 2012, pp. 455–467.

[22] M. Shahbaz, P. Mcminn, and M. Stevenson, "Automatic generation of valid and invalid test data for string validation routines using web searches and regular expressions," *Sci. Comput. Program.*, vol. 97, pp. 405–425, Jan. 2015.

[23] P. Arcaini, A. Gargantini, and E. Riccobene, "Fault-based test generation for regular expressions by mutation," *Softw. Test., Verification Rel.*, vol. 29, nos. 1–2, p. e1664, 2019.

[24] T. Wu and W. Pottenger, "A semi-supervised active learning algorithm for information extraction from textual data," *J. Assoc. Inf. Sci. Technol.*, vol. 56, no. 3, pp. 258–271, 2005.

[25] F. Brauer, R. Rieger, A. Mocan, and W. M. Barczynski, "Enabling information extraction by inference of regular expressions from sample entities," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 1285–1294.

[26] A. Cetinkaya, "Regular expression generation through grammatical evolution," in *Proc. 9th Annu. Conf. Companion Genetic Evol. Comput.*, 2007, pp. 2643–2646.

[27] A. Bartoli, A. De Lorenzo, E. Medvet, and F. Tarlao, "Automatic search-and-replace from examples with coevolutionary genetic programming," *IEEE Trans. Cybern.*, to be published.

[28] P. Wang, G. R. Bai, and K. T. Stolee, "Exploring regular expression evolution," in *Proc. IEEE 26th Int. Conf. Softw. Anal., Evolution Reeng. (SANER)*, Feb. 2019, pp. 502–513.

[29] R. L. Figueroa, D. A. Soto, and E. J. Pino, "Identifying and extracting patient smoking status information from clinical narrative texts in Spanish," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2014, pp. 2710–2713.

[30] R. L. Figueroa and C. A. Flores, "Extracting information from electronic medical records to identify the obesity status of a patient based on comorbidities and bodyweight measures," *J. Med. Syst.*, vol. 40, no. 8, pp. 1–9, Aug. 2016.

[31] Y. Lakhman, S. S. Katz, D. A. Goldman, D. Yakar, H. A. Vargas, R. E. Sosa, M. Miccò, R. A. Soslow, H. Hricak, and N. R. Abu-Rustum, "Diagnostic performance of computed tomography for preoperative staging of patients with non-endometrioid carcinomas of the uterine corpus," *Ann. Surgical Oncol.*, vol. 23, no. 4, pp. 1271–1278, 2016.

[32] G. Sidorov, H. Gómez-Adorno, I. Markov, D. Pinto, and N. Loya, "Computing text similarity using tree edit distance," in *Proc. Annu. Conf. North Amer. Fuzzy Inf. Process. Soc. (NAFIPS) Jointly 5th World Conf. Soft Comput. (WConSC)*, Aug. 2015, pp. 1–4.

[33] S. Ren, N. Ahmed, K. Bertels, and Z. Al-Ars, "GPU accelerated sequence alignment with traceback for GATK haplotypecaller," *BMC Genomics*, vol. 20, no. 2, pp. 103–116, 2019.

[34] J. Daily, "Parasail: SIMD C library for global, semi-global, and local pairwise sequence alignments," *BMC Bioinformat.*, vol. 17, no. 1, p. 81, Feb. 2016.

[35] V. K. Chauhan, K. Dahiya, and A. Sharma, "Problem formulations and solvers in linear SVM: A review," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 803–855, 2019.

[36] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, "Comparison between multinomial and bernoulli Naïve Bayes for text classification," in *Proc. Int. Conf. Autom., Comput. Technol. Manage. (ICACTM)*, 2019, pp. 593–596.

[37] M. Dickinson and A. Smith, "Simulating dependencies to improve parse error detection," in *Proc. 15th Int. Workshop Treebanks Linguistic Theor.*, Bloomington, IN, USA, Jan. 2017, pp. 76–88.

[38] A. M. El-Halees, "Arabic opinion mining using distributed representations of documents," in *Proc. Palestinian Int. Conf. Inf. Commun. Technol. (PICICT)*, May 2017, pp. 28–33.

[39] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques* (Series in Data Management Systems). Burlington, MA, USA: Morgan Kaufmann, 2011.

[40] G. Vanwinckelen and H. Blockeel, "On estimating model accuracy with repeated cross-validation," in *Proc. 21st Belg.-Dutch Conf. Mach. Learn.*, Ghent, Belgium, 2012, pp. 39–44.

[41] A. Charuvaka and H. Rangwala, "HierCost: Improving large scale hierarchical classification with cost sensitive learning," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, Porto, Portugal, 2015, pp. 675–690.

[42] R. Figueroa, Q. Zeng-Treitler, L. H. Ngo, S. Goryachev, and E. Wiechmann, "Active learning for clinical text classification: Is it better than random sampling?" *J. Amer. Med. Informat. Assoc.*, vol. 19, no. 5, pp. 809–816, 2012.

CHRISTOPHER A. FLORES received the B.S. degree in biomedical engineering and the M.S. degree in electrical engineering from the Universidad de Concepción, Concepción, Chile, in 2005 and 2017, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include natural language processing, text mining, and machine learning.

ROSA L. FIGUEROA received the B.Eng. degree from the University of Concepción, in 2004, and the Ph.D. degree in electrical engineering from the University of Concepción, in 2012. Her Ph.D. thesis explored different methods to obtain useful information from free text. She is currently a Faculty Member and a Researcher, in biomedical engineering degree part, with the Electrical Engineering Department, University of Concepción, and a Technical Board Member of the National Center on Health Information Systems. She has scientific publications in journals and conference proceedings. Her research interest is within the medical informatics area, mainly machine learning and text mining. She is currently working in research projects related to secondary use of medical data and text classification.

JORGE E. PEZOA (Member, IEEE) received the B.S. degree in electronics engineering and the M.S. degree in electrical engineering from the Universidad de Concepción, Concepción, Chile, in 1999 and 2003, respectively, and the Ph.D. degree in electrical engineering from The University of New Mexico, NM, USA, in 2010.

He is currently an Associate Professor and an Associate Chair of the Departamento de Ingeniería Eléctrica, Universidad de Concepción, in Concepción, Chile. He is also a member of the Society of Photo-Optical Instrumentation Engineers (SPIE), Optical Society of America (OSA), and Association for Computing Machinery (ACM). His research interests include distributed computing, pattern recognition, statistical signal processing, network optimization, and hyperspectral image and signal processing for industrial processes.

QING ZENG-TREITLER is currently a tenured Professor with the Department of Clinical Research and Leadership, George Washington University. She is also the Director of the Biomedical Informatics Center, George Washington University, and also the Co-Director of the Center for Data Science and Outcome Research, Washington DC Veterans Affairs (DCVA) Medical Center. She has published 130 peer-reviewed articles and has served as the PI and Co-PI on over a dozen VA HSR&D, NIH, and DOD funded research projects.