AN INTERPRETABLE CLASSIFICATION FRAMEWORK FOR INFORMATION EXTRACTION FROM ONLINE HEALTHCARE FORUMS

A Thesis

by

JUN GAO

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Xia Hu
Committee Members,	Ricardo Gutierrez-Osuna
	Xiaoning Qian
Head of Department,	Dilma Da Silva

December 2017

Major Subject: Computer Science

ABSTRACT

Online Healthcare Forums (OHFs) have become increasingly popular for patients to share their health-related experiences. The healthcare-related texts posted in OHFs could help doctors and patients better understand specific diseases and the situations of other patients. To extract the meaningful information of a post, a common way is to classify the sentences into several pre-defined categories of different semantics. However, the unstructured form of online posts brings challenges to existing classification algorithms. In addition, though many sophisticated classification models such as deep neural networks may have good predictive power, it is hard to interpret the models and the prediction results, which is, however, critical in healthcare applications. To tackle the challenges above, we propose an effective and interpretable OHF post classification framework. Specifically, we classify sentences into three classes: Medication, Symptom, and Background. Each sentence is projected into an interpretable feature space. A forest-based model is developed for categorizing OHF posts. An interpretation method is also developed, where the decision rules can be explicitly extracted to gain an insight of useful information in texts. Experiments and an application system will be implemented based on the proposed framework.

ACKNOWLEDGMENTS

Dr. Xia Hu has always been a great advisor. His guide took me into the research of data mining, his advice inspired me during this thesis, and his encouragement helped me brace up when facing difficulties. I am grateful for everything he does during my research. Also, I would like to thank Dr. Gutierrez and Dr. Qian for their support and advice as my committee members.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a thesis committee consisting of Professor Xia Hu and Professor Ricardo Gutierrez-Osuna of the Department of Computer Science and Engineering and Professor Xiaoning Qian of the Department of Electrical and Computer Engineering.

All work for the thesis was completed by the student, in collaboration with Ninghao Liu of the Department of Computer Science and Engineering.

Funding Sources

This work was made possible in part by Defense Advanced Research Projects Agency (DARPA) under Grant Number N66001-17-2-4031 and Grant Number W911NF-16-1-0565 and by National Science Foundation (NSF) under Grant Number IIS-1657196.

Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the DARPA or NSF.

NOMENCLATURE

CART	Classification and Regression Trees
DP	Discriminative Pattern
DPClass	Discriminative Patterns-Based Classification Framework
FSP	Frequent Sequential Pattern
FLSP	Frequent Labeled Sequential Pattern
LSP	Labeled Sequential Pattern
OHF	Online Health Forum
SVM	Support Vector Machine
UMLS	Unified Medical Language System

TABLE OF CONTENTS

Page

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

1. INTRODUCTION*

1.1 Background

The past few years have witnessed the increasing popularity of online health forums (OHFs), such as WebMD Discussions, Patient, etc., as communication platforms among patients. According to a survey by PwC in 2012, 54% of 1060 participants are comfortable with their doctors getting information related to their health conditions from online physician communities [1]. OHFs can be used for patients to ask for suggestions and share experiences. The abundant user-generated content related to healthcare on the OHFs could provide insightful information to the other patients, medical doctors, and decision makers to promote the understanding about diseases and the health conditions of patients.

To extract insightful information from OHF posts, a commonly adopted strategy is to split posts into sentences and classify each sentence into different categories according to their semantical meanings [2][3]. For example, Figure 1.1 shows a post from an OHF called patient.info. We highlight the sentences about symptoms in orange, and the one about medication in violet. The former ones provide the information about the user's symptoms, reflected by the terms "heartburn", "acid reflux", "abdominal pain" and "IBS". The latter one tells user's medication treatment, where the term "nexium" presents the medication for the disease. These pieces of information can help other users to gain a more comprehensive understanding of the disease.

However, it is a challenging task to effectively analyze the expressions in online health forums. First, the user-generated content in OHFs is usually unstructured and contains

*Reprinted with permission from "An Interpretable Classification Framework for Information Extraction from Online Healthcare Forums" by Jun Gao et al. 2017. Journal of Healthcare Engineering, Volume 2017 (2017), Copyright 2017 by Jun Gao et al.

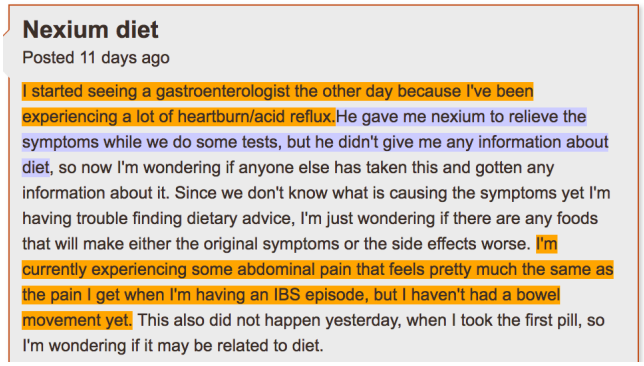

Figure 1.1: An example of online health forum post

background information that is relatively less important to analyze [3]. The irregularity and noises in data impede us from directly applying existing classification models to analyze posts automatically. A more sophisticated classification framework is needed for processing unstructured data in OHFs, in order to extract useful patterns (e.g., terms, text sequences) for accurate categorization. Second, when categorizing post sentences into different classes, it is difficult to make the tradeoff between classification accuracy and interpretability [4][5]. In health related tasks, besides desirable classification performance, human-understandable explanations for classification results are also crucial, because patients or doctors will not take the risk to trust the predictions they do not understand. Complex models (e.g., deep neural networks, SVM) are accurate in classification, but they do not directly provide the reasons for individual classification results. Simple models such as linear classifiers and decision trees can provide interpretations along with classification outcomes, but usually they cannot achieve comparable performances as complex models.

1.2 Contributions

In this paper, we propose an effective framework for analyzing OHF posts. We propose to develop a random forest model to classify the sentences into three categories, i.e.,

Medication, Symptom, and Background, in order to get an accurate understanding of the role of each sentence in the overall expression of the health situation. Besides, human-understandable interpretations for classification results are generated for the forest model. To enable interpretation, the features involved in the classification task are designed in a human understandable manner. Moreover, the contribution of features to a classification instance can be explicitly measured by the decision rules constructed during training process [6][7][8]. Specifically, we represent healthcare-related sentences with various semantic features such as labeled sequential patterns (LSPs), UMLS semantic type features[3], sentence-based and heuristic features. LSPs represent the frequent tag-based patterns in texts. UMLS features indicate the existence terminologies defined by domain experts. In this way, each unstructured sentence is mapped to the feature space which facilitates further analysis. Also, word-based and heuristic information can also be used to enhance the classification performance. The contributions of this paper are summarized as below:

- We propose a forest-based framework to deal with the healthcare-related text classification problem. Labeled sequential pattern features are involved in characterizing the unstructured healthcare-related texts from both syntactic and semantic levels.
- We develop a method for constructing decision rules integrated from decision trees in forest-based models to achieve model interpretability.
- The effectiveness and interpretability of our framework are demonstrated through experiments on a real OHF dataset, where we analyze the interpretations provided by our framework in detail.

1.3 Framework Overview

In this section, we will briefly introduce each module of our proposed framework (Figure 1.2) including data pre-processing, interpretable feature extraction, and forest-based

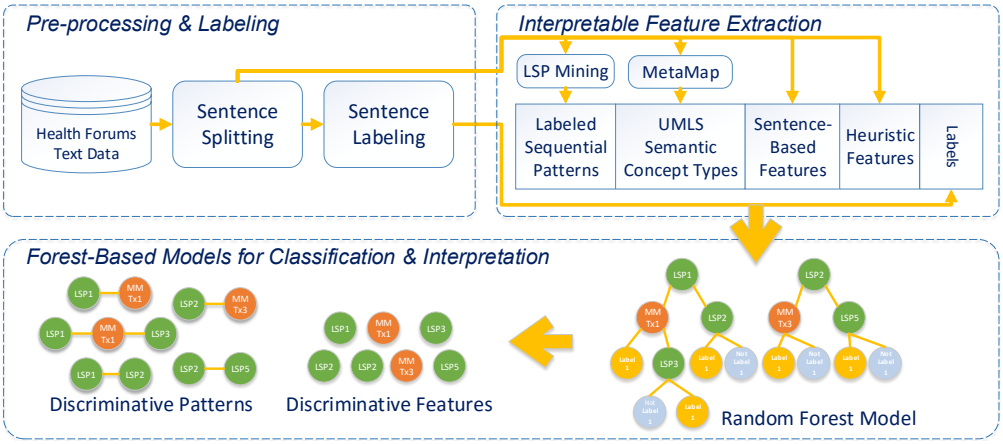

Figure 1.2: An overview of the interpretable classification framework

models for classification and interpretation. We categorize each sentence of posts into one of the three categories: *Medication*, *Symptom*, and *Background*. The definition of each category is given as below.

- *Medication:* If a sentence contains information relevant to curing diseases, treating any medical conditions, relieving any symptoms of diseases, or preventing any diseases, then we assign the sentence to the *Medication* category.

- *Symptom:* If a sentence contains any contents relevant to departures from normal functioning or feelings of individuals, which may express the phenomenon affected by diseases, we assign the sentence to the *Symptom* category.

- *Background:* If a sentence cannot be classified to Medication or Symptom category, then we assign the sentence to *Background* category.

Given a sentence *"I am taking 90 units Lantus twice a day"* for classification, for example, we will first convert it into an instance in a feature space through pre-processing to identify the number term "90", the drug term "Lantus", the frequency term "twice a day", the context of each term, etc. Then we will use the forest-based model to classify

the sentence, along with the explanations based on the discriminative features identified by the model.

1.3.1 Module 1: Pre-processing and Labeling

In this module, we split the collected online health community posts into sentences and manually assign each sentence one label from the classes {*Medication*, *Symptom*, *Background*}. Formally, let $\mathbb{H}$ be the healthcare-related natural language space, and $\mathbb{L} = \{$*Medication*, *Symptom*, and *Background*$\}$ be the target label space. Suppose a collection of N labeled sentences

$$\mathcal{S} = \{(s_i, l_i) | 1 \leq i \leq N, s_i \in \mathbb{H}, l_i \in \mathbb{L}\}$$

are available for model training and testing, s_i represents the original text of the i-th sentence, and l_i represents the label of the i-th sentence. In other words, each sentence is labeled as *Medication*, *Symptom*, or *Background*.

1.3.2 Module 2: Interpretable Features Extraction

In this module, we propose the feature extraction method $f : \mathbb{H} \rightarrow \mathbb{R}^D$ to convert healthcare-related sentences into instances in a D-dimensional numerical space, where D is the number of features used to represent each sentence. In this way, we can represent each unstructured sentence with a numerical vector, which facilitates model training and testing. After that, the overall dataset is transformed to

$$\mathcal{X} = f(\mathcal{S}) = \{(\mathbf{x}_i, l_i) | 1 \leq i \leq N\},$$

where $\mathcal{S}$ is the original labeled sentence dataset, N is the number of sentences, while $\mathbf{x}_i = (x_1, x_2, \ldots, x_D)$ is the resultant numerical instance represented by D features. These

features are also intuitive and insightful to help people better understand the sentences. We will discuss this module in detail in Section 3.

1.3.3 Module 3: Forest-Based Models for Classification & Interpretation

In this module, the task is to train a model $F : \mathbb{R}^D \rightarrow \mathbb{L}$ that can classify an instance into a class from {*Medication*, *Symptom*, or *Background*} and interpret the sentences belonging to class *Medication* and *Symptom*. We mainly introduce building forest-based models to classify and interpret the instances: (1) Random forests [9] are grown on the numerical instances obtained from the feature engineering module, which can be interpreted by the features of higher importance according to some criterion, e.g. Gini impurity. (2) DPClass [7] is a method based on random forest models to extract discriminative combinations of decision rules in the forest, which can be implemented by using forward selection to choose the top combinations. This module will be discussed in detail in Section 4.

2. LITERATURE REVIEW*

2.1 Medication Information Extraction

Previous medication information extraction research mainly focused on extracting medication information from clinical notes and online healthcare-related texts. [22], by Patrick et al., proposes to use conditional random fields and support vector machines to classify a variety of sets of generated features based on the results of records pre-processed by a sentence boundary detector and a tokenizer. Specifically, the approach is based on cascaded classifiers, i.e. conditional random fields and support vector machines. The first classifier is responsible for identifying named entities while the second is for classifying entity relationships. [23], by Sirohi et al., proposes to use three sets of drug lexicons as sources to conduct medication extraction. [24] by Xu et al. claims that using simple methods like regular expressions is not effective enough to extract medication information in the free texts from clinical notes. This paper proposes to adopts a variety of tags, including drug lexicons, number tag, etc., to tag the corresponding terms in sentences and parse sentences into structured forms by using a dynamic programming parsing method. [25] proposes an unsupervised method to extract adverse drug reactions on the online health forums. This paper adopts a probabilistic topic model to mine adverse drug reactions from OHF posts. [26] proposes to mine drug-related adverse events on large-scale tweets by using support vector machines.

*Part of this section is reprinted with permission from “An Interpretable Classification Framework for Information Extraction from Online Healthcare Forums" by Jun Gao et al. 2017. Journal of Healthcare Engineering, Volume 2017 (2017), Copyright 2017 by Jun Gao et al.

2.2 Sentence Classification

In terms of sentence classification, previous works mainly focus on classifying the sentences. [3], by Sondhi et al., uses conditional random fields and support vector machines to classify texts from online health forums and word-based features, semantic features, and other heuristic features. The problem of this representation is that word-based features have a huge dimension but the data are usually sparse, which introduces considerable computation costs for feature selection and building models. [27], by Ding et al., proposes to represent the texts using word features, local context features, and web context features. In addition to the large and sparse data in word feature space, the web context features are generated online during the training process by querying Google and collecting titles and snippets, which could also introduce a significant amount of crawling and extracting computations and increase the feature representation dimension. A method to represent the texts in a space with low dimension is proposed in [10]. This method adopts the labeled sequential patterns as features and achieves both decent performance and efficiency.

2.3 Model Interpretability

In addition to classifying healthcare-related texts, our work focuses on both classifying and interpreting the online healthcare-related texts. The major challenges in healthcare-related text classification and interpretation are how to represent the texts and how to classify and interpret the data. In terms of modeling and enabling the interpretability, lasso [21] is proposed to enhance both the performance and the interpretability of regression models by tuning the parameter to shrink the features. Features with greater weights can be considered as more important, which enables the interpretability of the regression models. Tree-based and forest-based methods, e.g. CART [17] and random forest [9], are also widely utilized to handle classifying and interpreting the data using the decision rules in the trees.

3. EXTRACTING INTERPRETABLE FEATURES*

Interpretable features play an essential role in enabling users to understand prediction results. In this section, we discuss how to convert health-related sentences into instances in numerical feature space composed of labeled sequential patterns, UMLS semantic type features, sentence-based features, and heuristic features. The method of extracting labeled sequential patterns is introduced in detail.

3.1 Labeled Sequential Patterns

In sentences classification, if we simply use bag of words to represent each sentence, the overall data matrix will be huge and sparse, because there are a large number of terms, and many terms only occur in few sentences about some specific diseases. It is undesirable to use these raw terms to explain their correlations with sentence category as interpretations for classification results. The reason is that the raw terms do not explicitly specify the semantics of words, or contain the structural information of sentences. Therefore, we propose to use higher-level features to represent a sentence rather than words. We will rely on these higher-level features to interpret the sentences classification results.

3.1.1 Labeled Sequence Mapping

We first extract *labeled sequences* as preliminary representations of sentences [10]. A labeled sequence is in the form: *Sequence* $\rightarrow$ *Label*, where *Sequence* is a sequence of tags and *Label* is the class label. To convert a sentence into a sequence, we use the tags in Table 3.1 to replace the words in the sentence. Words with similar semantics are mapped to the same tag.

*Reprinted with permission from "An Interpretable Classification Framework for Information Extraction from Online Healthcare Forums" by Jun Gao et al. 2017. Journal of Healthcare Engineering, Volume 2017 (2017), Copyright 2017 by Jun Gao et al.

For example, the medication sentence *"I am taking 90 units Lantus twice a day"*, can be converted into a sequence of tag-word pairs *((PRP, "I"), (VBP, "am"), (VBG, "taking"), (CD, "90"), (NNS, "units"), (DRUG, "Lantus"), (FREQ, "twice a day"))* and the entire sentence is represented as a labeled sequence: *(PRP, VBP, VBG, CD, NNS, DRUG, FREQ)* $\rightarrow$ *Medication*.

Tag	Description
CC, CD, DT, EX, ...	Part-of-Speech Tags
DRUG	Medications or Drug Terms
SYMP	Symptom Terms
FREQ	Frequency Phrases

Table 3.1: Tags introduction

Given a training set of labeled sentences $\mathcal{S} = \{(s_1, l_1), \ldots, (s_n, l_n)\}$, we convert each pair into a labeled sequence $p_i \rightarrow l_i$ by applying the method mentioned above so that we can obtain the database $\mathcal{D}$ of labeled sequences. Our next goal is to mine the frequent patterns in the labeled sequences from $\mathcal{D}$ and adopt these frequent patterns as features to capture the characteristics of the healthcare-related sentences. This task can be divided into two steps: (1) frequent sequential pattern mining and (2) building frequent labeled sequential patterns.

3.1.2 Frequent Sequential Pattern Mining

We now focus on mining the frequent sequential patterns from database $\mathcal{D}$. Before that, we first define sequential pattern as below:

Definition 3.1.1. A *sequential pattern* is a sequence of tags which is a subsequence of one or more $Sequence$s in the database. The adjacent tags are not necessarily adjacent in the

original $Sequence$s, but their distance should be not greater than a threshold in the original $Sequence$s, which is set as 5 in experiments [10].

For example, given two labeled sequences $(a, b, c, d, e, f) \rightarrow l_1$ and $(a, c, d, e, g, h) \rightarrow l_2$ in the database $\mathcal{D}$, (a, c, e) can be considered as a sequential pattern of both $Sequence$s. Note that a $Sequence$ is different from a labeled sequence. The former only consists of the sequence of tags, while the latter includes the mapping from sequence to the label, i.e., $p_i \rightarrow l_i$.

Definition 3.1.2. A *frequent sequential pattern* (FSP) is a *sequential pattern* p' with $sup(p') \geq \mu$, where μ is a customized threshold, and $sup(p')$ denotes the support of p' in $\mathcal{D}$, i.e.,

$$sup(p') = \frac{|\{p|p \text{ contains } p', p \in \mathcal{D}\}|}{|\mathcal{D}|}, \tag{3.1}$$

where p is any $Sequence$ in the database $\mathcal{D}$ that contains p'. $sup(p')$ represents the percentage of the sequences in the database that contain p', which shows the generality of p' in the database $\mathcal{D}$.

There are several algorithms to mine frequent patterns from a database. We select CM-SPAM [11] to obtain FSPs from $\mathcal{D}$. The minimum threshold μ is customized by users such that the resultant FSPs would be general enough.

3.1.3 Frequent Labeled Sequential Patterns

With FSPs available, the next step is to select a subset of promising FSPs called frequent labeled sequential patterns (FLSPs) which are then used for classification.

Note that we have two classes *Medication* and *Symptom*, thus the FLSPs are different for each class. Formally, an FLSP of label l is defined as the FSP with high *confidence* with respect to l. Given a specific label l, the confidence of a frequent sequential pattern

p', denoted by $conf(p')$, is computed as:

$$conf(p') = \frac{|\{p|p \text{ contains } p', p \to l \in \mathcal{D}\}|}{|\{p|p \text{ contains } p', p \in \mathcal{D}\}|}, \tag{3.2}$$

which is the ratio of $Sequence$s that contain the FSP p' and are labeled l to the $Sequence$s containing the FSP p'. FSPs with high confidence show strong relations to the given label l, since a large portion of those frequent sequential patterns are labeled as l.

We would also like to set the minimum support threshold to a small percentage in order to include more FSPs. In our experiments, we set the minimum $support$ to 5%. Besides, the minimum confidence threshold might also not necessarily be set very large since we would like to obtain more FLSPs by reducing some predictive ability of them in the early stage. In the experiments, we set the minimum $confidence$ to 85% [10]. Algorithm 1 shows the entire process of generating FLSPs from text data.

Finally, we obtain a set of FLSPs, which can be used as features to identify the relationship between labels and patterns in sentences [12]. We use each frequent labeled sequential pattern as a feature. For each instance in the training set, if its mapped $Sequence$ contains a FLSP, we will set the value of the corresponding feature entry to 1; otherwise 0.

3.2 UMLS Metathesaurus Semantic Types

In addition to FLSPs, we also use UMLS [13] Metathesaurus semantic types as features. There are 133 UMLS Metathesaurus semantic types in total. By using the third-party software MetaMap [14], we can map the sentence to these semantic types. Thus, for each semantic type feature, we set the value to 1 if the sentence contains any phrases related to the semantic type; otherwise 0.

Generally, for each sentence s_i in $\mathcal{S}$, it is converted into $\mathbf{x_i}$ which is a representation vector of the sentence in the feature space of FLSPs and the UMLS semantic types. If s_i contains any FLSPs or phrases related to UMLS semantic types, the value of the corre-

Algorithm 1: Frequent Labeled Sequential Patterns Generation

Input : A collection of labeled sentences $\mathcal{S}$, a minimum support threshold sup, and a minimum confidence threshold $conf$

Output: A collection of FLSPs denoted as $\mathcal{P}$

Labeled sequence database $\mathcal{D} := \varnothing$;

for *each sentence sample* (s_i, l_i) *in* $\mathcal{S}$ **do**

  Convert s_i into a sequence p_i that consists of POS tags, *DRUG*, *SYMP*, and *FREQ*;

  $ls_i := p_i \rightarrow l$;

  $\mathcal{D} := \mathcal{D} \cup \{ls_i\}$;

end

FSP set $\mathcal{P}'$:= CM-SPAM($\mathcal{D}$, sup) [11];

FLSP set $\mathcal{P} := \varnothing$;

for *each FSP* p' *in* $\mathcal{P}'$ **do**

  if *conf*(p') $\geq conf$ **then**

    $\mathcal{P} := \mathcal{P} \cup \{p'\}$;

  end

end

return $\mathcal{P}$

sponding feature entry in $\mathbf{x_i}$ is set to 1.

3.3 Sentence-Based Features

Sentence-based features are capable of representing the sentence in a direct way [3]. In this paper, we use the following sentence-based features to represent sentences.

Word-Based Features

Although word-based features such as bag-of-word representation usually suffer from the curse of dimensionality, we still take them into account to compare the classification performance because of their effectiveness [15]. Unigrams and bigrams can capture those significant and frequent words or phrases related to a specific label. For example, it is likely that a sentence is classified into medication category if the word “prescribe" occurs. Each unigram or bigram corresponds a binary feature to indicate if a sentence contains this feature or not.

Morphological Features

Capitalized words and abbreviations can be good indicators of whether there are any medical terminologies in the sentence, which could be highly related to medication or symptom sentences. We can use two binary features to indicate whether the sentence contains any capitalized words or abbreviations, respectively.

3.4 Heuristic Features

In addition to all the features originated from the texts of the sentences, we can also adopt useful side information of posts [3]. Specifically, a sentence written by the thread creator is more likely to be symptom-related compared to the one written by the other users, because thread creators tend to ask for help from other users by posting their own conditions. Besides, the position of the post which a sentence is from can also indicate the category, because the first post written by the thread creators are usually describing the patients' situations, while the latter posts tend to answer the potential questions that arise in the first couple of posts. Thus, two binary features are considered to indicate whether a sentence is written by the thread creator, and the position of the post which the sentence is from, respectively.

In general, we can select different combinations of the features introduced in this section to represent health-related sentences, and then build models to predict the categories of sentences with interpretations.

4. INTERPRETABLE CLASSIFICATION WITH FOREST-BASED MODELS*

In this chapter, we first introduce the classification of health forum sentences using a random forest model, and how to interpret the forest model with features of high importance. Second, we introduce how to collect rules from decision trees in the forest to construct a new pattern space [7], and achieve the interpretability by selecting the top patterns.

4.1 Classification with Random Forests

A random forest consists of an ensemble of tree-based classifiers and calculates the votes from the trees for the most popular class in classification problems [9]. The growth of the ensemble is determined by the growth of each tree therein. The process of tree growth is introduced as follows [16]:

1. Sample N_T instances at random with replacement from the training set. The samples will then be used to grow the tree model.

2. A subset of m features are selected from the total D features at random, where $m \ll D$. The best split on the m features will be used to construct the tree nodes such that the Gini impurity for the descendants will be less than that of the parent node, using the method introduced in CART [17]. The value of m remains constant during the forest growing process.

3. Each tree grows to the maximum size without pruning.

When growing a tree using the samples from the original training set, about one-third

*Reprinted with permission from "An Interpretable Classification Framework for Information Extraction from Online Healthcare Forums" by Jun Gao et al. 2017. Journal of Healthcare Engineering, Volume 2017 (2017), Copyright 2017 by Jun Gao et al.

of the instances in the training set are left out of the samples selected at random. This out-of-bag data will be an unbiased estimate of the classification accuracy for the currently growing tree and also can be used to estimate features importance.

4.2 Interpretation with Discriminative Features

The classification mechanism of a random forest is explained through a set of decision paths. To interpret random forest models, we propose to quantify the contributions of node features, rank them according to their contributions, and find out the most discriminative ones [6][18].

For a decision tree in the random forest, its decision function can be formulated as below,

$$f(\mathbf{x}) = \sum_{m=1}^{M} c_m I(\mathbf{x}, R_m), \tag{4.1}$$

where M is the number of leaf nodes in the tree. c_m denotes the criterion score, which is a scalar in regression problems or a vector in classification problems, learned from the training process. $\mathbf{x}$ is the input sample. R_m is the path from the root to the m-th leaf node. $I(\cdot, \cdot)$ is an indicator function identifying whether x is run through R_m. As we are solving a classification problem, c_m and $f(\mathbf{x})$ should be vectors whose sizes are the number of the classes. The i-th value in the vector $f(\mathbf{x})$ represents the criterion score of the instance $\mathbf{x}$ being classified into the i-th class, which can be converted to a probability value by normalization. In our problem of classification, an input instance $\mathbf{x}$ is classified into one class from the classes {*Medication*, *Symptom*, *Background*} according to the maximum probability specified by $f(\mathbf{x})$ of the decision tree.

From another perspective, we can observe how a feature contributes to the *criterion score* (i.e., Gini impurity or entropy) vector by calculating the score vector difference between the current node and the next node in the path. The final prediction result along a tree path is determined under the cumulative influences of nodes in the path. Therefore, a

prediction can be defined as a sum of feature contributions plus a bias:

$$f_t(\mathbf{x}) = \sum_{k=1}^{D} FC_{t,k}(\mathbf{x}) + \beta_t, \tag{4.2}$$

where $FC_{t,k}(\mathbf{x})$ is the feature contribution vector from the k-th feature in the t-th tree for an input vector $\mathbf{x}$, D is the number of features, and β_t is the bias of tree t. Both $FC_{t,k}(\mathbf{x})$ and β_t are criterion score vectors. Our goal is to calculate the feature contributions for an instance $\mathbf{x}$ classified by a decision tree t that has been trained on the training set. Specifically, it is achieved by running through the decision paths in tree t. On the root node in the path, $FC_{t,k}(\mathbf{x}) = 0$ and $f_t(\mathbf{x})$ is initialized to β_t. Each time the instance arrives at a node with a decision branch on the r-th feature, and $FC_{t,r}(\mathbf{x})$ will be incremented by the difference between the criterion scores at the child node along the path and the current node. Once the decision process of $\mathbf{x}$ reaches a leaf node, we assign a class to $\mathbf{x}$ and obtain all feature contributions along the decision path.

The prediction function of a forest, which is an ensemble of decision trees, takes the average of the predictions of its trees:

$$F(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} f_t(\mathbf{x}), \tag{4.3}$$

where T is the number of trees in the forest. Similarly, the prediction function of a forest can also be decomposed with respect to feature contributions:

$$\begin{aligned} F(\mathbf{x}) &= \frac{1}{T} \sum_{t=1}^{T} \left(\sum_{k=1}^{D} FC_{t,k}(\mathbf{x}) + \beta_t \right) \\ &= \sum_{k=1}^{D} \left(\frac{1}{T} \sum_{t=1}^{T} FC_{t,k}(\mathbf{x}) \right) + \frac{1}{T} \sum_{t=1}^{T} \beta_t, \end{aligned} \tag{4.4}$$

where $FC_{t,k}$ is the contribution of the k-th feature in the t-th tree. Therefore, the contri-

bution of the k-th feature to classify an instance $\mathbf{x}$ can be defined as

$$\overline{FC}_k(\mathbf{x}) = \frac{1}{T}\sum_{t=1}^{T} FC_{t,k}(\mathbf{x}), \tag{4.5}$$

and the bias of the forest $\bar{\beta} = \frac{1}{T}\sum_{t=1}^{T}\beta_t$. The idea of interpreting the random forest model, which classifies sentences into *Medication* or *Symptom* category, is to find out those features with the most contribution to leading an instance to *Medication* or *Symptom* leaf nodes. We will not interpret *Background* sentences since they are not as informative as the other two classes.

Suppose a random forest model $F(\mathbf{x})$ is constructed given the training set $\mathcal{X} = \{(\mathbf{x}_i, l_i)|1 \leq i \leq N\}$ with N labeled instances. To find out the important features for category *Medication* and *Symptom*, we select two subsets of training sets whose labels are *Medication* and *Symptom*, respectively. Let $\mathcal{X}_M = \{(\mathbf{x}, l)|(\mathbf{x}, l) \in \mathcal{X}, l = Medication\}$ be the subset of Medication instances, and $\mathcal{X}_S = \{(\mathbf{x}, l)|(\mathbf{x}, l) \in \mathcal{X}, l = Symptom\}$ be the subset of Symptom instances, the average feature contributions for the two classes can be calculated as follows:

$$\overline{FC}_{M,k} = \frac{1}{|\mathcal{X}_M|}\sum_{(\mathbf{x},l)\in\mathcal{X}_M} \overline{FC}_k(\mathbf{x}), \tag{4.6}$$

$$\overline{FC}_{S,k} = \frac{1}{|\mathcal{X}_S|}\sum_{(\mathbf{x},l)\in\mathcal{X}_S} \overline{FC}_k(\mathbf{x}), \tag{4.7}$$

where $\overline{FC}_{M,k}$ and $\overline{FC}_{S,k}$ are the positive contribution vectors of the k-th feature for Medication class and Symptom class, respectively. After computing the contribution of features for each class, we rank these features to indicate their relative significance. Finally, the ones with larger contribution are selected as the discriminative features of each class.

4.3 Interpretation with Discriminative Patterns

To further exploit interpretability, we extract decision rules from the forest model to form a new space, where the forward selection is applied to select the top discriminative decision rule combinations, i.e., discriminative patterns [7].

Specifically, a *pattern* is defined as the form of

$$(x_{i,j_1} \leq v_{j_1}) \wedge (x_{i,j_2} > v_{j_2}) \wedge \cdots \wedge (x_{i,j_k} \leq v_{j_k}), \tag{4.8}$$

where $x_{i,j}$ is the value on feature j of instance $\mathbf{x_i}$, and v_j is a scalar threshold. In our problem, a pattern can be any combination of rules from a decision tree. Furthermore, *discriminative patterns* (DPs) are those strong signaling patterns with high information gain or low Gini impurity in classification. In our problem, a pattern refers to a complete decision path, and a discriminative pattern is the path with low Gini impurity.

However, since the dimension $|DP|$ of discriminative patterns is still high, we need to identify the most informative ones from them. To this end, we apply forward selection [19] to select the top K discriminative patterns. Let f_s be the forward selection function, then we have $f_s : \{0,1\}^{|DP|} \rightarrow \{0,1\}^K$. We run K iterations, where the DP set at iteration k is denoted as Pat_k. At iteration I, we traverse the discriminative patterns $dp_j \notin Pat_{I-1}$. A temporary DP set Pat_{Ij} at current iteration is built by adding dp_j to the DP set obtained in iteration $I-1$, i.e.,

$$Pat_{Ij} = Pat_{I-1} \bigcup \{dp_j\}. \tag{4.9}$$

Then we build a classifier using support vector machines [20] based on the selected patterns Pat_{Ij}, and obtain the accuracy acc_{Ij} of the classifier. The best pattern dp_{j^*} is added into the DP set, where $j^* = arg\,max_j Pat_{Ij}$ and $acc_{Ij^*} > acc_{I-1}$, so that $Pat_I = Pat_{Ij^*}$. After K iterations, we obtain the top K discriminative patterns Pat_K. At last, each in-

stance **x** in the dataset is mapped to the DP space as $\mathbf{y} \in \{0,1\}^K$. If the k_{th} pattern appears in **x**, then the corresponding entry $\mathbf{y}_k$ is set to 1; otherwise 0.

5. EXPERIMENT RESULTS AND DISCUSSIONS*

In this chapter, first we present the experiments results which show that the forest-based models outperform the baseline methods. Second, we compare the interpretability between Lasso and our forest-based model by analyzing their discriminative features and discriminative patterns.

5.1 Experimental Setup

5.1.1 Dataset

Since there are few datasets available for health-related texts classification, we created our dataset by collecting texts from online health communities to solve this problem. The data used for the experiment in this study were crawled from Patient.info using Scrapy, a python framework. The ground truth was obtained by assigning a label to each sentence in the data set. 257187 discussions in 616 sub-forums from the forum were crawled. Then we used NLTK tokenize package to split the texts in each discussion into a list of sentences. Given lists of sentences from all the discussions, we randomly select sentences from each list in portion and the number of selected sentences is 2585. We recruited two volunteers to complete the labeling work. Both volunteers were provided with the total 2585 randomly selected sentences and asked to categorize each of the sentences into Medication, Symptom, or Others. The labeled sentences were merged based on unanimous voting. We discarded the sentences that were labeled with disagreements and obtained 2099 sentences categorized into the same label. The result of the sentences labeling is in Table 5.1. In the experiments, we set the label of class *Background*, *Medication*, and

*Reprinted with permission from “An Interpretable Classification Framework for Information Extraction from Online Healthcare Forums" by Jun Gao et al. 2017. Journal of Healthcare Engineering, Volume 2017 (2017), Copyright 2017 by Jun Gao et al.

Symptom to 0, 1, and 2, respectively.

Med.	Symp.	Others	Total
1127	772	200	2099

Table 5.1: Labeled sentences

5.1.2 Baseline Methods

The contributions of our study we want to claim are how much improvement of the performance our proposed method can achieve by introducing the labeled sequential patterns as features and how the interpretability can be enabled by applying our proposed methods to sentence representatives in a variety of spaces to gain an insight of the health-related text classification model. To show the first contribution, we choose support vector machines trained on a variety of features proposed in [3]. We built binary classification SVM models for class *Medication* and *Symptom* with RBF kernel $\exp(-\gamma|x-x'|^2)$, where γ is the reciprocal of the number of features. To predict an instance, the SVM models calculate the probabilities using Platt scaling. If the probabilities to classify the instance into *Medication* and *Symptom* are both less than 0.5, then we classify the instance into class *Background*; otherwise, it is classified into the class with greater probability. In order to ensure the performance, we implement feature selection based on entropy using a decision tree model. In terms of the second contribution, we compare the model interpretability between lasso [21] to random forests and DPClass and interpret the models using the features with non-zero weights in lasso with L1 term coefficient set to 0.001.

5.1.3 Evaluation Metrics

The metrics for the evaluation are accuracy, weighted average precision, weighted average recall, and weighted average F_1 score. For multi-class classification, the weighted average precision, recall, and F_1 score can be computed as follows:

$$Precision = \frac{1}{N} \sum_{l \in L} N_l Precision_l,$$

$$Recall = \frac{1}{N} \sum_{l \in L} N_l Recall_l,$$

$$F_1 = \frac{1}{N} \sum_{l \in L} N_l F_{1l},$$

where N is the size of the test set, L is the label set, i.e. $L = \{$*Medication*, *Symptom*, *Background*$\}$, N_l is the size of the test subset with label l, and $Precision_l$, $Recall_l$, and $F1_l$ are the precision, the recall, and the F_1 score of the binary classification for instances with label l.

5.2 Classification Performance Evaluation

Table 5.2, Table 5.3, and Table 5.4 show the evaluation of each model using 5-fold cross validation. Each row represents the evaluation results of a model trained on data in different feature spaces. Each type of features used for training models are the ones selected with entropy-based methods, so that they are more informative and more discriminative in classification. For each model, the average accuracy (Acc), weighted average precision (Prec), recall (Rec), and F-score (F1) for Medication class (M), Symptom class (S), and the overall classes are presented respectively.

For the SVM model, the entire average predicting accuracy achieves 79.8% with only word-based features, which outperforms the accuracies of Lasso. SVM also performs

very well in terms of precision, recall, and F1 score. The model trained on LSP features alone fails to outperform the model trained on word-based features, but the former could achieve better performance than the latter if we add the UMLS semantic type features. Note that there are only hundreds of LSP features while there are more than 16k word-based ones. Without feature selection, the performance of SVM is not very good, since the word-based features are considerably sparse. Furthermore, SVMs with RBF kernels do not provide interpretability directly for us to gain an insight of the sentences although the models achieve good performance.

From the experiments results using Lasso, we can find that the recall scores for classifying medication sentences are better than those for symptom ones, while the accuracies and precision scores indicate the opposite trend. As we use multi-class classifiers, many of the test instances are classified as medication class. The Lasso models trained on the word-based features slightly outperform the ones trained on the LSP features. As Table 5.6 shows, the weights of the LSP features are much smaller than those of the word-based ones in Lasso.

For the forest-based model, we can find that the accuracies of Medication and Symptom class can both achieve more than 80% with only LSP features and UMLS semantic type features. The overall accuracy achieves 80.9% and outperforms the other methods. Besides, with LSP and UMLS semantic type features, the precisions and recalls of both classes are greater 0.8. Moreover, with position feature and word-based features, the performance of the forest-based model is even better. In general, the random forest model can achieve the relatively better F1 scores for both medication and symptom sentences classification. Similarly, the random forest models trained on the word-based features slightly outperform those trained on the LSP features.

Although it is not guaranteed that the models trained on LSP features outperform the ones trained on word-based features, we would still like to take advantage of LSP features

since the feature dimension is significantly reduced without sacrificing the discrimination ability of models. In addition, LSP features provide a valuable perspective in both tag and structural levels to interpret classification results for health-related sentences.

5.3 Interpretability Evaluation

5.3.1 Interpretability of Lasso

Table 5.5, Table 5.6, and Table 5.7 list the features with the largest weights in the combination of the word-based features, LSP features, UMLS Metathesaurus semantic type features, position feature, thread creator indicator feature, and word count features. After learning process, medication-related features are assigned negative weights, while potential symptom-related features are assigned positive weights. Meanwhile, most of the word-based features have greater weights than the other features. The words “avoid", “prescribe", and “increase" are the most signaling words in medication sentences. The possible reason behind might be that medications usually require patients to avoid certain things, to take prescription drugs, or to adjust the dosages. The words such as “bleeding", “anxiety", “swelling", “migraines", and “fever" are common for symptom sentences in the forum, as they express external physical injury and mental diseases.

For LSPs, they are usually assigned with positive weights as they are capable of mining the symptom terms in the sentences. The pattern *(PRP, PRP, RB, SYMP)*, for example, is common for symptom-related sentences like “someone suffers from some symptom frequently/occasionally." However, we also find that the tag *SYMP* is very frequent in both medication and symptom sentences, which is due to the reason that Lasso could not achieve good performance using LSP features, and is also hard to interpret the differences between class *Medication* and class *Symptom*.

Several UMLS semantic type features are assigned relatively larger weights to identify symptom sentences. For examples, the term “sosy", short for “Sign or Symptom", is

obviously a useful feature to identify symptom sentences. The term “mobd” (i.e., “Mental or Behavioral Dysfunction") can be used to detect mental disease symptoms. “patf" (i.e., “Pathologic Function") is a parent semantic type of “mobd", which is also an informative feature to detect pathologic terms.

5.3.2 Interpretability of Forest-Based Model

To interpret healthcare-related sentences in forest-based models, we calculate the feature contributions from decision trees in the forest. We select one random forest model with the best accuracies in the experiments and list the 10 features with the greatest contributions for each class in Table 5.8 and Table 5.9.

In identifying *Medication* sentences, the unigram feature “prescribed" has the largest contribution. This is because such kind of sentences usually contain information about prescribing drugs. LSP features *(PRP, CD, CD), (PRP, CD, IN, NN, NN), (CD, IN, CD, CD),* and *(PRP, CD, JJ, JJ)* also contribute to recognizing sentences as medication-related ones as they all contain the POS tag *CD*, which represents the numbers in describing the dosages of medications. The morphological features are selected as the names of many drugs are capitalized terms or abbreviations. The UMLS semantic type feature “hlca" (i.e., “Health Care Activity") is important since healthcare activity terms are commonly seen in medication sentences. On the contrary, if a sentence does not contain LSP *(NN, SYMP, SYMP, CC)* or “sosy" (“Sign or Symptom"), or is not posted by the user (thr. crt. = 0), this sentence may also be classified into medication class, as it is less likely to be symptom-related.

For the *Symptom* class, the UMLS semantic type features “sosy" and “patf" are among the top relevant ones since they are capable of detecting symptom terms and pathologic terms, respectively. Thread Creator indicator is also useful since symptom sentences are mainly posted by users to share their situations and ask for more information. If a sentence

does not contain the word “prescribed", then it is less likely to be medication-related. LSP features *(SYMP, SYMP, SYMP), (NN, SYMP, SYMP, CC),* and *(SYMP, CC, JJ)* are selected since there are usually multiple terms matching the tag *SYMP* in symptom sentences. The position feature is also important in identifying Symptom class, as it is natural for users to mention their symptoms in the first v_{th1} posts, where v_{th1} is a threshold learned by the decision tree. Similarly, if the number of words from a sentence is greater than v_{th2} learned by the decision tree, the sentence will be more likely to be a symptom sentence.

Compared to the feature ranking in Lasso, we can have a better understanding from the feature contribution rankings for each class in the random forest. The relationships between features and classes can be learned from the feature contribution vectors while Lasso only provides the weights of the features, which may not be expressive enough to represent the relationships between features and classes. The random forest model can achieve both better performance and interpretability compared to Lasso.

DPClass [7] proposes to take further advantages of the discriminative patterns in a random forest built on the training set. The selected DPs can help users gain insights of the data. In the experiments, we chose $K = 30$ to obtain the top 30 DPs. Table 5.10 lists the selected 10 DPs of a forest-based model trained on all proposed features. For example, considering the discriminative pattern ((RB,CD,IN,IN)=0) $\cap$ ((VBP,IN,CD ,CD,NN)=0) $\cap$ (“mg"=0) $\cap$ (“prescribed"=1) $\cap$ (dsyn=0), if an instance satisfies each rule in the pattern, its corresponding DP feature entry will be set to 1. The existence of this pattern increase the likelihood of classifying the instance into Medication class in the decision tree. From the patterns in the table, we can find that the terms matching tag $SYMP$ are likely to occur in symptom sentences, while tag CD and $DRUG$ often lead to non-Symptom leaves as they are more likely to occur in medication sentences. In another word, symptom sentences usually contain symptom terms while medication sentences usually contain drug terms and numbers which represent the dosages of the medications. In addition to LSP features,

there are two conspicuous unigram patterns “anxiety" and “cough", because the training set contains many sentences related to anxiety and cough conditions.

	Ft. Set	M. Acc.	M. Prec.	M. Rec.	M. F1.
Select+SVM	Word-Based	0.843	0.846	0.867	0.856
	+ Semantic	**0.851**	0.854	**0.871**	**0.862**
	+ Position	0.843	0.846	0.867	0.856
	+ Thr. Crt.	0.844	0.846	0.867	0.857
	+ Morpho.	0.848	0.855	0.864	0.859
	+ Word Cnt.	0.802	0.785	**0.871**	0.826
	LSP	0.799	**0.894**	0.709	0.790
	+ Semantic	0.849	0.865	0.852	0.858
	+ Position	0.841	0.851	0.852	0.851
	+ Thr. Crt.	0.844	0.852	0.859	0.855
	+ Morpho.	**0.851**	0.860	0.864	0.861
	+ Word Cnt.	0.848	0.856	0.862	0.859
	+ Word-Based	0.810	0.810	0.844	0.826
Lasso	Word-Based	0.794	0.730	**0.979**	**0.837**
	+ Semantic	0.793	0.741	0.947	0.831
	+ Position	0.795	0.742	0.947	0.832
	+ Thr. Crt.	0.796	0.745	0.945	0.833
	+ Morpho.	0.797	0.745	0.947	0.834
	+ Word Cnt.	0.798	**0.746**	0.947	0.834
	LSP	0.715	0.663	0.955	0.782
	+ Semantic	0.769	0.712	0.955	0.816
	+ Position	0.767	0.710	0.955	0.814
	+ Thr. Crt.	0.771	0.715	0.953	0.817
	+ Morpho.	0.771	0.715	0.953	0.817
	+ Word Cnt.	0.771	0.715	0.953	0.817
	+ Word-Based	**0.799**	0.745	0.950	0.835
Forest-Based	Word-Based	**0.848**	0.795	**0.969**	**0.873**
	+ Semantic	0.815	0.761	0.956	0.847
	+ Position	0.820	0.767	0.957	0.851
	+ Thr. Crt.	0.817	0.765	0.949	0.847
	+ Morpho.	0.832	0.776	0.965	0.860
	+ Word Cnt.	0.830	0.779	0.954	0.858
	LSP	0.786	0.742	0.921	0.822
	+ Semantic	0.837	0.824	0.887	0.854
	+ Position	0.840	**0.836**	0.873	0.854
	+ Thr. Crt.	0.832	0.825	0.875	0.849
	+ Morpho.	0.841	0.829	0.886	0.856
	+ Word Cnt.	0.829	0.816	0.881	0.847
	+ Word-Based	0.848	0.816	0.927	0.868

Table 5.2: Model evaluation. We evaluate each model using 5-fold cross validation. Each of the average accuracy, weighted average precision, weighted average recall, and weighted average F-score for **Medication class performance** is presented in each column. Each row represents the performance of each model trained on different feature combinations.

	Ft. Set	S. Acc.	S. Prec.	S. Rec.	S. F1.
Select+SVM	Word-Based	0.886	0.875	0.804	0.838
	+ Semantic	0.884	0.874	0.801	0.836
	+ Position	0.886	0.875	0.805	0.838
	+ Thr. Crt.	0.896	**0.894**	0.814	0.852
	+ Morpho.	0.891	0.883	0.811	0.846
	+ Word Cnt.	0.864	0.888	0.722	0.796
	LSP	0.831	0.862	0.644	0.737
	+ Semantic	0.891	0.878	0.818	0.846
	+ Position	0.893	0.883	0.817	0.848
	+ Thr. Crt.	**0.897**	0.885	0.826	0.855
	+ Morpho.	0.896	0.883	0.826	0.854
	+ Word Cnt.	**0.897**	0.884	**0.830**	**0.856**
	+ Word-Based	0.870	0.887	0.739	0.806
Lasso	Word-Based	0.886	**0.969**	0.712	0.820
	+ Semantic	0.886	0.923	0.752	0.828
	+ Position	0.886	0.920	0.754	0.829
	+ Thr. Crt.	0.889	0.922	0.762	0.834
	+ Morpho.	0.889	0.924	0.759	0.833
	+ Word Cnt.	0.891	0.927	0.762	0.836
	LSP	0.802	0.875	0.538	0.666
	+ Semantic	0.861	0.911	0.689	0.785
	+ Position	0.860	0.910	0.686	0.782
	+ Thr. Crt.	0.864	0.911	0.700	0.791
	+ Morpho.	0.864	0.910	0.698	0.790
	+ Word Cnt.	0.864	0.910	0.698	0.790
	+ Word-Based	**0.893**	0.930	**0.765**	**0.839**
Forest-Based	Word-Based	0.881	0.891	0.773	0.827
	+ Semantic	0.878	0.901	0.751	0.819
	+ Position	0.887	**0.908**	0.772	0.833
	+ Thr. Crt.	0.872	0.884	0.749	0.811
	+ Morpho.	0.890	0.907	0.781	0.838
	+ Word Cnt.	**0.893**	0.893	0.804	**0.846**
	LSP	0.863	0.861	0.748	0.801
	+ Semantic	0.879	0.860	0.802	0.829
	+ Position	0.882	0.844	**0.834**	0.839
	+ Thr. Crt.	0.879	0.849	0.814	0.831
	+ Morpho.	0.881	0.843	0.832	0.837
	+ Word Cnt.	0.880	0.856	0.808	0.831
	+ Word-Based	0.887	0.861	0.827	0.843

Table 5.3: Model evaluation. We evaluate each model using 5-fold cross validation. Each of the average accuracy, weighted average precision, weighted average recall, and weighted average F-score for **Symptom class performance** is presented in each column. Each row represents the performance of each model trained on different feature combinations.

	Ft. Set	Acc.	Prec.	Rec.	F1.
Select+SVM	Word-Based	0.798	0.808	0.798	0.802
	+ Semantic	0.804	0.816	0.804	0.808
	+ Position	0.798	0.808	0.798	0.802
	+ Thr. Crt.	0.800	0.812	0.800	0.805
	+ Morpho.	0.801	0.816	0.801	0.807
	+ Word Cnt.	0.761	0.773	0.761	0.763
	LSP	0.691	0.821	0.691	0.731
	+ Semantic	0.806	**0.823**	0.806	**0.813**
	+ Position	0.800	0.815	0.800	0.806
	+ Thr. Crt.	0.801	0.814	0.801	0.807
	+ Morpho.	**0.808**	0.820	**0.808**	**0.813**
	+ Word Cnt.	0.807	0.819	0.807	0.812
	+ Word-Based	0.768	0.792	0.768	0.776
Lasso	Word-Based	0.791	**0.785**	0.791	0.756
	+ Semantic	0.789	0.754	0.789	0.757
	+ Position	0.790	0.757	0.790	0.758
	+ Thr. Crt.	0.791	0.756	0.791	0.759
	+ Morpho.	0.792	0.757	0.792	0.760
	+ Word Cnt.	0.793	0.759	0.793	0.762
	LSP	0.711	0.678	0.711	0.665
	+ Semantic	0.767	0.727	0.767	0.728
	+ Position	0.765	0.716	0.765	0.725
	+ Thr. Crt.	0.769	0.728	0.769	0.731
	+ Morpho.	0.769	0.728	0.769	0.730
	+ Word Cnt.	0.769	0.728	0.769	0.730
	+ Word-Based	**0.795**	0.759	**0.795**	**0.763**
Forest-Based	Word-Based	0.819	0.808	0.819	0.795
	+ Semantic	0.802	0.805	0.802	0.778
	+ Position	0.807	0.791	0.807	0.779
	+ Thr. Crt.	0.799	0.792	0.799	0.774
	+ Morpho.	0.816	**0.815**	0.816	0.789
	+ Word Cnt.	0.814	0.797	0.814	0.783
	LSP	0.771	0.725	0.771	0.739
	+ Semantic	0.809	0.805	0.809	**0.805**
	+ Position	0.808	0.800	0.808	0.803
	+ Thr. Crt.	0.802	0.796	0.802	0.797
	+ Morpho.	0.812	0.802	0.812	0.804
	+ Word Cnt.	0.800	0.791	0.800	0.793
	+ Word-Based	**0.821**	0.803	**0.821**	0.802

Table 5.4: Model evaluation. We evaluate each model using 5-fold cross validation. Each of the average accuracy, weighted average precision, weighted average recall, and weighted average F-score for **overall performance** is presented in each column. Each row represents the performance of each model trained on different feature combinations.

Word-Based	Average Weight
avoiding	-0.413
wrong	-0.363
avoid	-0.343
prescribe	-0.323
bleeding	0.283
anxiety	0.281
swelling	0.233
increased	-0.185
migraines	0.185
fever	0.160

Table 5.5: Top 10 average weight of word-based features in Lasso

LSP	Average Weight
(PRP, PRP, RB, SYMP)	0.081
(PRP, PRP, VB, SYMP)	0.060
(PRP, PRP, VB, SYMP)	0.060
(VBZ, CC, SYMP)	0.058
(SYMP, SYMP, SYMP)	0.054
(PRP, SYMP, CC, SYMP, IN)	-0.053
(CC, SYMP, IN, SYMP)	-0.052
(PRP, SYMP, VBG)	0.049
(RB, SYMP, VB)	0.048
(JJ, IN, JJ, SYMP)	0.036
(NN, SYMP, RB, SYMP)	-0.033

Table 5.6: Top 10 average weight of LSP features in Lasso

Semantic	Average Weight
sosy	0.329
mobd	0.207
patf	0.190
resa	-0.173
inpo	0.100
anab	0.094
mcha	-0.092
aggp	-0.090
plnt	-0.063
mamm	-0.052

Table 5.7: Top 10 average weight of semantic features in Lasso

Top 10 FC for Medication Sentences			
Feature	Back.	Med.	Sym.
prescribed = 1	-0.00275	0.01195	-0.00920
(PRP, CD, CD) = 1	-0.00251	0.01156	-0.00905
morpho. = 1	-0.00206	0.00660	-0.00455
hlca = 1	-0.00071	0.00559	-0.00489
(NN, SYMP, SYMP, CC) = 0	0.00115	0.00429	-0.00544
sosy = 0	0.00191	0.00406	-0.00597
(PRP, CD, IN, NN, NN) = 1	-0.00075	0.00402	-0.00327
(CD, IN, CD, CD) = 1	-0.00120	0.00396	-0.00276
thr. crt. = 0	0.00154	0.00381	-0.00535
(PRP, CD, JJ, JJ) = 1	-0.00086	0.00362	-0.00276

Table 5.8: Top 10 feature contributions for **medication class** in a random forest model

Top 10 FC for Symptom Sentences			
Feature	Back.	Med.	Sym.
sosy = 1	-0.00589	-0.00783	0.01371
prescribed = 0	0.00234	-0.015734	0.01339
thr. crt. = 1	-0.00381	-0.00683	0.01064
(PRP, CD, CD) = 0	0.00271	-0.01264	0.00993
(SYMP, SYMP, SYMP) = 1	-0.00330	-0.00564	0.00895
(NN, SYMP, SYMP, CC) = 1	-0.00209	-0.00667	0.00876
position $\leq v_{th1}$	-0.00334	-0.00540	0.00874
patf = 1	-0.00254	-0.00379	0.00633
(SYMP, CC, JJ) = 1	-0.00172	-0.00404	0.00576
word count $> v_{th2}$	-0.00131	-0.00423	0.00554

Table 5.9: Top 10 feature contributions for **symptom class** in a random forest model

Pattern	Leaf Class
((RB,CD,CD)=0)∩((PRP,CD,CD,JJ)=0)∩ ((PRP,CD,NN,NN,NN)=0)∩((TO,VB,CD)=1)	Med.
((IN,NN,NN,comma,SYMP)=0)∩((CD,RB,CD)=1)∩ ((RB,IN,IN,CD,IN)=0)∩((PRP,CC,CD,NN)=1)	Med.
((SYMP,NN,VBG)=1)	Sym.
((VBP,CD,NN,NN)=0)∩((SYMP,SYMP,NN)=1)	Sym.
((RB,CD,IN,IN)=0)∩((VBP,IN,CD,CD,NN)=0)∩ (“mg"=0)∩(“prescribed"=1)∩(dsyn=1)	Med.
((PRP,VBP,CD)=0)∩((CD,CD,NN,NN)=1)∩ ((TO,CD,IN)=1)	Med.
(“cough"=1)	Sym.
((RB,CD,IN,IN)=0)∩((VBP,IN,CD,CD,NN)=0)∩(“mg"=0)∩ (“prescribed"=0)∩(fndg=0)∩((NN,comma,comma,SYMP)=1)	Sym.
((RB,CD,IN,IN)=0)∩((VBP,IN,CD,CD,NN)=0)∩ (“mg"=0)∩(“prescribed"=1)∩(dsyn=0)	Med.
(“anxiety"=1)	Sym.

Table 5.10: Top 10 discriminative patterns in a DPClass model

6. CONCLUSIONS AND FUTURE WORK*

In our research, we propose to use labeled sequential patterns to represent the healthcare-related sentences in order to reduce the dimension and sparsity of the data, which can both guarantee the performance and enhance the efficiency. Then we build forest-based models on the training data which is capable of predicting with decent performance and interpreting the healthcare-related sentences by extracting the important features used in the decision rules, ranked by their contributions, and the discriminative patterns consist of the decision rules. Overall, the forest-based models trained on the proposed feature space can achieve good performance and enable the interpretability of the data. In the future, we will build a compact system based on this framework to help users directly extract and highlight the insightful sentences while they are viewing healthcare-related articles, posts, etc. Also, in addition to OHF posts, we will also apply our framework to real-world clinical notes. Moreover, we will also target to extract and interpret the insightful sentences from other categories such as medication effects, user questions, etc., and include data from other sources like clinical notes.

*Reprinted with permission from "An Interpretable Classification Framework for Information Extraction from Online Healthcare Forums" by Jun Gao et al. 2017. Journal of Healthcare Engineering, Volume 2017 (2017), Copyright 2017 by Jun Gao et al.

REFERENCES

[1] PwC, "Social media ?likes? healthcare from marketing to social business," 2012.

[2] X. Hu, N. Sun, C. Zhang, and T.-S. Chua, "Exploiting internal and external semantics for the clustering of short texts using world knowledge," in *Proceedings of the 18th ACM conference on Information and knowledge management*, pp. 919–928, ACM, 2009.

[3] P. Sondhi, M. Gupta, C. Zhai, and J. Hockenmaier, "Shallow information extraction from medical forum data," in *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pp. 1158–1166, Association for Computational Linguistics, 2010.

[4] Z. Che, S. Purushotham, R. Khemani, and Y. Liu, "Distilling knowledge from deep networks with applications to healthcare domain," *arXiv preprint arXiv:1512.03542*, 2015.

[5] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "Retain: An interpretable predictive model for healthcare using reverse time attention mechanism," in *Advances in Neural Information Processing Systems*, pp. 3504–3512, 2016.

[6] A. Palczewska, J. Palczewski, R. M. Robinson, and D. Neagu, "Interpreting random forest classification models using a feature contribution method," in *Integration of reusable systems*, pp. 193–218, Springer, 2014.

[7] J. Shang, W. Tong, J. Peng, and J. Han, "Dpclass: An effective but concise discriminative patterns-based classification framework," 2016.

[8] N. Liu, X. Huang, and X. Hu, "Accelerated local anomaly detection via resolving attributed networks," in *Proceedings of the 26th International Joint Conference on*

Artificial Intelligence, IJCAI/AAAI, 2017.

[9] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[10] G. Cong, L. Wang, C.-Y. Lin, Y.-I. Song, and Y. Sun, "Finding question-answer pairs from online forums," in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 467–474, ACM, 2008.

[11] P. Fournier-Viger, A. Gomariz, M. Campos, and R. Thomas, "Fast vertical mining of sequential patterns using co-occurrence information," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 40–52, Springer, 2014.

[12] N. Jindal and B. Liu, "Identifying comparative sentences in text documents," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 244–251, ACM, 2006.

[13] O. Bodenreider, "The unified medical language system (umls): integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl 1, pp. D267–D270, 2004.

[14] A. R. Aronson, "Effective mapping of biomedical text to the umls metathesaurus: the metamap program.," in *Proceedings of the AMIA Symposium*, p. 17, American Medical Informatics Association, 2001.

[15] X. Hu and H. Liu, "Text analytics in social media," in *Mining text data*, pp. 385–414, Springer, 2012.

[16] L. Breiman, "Random forests leo breiman and adele cutler," *Random Forests-Classification Description*, 2015.

[17] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[18] A. Saabas, "Interpreting random forests." http://blog.datadive.net/interpreting-random-forests/, 2014.

[19] S. Derksen and H. Keselman, "Backward, forward and stepwise automated subset selection algorithms: Frequency of obtaining authentic and noise variables," *British Journal of Mathematical and Statistical Psychology*, vol. 45, no. 2, pp. 265–282, 1992.

[20] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[21] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288, 1996.

[22] J. Patrick and M. Li, "High accuracy information extraction of medication information from clinical notes: 2009 i2b2 medication extraction challenge," *Journal of the American Medical Informatics Association*, vol. 17, no. 5, pp. 524–527, 2010.

[23] E. Sirohi and P. Peissig, "Study of effect of drug lexicons on medication extraction from electronic medical records.," in *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pp. 308–318, 2004.

[24] H. Xu, S. P. Stenner, S. Doan, K. B. Johnson, L. R. Waitman, and J. C. Denny, "Medex: a medication information extraction system for clinical narratives," *Journal of the American Medical Informatics Association*, vol. 17, no. 1, pp. 19–24, 2010.

[25] S. Wang, Y. Li, D. Ferguson, and C. Zhai, "Sideeffectptm: An unsupervised topic model to mine adverse drug reactions from health forums," in *Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics*, pp. 321–330, ACM, 2014.

[26] J. Bian, U. Topaloglu, and F. Yu, "Towards large-scale twitter mining for drug-related adverse events," in *Proceedings of the 2012 international workshop on Smart health and wellbeing*, pp. 25–32, ACM, 2012.

[27] H. Ding and E. Riloff, "Extracting information about medication use from veterinary discussions," in *ACL*, 2015.