An Adaptive Semi-supervised Learning Approach to Automatic Abbreviation Identification

VO THI NGOC CHAU, University of Technology, Vietnam National University at Ho Chi Minh City, Vietnam
CAO HOANG TRU, University of Technology, Vietnam National University at Ho Chi Minh City, Vietnam
HO TU BAO, Japan Advanced Institute of Science and Technology, Japan, John von Neumann Institute, Vietnam National University at Ho Chi Minh City, Vietnam

Abbreviations in clinical notes of the electronic medical records might hinder users in their efforts to understand and consume those records. They further greatly affect all automatic computer-based data processing tasks. In this paper, we propose a solution to the automatic abbreviation identification task on clinical notes to lay the foundations for de-noising clinical text. Compared to the existing solutions, our solution is novel by regarding the abbreviation identification task in a semi-supervised learning approach using level-wise feature engineering to construct an abbreviation identifier from a small set of labeled clinical text and a larger set of unlabeled clinical text which can be exploited for the learning process. In particular, a semi-supervised learning algorithm named Semi-RF and its advanced version, an adaptive semi-supervised learning algorithm named Adaptive Semi-RF are defined by taking advantages of random forest models and Tri-training. Adaptive Semi-RF is different from Semi-RF as equipped with a new weighting scheme via adaptation on the current set of labeled data to provide an appropriate treatment on each instance based on its being truly labeled. As a result, these algorithms are practical with no parameter setting requirement to generate an effective abbreviation identifier for identifying abbreviations automatically in clinical notes.

CCS Concepts:•**Theory of computation** → **Semi-supervised learning;**•**Information systems** → *Content analysis and feature selection;*•**Applied computing** → Annotation;

Additional Key Words and Phrases: Electronic medical record, clinical note, abbreviation identification, semi-supervised learning, self-training, random forest

ACM Reference Format:
Vo Thi Ngoc Chau, Cao Hoang Tru, and Ho Tu Bao, 2016. An Adaptive Semi-supervised Learning Approach to Automatic Abbreviation Identification. *ACM Trans. Embedd. Comput. Syst.* 9, 4, Article 39 (March 2010), 22 pages.
DOI: 0000001.0000001

1. INTRODUCTION

In recent years, electronic medical records (EMRs) which are electronic versions of medical records have got popular and significant in medical, biomedical, and healthcare research activities, leading to a growing need for their sharing and utilization from the outside. Once shared and processed by both human and computing machines, the content in these EMRs needs to be readable and understandable. Nev-

This work is funded by Vietnam National University at Ho Chi Minh City under the grant number B2016-42-01.
Author's addresses: C. T.N. Vo and T. H. Cao, Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology - Vietnam National University at Ho Chi Minh City, Vietnam; B. T. Ho, Japan Advanced Institute of Science and Technology, Japan, John von Neumann Institute, Vietnam National University at Ho Chi Minh City, Vietnam.

DOI: 0000001.0000001

ertheless, free text in clinical notes of these EMRs often contains explicit noises such as spelling errors, variants of terms (acronyms, abbreviations, synonyms ...), unfinished sentences, etc. [Kim et al. 2015]. One kind of the pervasive explicit noises is a set of abbreviations written in clinical notes due to a common use of abbreviations for writing-time saving and record simplification. Those abbreviations might result in misinterpretation and confusion of the content in the EMRs as mentioned in [Collard and Royal 2015]. They also greatly affect all automatic computer-based data processing tasks for analysis and knowledge discovery. Therefore, identifying abbreviations and replacing them with their correct long forms are necessary for enhancing the readability and sharability of the EMRs.

Aware of the significance of abbreviation resolution, many existing works have concentrated on different related tasks and purposes. As one of the first works considering medical abbreviations, [Berman 2004] has provided a listing of medical abbreviations in 6 nonexclusive groups for English medical records. The result from [Berman 2004] was well-known as Bermans list of abbreviations, which has been widely used for abbreviation disambiguation in English clinical notes. Another effort in [Wu et al. 2013] has been given for normalizing abbreviations in clinical text and another one in [Adnan et al. 2013] for enhancing the readability of discharge summaries. Furthermore, [Wu et al. 2012] has examined three natural language processing systems (MetaMap, MedLEE, cTAKES) for English clinical text to see how well these systems deal with abbreviations in discharge summaries. The authors have also suggested that accurate identification of clinical abbreviations is a challenging task. This suggestion is understandable because many abbreviations are dependent on context for their interpretation. As mentioned in [Long 2003], many of the abbreviations encountered have been commonly used but many dependent on context. Thus, capturing the surrounding context of each abbreviation is important to distinguish itself from non-abbreviations in clinical text. As for abbreviation detection in the related works, we witnessed the use of regular expressions in [Liu et al. 2015], word lists and heuristic rules in [Xu et al. 2007], and supervised learning approaches in [Wu et al. 2011], [Xu et al. 2007]. Nowadays, abbreviations in clinical notes are more difficult to be handled as compared to those in medical and biomedical articles where extraction patterns can be formulated with their long forms nearby. Moreover, the existing rule-based approaches with pattern matching cannot either cover the ambiguity of abbreviations with other non-abbreviations or especially capture the surrounding context of each abbreviation in free text of clinical notes well. Machine learning-based approaches become an advanced solution. Indeed, supervised learning has been utilized for abbreviation identification as proposed in [Wu et al. 2011], [Xu et al. 2007] while semi-supervised learning has been defined only in [Pakhomov et al. 2005] for acronym disambiguation. In our opinion, semi-supervised learning is preferred in practice because preparing a set of labeled data from all abbreviations and non-abbreviations in free text large enough for a learning process is a costly task. Generally speaking, the abbreviation identification task needs to be addressed for a more effective and practical solution as also mentioned in [Moon et al. 2015]. Besides, a semi-supervised learning approach should also be considered for abbreviation identification to exploit a large unlabeled data set in building an abbreviation identifier from a small labeled data set in practice so that abbreviations in clinical notes can be automatically and effectively detected.

In this paper, we propose an effective solution to automatic abbreviation identification in electronic medical records with a new adaptive semi-supervised learning approach. First, we characterize each token that is either an abbreviation or a non-abbreviation in clinical text at the token, sentence, and note levels using level-wise feature engineering. Many aspects of an abbreviation or a non-abbreviation can be examined and captured to be able to discriminate between an abbreviation and a

non-abbreviation. Especially, their contexts are defined according to their surrounding neighbors in a continuous bag-of-words model introduced in [Mikolov et al. 2013]. As a result, a comprehensive token-level vector representation for each token is achieved in a vector space, available for a learning process. Second, we define a traditional semi-supervised learning algorithm named Semi-RF and another adaptive semi-supervised learning algorithm named Adaptive Semi-RF to build an effective abbreviation identifier in a parameter-free self-training mechanism taking advantages of random forest models [Breiman 2001] and Tri-training [Zhou and Li 2005]. Adaptive Semi-RF is an advanced version of Semi-RF enhanced by a new weighting scheme via adaptation on a current set of labeled data to prepare an adaptive training set for identifier construction. The resulting identifier can be utilized for identifying abbreviations automatically in clinical text. Experimental results on various real clinical note types have confirmed the effectiveness of our solution with the smallest number of tokens incorrectly identified as abbreviations or non-abbreviation on average. Another intensive empirical study on benchmark data sets is carried out to show the effectiveness of the proposed algorithms consistently.

The rest of this paper is structured as follows. Section 2 presents a review on a few related works as compared to ours. In section 3, we define an abbreviation identification task along with level-wise feature engineering for clinical notes to represent each token (abbreviation or non-abbreviation) in a vector space. After that, we propose a new adaptive semi-supervised learning approach with two aforementioned algorithms, Adaptive Semi-RF and Semi-RF, in section 4 before making use of them as an effective solution to the abbreviation identification task. Section 5 then introduces our experimental results on UCI benchmark data sets and real clinical notes to show how effective the proposed algorithms are in comparison with some existing approaches. Finally, our work is concluded and some future plans are stated in section 6.

2. RELATED WORKS

In this section, several existing related works are reviewed in comparison with ours. First of all, we take into account abbreviation resolution in clinical text with the existing works where many related tasks have been considered to make some certain contribution to abbreviation resolution and then further to noise cleaning and readability improvement on electronic medical records. In particular, abbreviation detection has been focused on by the works in [Kreuzthaler and Schulz 2015], [Wu et al. 2011], [Xu et al. 2007], abbreviation resolution with disambiguation and expansion by [Henriksson et al. 2014], [Kim et al. 2011], [Kim et al. 2013], [Kim and Yoon 2015], [Liu et al. 2015], [Moon et al. 2013], [Moon et al. 2015], [Pakhomov et al. 2005], [Wong and Glance 2011], [Wu et al. 2013], [Wu et al. 2015], and sense inventory construction for abbreviations by [Liu et al. 2015], [Moon et al. 2014], [Wu et al. 2015], [Xu et al. 2007], [Xu et al. 2009]. In this paper, our work concentrates on the first important phase of abbreviation resolution that is automatic abbreviation identification. Using the abbreviations correctly identified, long forms can be determined for each abbreviation. As compared to the related works, our work aims to a more general solution to automatic abbreviation identification. Indeed, a few related works were specific for dealing with some kinds of abbreviations in clinical text. For example, [Kreuzthaler and Schulz 2015] has connected their solution to German abbreviation writing styles and [Kim and Yoon 2015] has paid attention to only the abbreviations that are 3 letters long. Another important point is that our work follows an unsupervised approach to level-wise feature engineering to be able to handle other unknown abbreviations in the future. Defining a different supervised approach, the related works in [Moon et al. 2013], [Wu et al. 2013], [Wu et al. 2015] used target abbreviations in their feature engineering. In addition, our work captures the context of each token at the sentence level

using a continuous bag-of-words model in a vector space while [Wu et al. 2011] only uses local context based on the characteristics of the previous/next word of each current word and [Xu et al. 2009] uses word forms of the surrounding words. Moreover, a comprehensive token-level vector representation is presented with level-wise features in a vector space in our work while many related works such as [Moon et al. 2013], [Moon et al. 2015], [Xu et al. 2007] are not based on a vector space model, leading to the different representations for clinical notes. Regarding the process of abbreviation identification, our work defines a new adaptive semi-supervised learning approach for abbreviation identifier construction while several related works have made use of regular expressions [Liu et al. 2015], word lists and heuristic rules [Xu et al. 2007] and some related works have followed a supervised learning approach [Wu et al. 2011], [Xu et al. 2007]. Furthermore, [Pakhomov et al. 2005] considered the availability of a large clinical report set of concern to automatic abbreviation resolution and thus, defined a semi-supervised approach with four steps: sense inventory, data collection, data merging, and context vectors generation. In [Pakhomov et al. 2005], their semi-supervised approach aimed at exploring the external sources such as World Wide Web, Medline abstracts, and Mayo Clinic corpus for building training sets and after that, made use of the decision tree C5.0 and the best matching based on cosine measure to find the sense of an acronym. In contrast, our work exploits both labeled and unlabeled data sets in constructing an abbreviation identifier in a semi-supervised learning approach so that abbreviations in clinical notes can be automatically and effectively detected. Last but not least, there is no available benchmark clinical data set for abbreviation identification in the present for empirical comparisons to be made with no bias. Each work has resolved this task using its own data set. This might be because of a high cost for clinical data preparation in each work, especially in machine learning-based works requiring large annotated data sets. Based on this review on the related works, it can be seen that our work has the merits of automatic abbreviation identification so that a correct set of abbreviations can be further taken into consideration for finding their true long forms and de-noising clinical notes for readability and sharability enhancement. The effectiveness of our proposed solution in an adaptive semi-supervised learning approach are proved with the experimental results on various real clinical note types. Besides, the general prediction power of the resulting classifiers learned by our semi-supervised learning algorithms is confirmed via the experimental results on UCI benchmark data sets [Archives 2016]. As compared to the existing related works, our proposed approach is novel and significant for the abbreviation identification task. This is because our resulting semi-supervised learning algorithms are parameter-free, able to start the learning process with a small set of labeled data available in practice, and then able to exploit a current larger set of unlabeled data for improving the prediction power of a resulting abbreviation identifier.

3. AUTOMATIC ABBREVIATION IDENTIFICATION IN ELECTRONIC MEDICAL RECORDS IN A VECTOR SPACE

Different from the related works, automatic abbreviation identification in electronic medical records is considered in a vector space in our work. The abbreviation identification task is the first phase of the clinical note de-noising process with abbreviation resolution. It contributes to cleansing free texts of abbreviations in clinical notes, one kind of explicit noises such as spelling errors, variants of terms (acronyms, abbreviations, synonyms, ...), unfinished sentences, etc. as mentioned in [Kim et al. 2013]. Our abbreviation identification task along with vector representation for clinical notes is defined below.

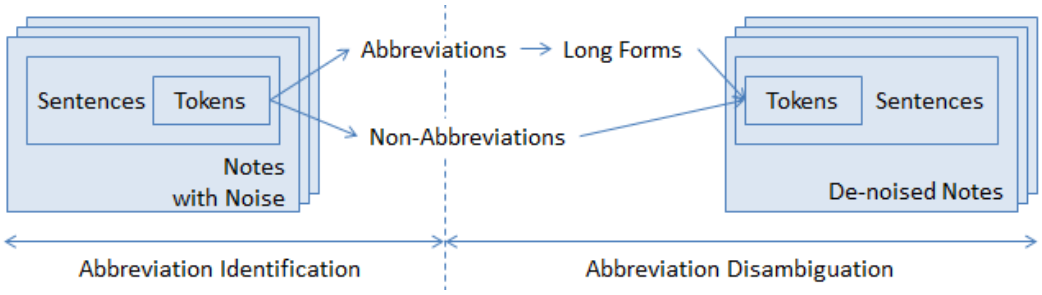

Fig. 1. De-noising Clinical Notes with Abbreviation Resolution.

3.1. Abbreviation Identification Task Definition

Introduced in Figure 1, de-noising clinical notes with abbreviation resolution consists of two consecutive phases: (1). Abbreviation Identification and (2). Abbreviation Disambiguation. The first phase will extract abbreviations from the clinical notes and the second phase finds an appropriate long form corresponding to each abbreviation. The entire process will make clinical notes with abbreviations readable from the outside and further shared for other computer-based data processing and analysis tasks.

In the context of abbreviation resolution, we formulate the abbreviation identification task as a token-level binary classification task on the free text of the clinical notes. Binary means that there are two predefined classes (groups) corresponding to a group of abbreviations (class = 1) and another group of non-abbreviations (class = 0). Token-level means that each object in the classification task is defined at the token level of the clinical notes. In particular, the input of this abbreviation identification task includes clinical notes that contain free text and its output is a set of abbreviations and a set of non-abbreviation written in the clinical notes. Indeed, each token obtained from the free text in the clinical notes is an object (instance) in the classification task. A token is either an abbreviation or a non-abbreviation. If at the beginning, there is an available set of tokens given to be either an abbreviation or a non-abbreviation, i.e. an available set of labeled tokens, the classification task can be performed in a supervised learning or semi-supervised learning mechanism. In practice, a semi-supervised learning mechanism is preferred in such a case that the available set of labeled tokens is small and there exists a large set of unlabeled tokens that need to be determined as either abbreviation or non-abbreviation. Besides, it will be helpful if this set of unlabeled tokens is able to be exploited for the learning mechanism. As for our work, we decide to approach this abbreviation identification task in a semi-supervised learning mechanism for more effectiveness by using the set of unlabeled tokens for advantage. Particularly in the next section, a semi-supervised learning algorithm based on random forest models called Semi-RF and its extended version, an adaptive semi-supervised learning algorithm also based on random forest models called Adaptive Semi-RF, are defined in detail. These algorithms can facilitate our abbreviation identification task because of their parameter-free configuration scheme.

In addition, each token must be represented in a computational form to be processed in the classification task. In our work, we use a vector space model that allows each token to be represented as a vector, called a token-level feature vector, in a vector space. A vector corresponding to a token in the set of labeled tokens is used in the phase of constructing the abbreviation identifier and called a training (labeled) vector. Each training vector is given a true class value which is either 1 for an abbreviation or 0 for a non-abbreviation. On the other hand, a vector corresponding to a token that needs to be determined as an abbreviation or not in the set of unlabeled tokens will be

assigned a class value, abbreviation or non-abbreviation, after the abbreviation identifier resulted in the classification task classifies a vector to an appropriate class in the abbreviation identification phase. In general, each token is characterized by p features corresponding to p dimensions of a vector space where a p-dimension vector representing each token is located and the abbreviation identification task is conducted.

3.2. Representing Clinical Notes in Electronic Medical Records in a Vector Space

In order to represent each token in the clinical notes in EMRs in a vector space, we first design the structure of each token in the form of a vector and then process the clinical notes to generate its vector by extracting and calculating its feature values.

As depicted in Figure 2, both labeled and unlabeled clinical notes in EMRs include sentences each of which contains many tokens that can be attained with tokenization. In such a multilevel view on clinical notes in EMRs, we do feature engineering by capturing many different aspects of each token from the most detailed level (token) to the coarsest one (note): token, sentence, and note. In particular, we consider the features at the token, sentence, and note levels for the first step in representing clinical notes, i.e. (1). Unsupervised Feature Vector Space Building. For the second step, i.e. (2). Feature Value Extraction, each element of a vector corresponding to a token is determined according to the characteristics of the token at the three levels of detail. A training (labeled) vector will be annotated additionally with its class value, which is abbreviation (1) or non-abbreviation (0). As a result, a token is represented in the following form of a vector:

$$\mathrm{X} = (x_1^t, ..., x_{tp}^t, x_1^s, ..., x_{sp}^s, x_1^n, ..., x_{np}^n)$$

in a vector space of p dimensions where x_i^t is a feature value of the i-th feature at the token level for i=1..tp, x_j^s a feature value of the j-th feature at the sentence level for j=1..sp, and x_k^n a feature value of the k-th feature at the note level for k=1..np; and tp is the number of token-level features, sp the number of sentence-level features, and np the number of note-level features, leading to p=tp + sp + np. Details of these level-wise features are delineated below.

At the token level, each token is characterized by its own aspects such as word form with orthographic properties (e.g. containing any digit, containing any special character, composed of all consonants), word length, and semantics (e.g. being a medical term and being an acronym of any medical term). The token-level features are described as follows:

— *AnyDigit*: indicating if the current token contains any digit such as 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9. If yes, one (1) is the corresponding feature value. Otherwise, zero (0) is used. The use of digits in abbreviations is little; however, they might be used to shorten the long form of some number or combined with letters in abbreviations.
— *AnySpecialChar*: indicating if the current token contains any special character such as ., ;, ,, -, _, (,), @, %, &, and so on. If yes, one (1) is the corresponding feature value. Otherwise, zero (0) is used. It is found that abbreviations dont often contain special characters except for -, _, and . for connecting the components of their long forms.
— *AllConsonants*: indicating if the current token is composed of all consonants such as b, c, d, ..., w, x, z. If yes, one (1) is the corresponding feature value. Otherwise, zero (0) is used. In our work, we consider acronyms to be a special group of abbreviations. Thus, all abbreviations which are acronyms created by a sequence of the first letters of the components of their long forms tend to contain all consonants. Nonetheless, there exist some abbreviations including vowels.

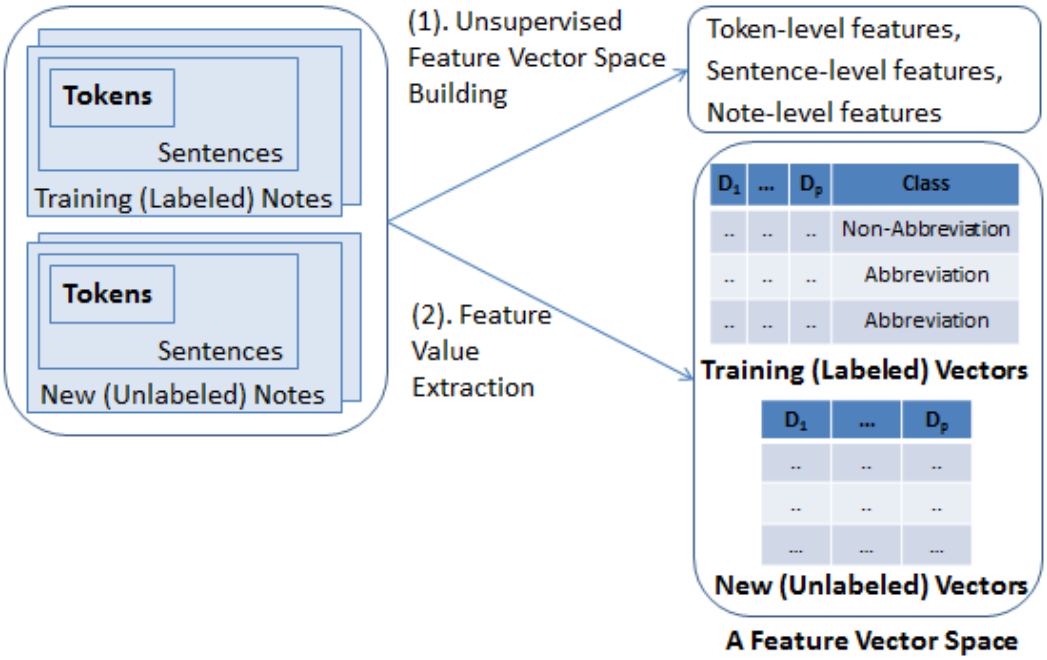

Fig. 2. Representing clinical notes in electronic medical records in a vector space.

— *Length*: the number of characters in the current token. We find out that most of the abbreviations are short due to the main usage purpose of abbreviation writing which is time-saving.
— *inDictionary*: indicating if the current token is included in a given medical dictionary. This dictionary is regarded as an external resource to provide us with the semantics of the tokens in case they are medical terms in the biomedical domain. If yes, one (1) is the corresponding feature value. Otherwise, zero (0) is used. This token-level feature helps realizing that the current token might be a non-abbreviation as medical terms in the dictionary are in their full forms.
— *isAcronym*: indicating if the current token matches any acronym of a medical term in the aforementioned dictionary. If yes, one (1) is the corresponding fea-ture value. Otherwise, zero (0) is used. In contrast to inDictionary, this token-level feature helps us point out that the current token might be an abbreviation.

At the sentence level, many contextual features are defined from the surrounding words of each token in the sentence where it is contained. In order to encode the context of each token in a vector space, we use a word embedding vector. As introduced in [Mikolov et al. 2013], each word can be represented as a continuous vector in a vector space using either a continuous bag-of-words model or a continuous skip-gram model. The previous model predicts the current word based on the context words while the latter predicts the words in a certain range before and after the current word which is now an input. Because our abbreviation identification task concentrates on deciding if a current token is an abbreviation or not, we would like to capture the context of each word based on its surrounding words. Therefore, a continuous bag-of-words model is an appropriate choice of focusing on the current token and generating the sentence-level contextual features. The number of the resulting sentence-level contextual features is the output layer size V of the continuous bag-of-words model.

At the note level, occurrence of each token in clinical notes is considered as a note-level feature. We use a term frequency to capture the number of occurrences of each token for simplifying the computation of the feature value extraction step.

For our encodings in the abbreviation identification task, we have designed each token representation in a (7+V)-dimension vector space with tp = 6, sp = V, and np = 1 where V is sp, an output layer size in the continuous bag-of-words model. Such a token representation has the advantages highlighted as follows. First, unlike the related works [Moon et al. 2013], [Wu et al. 2013], [Wu et al. 2015] where feature engineering is supervised with class information in terms of target abbreviation, ours does level-wise feature engineering in an unsupervised manner with no class information. Thus, our approach is more practical and applicable for abbreviation identification in abbreviation resolution to the coming electronic medical records over time. Second, our work has defined a comprehensive representation of each token in clinical notes that can capture many different aspects of each token from the most detailed level to the roughest level, suitable for the con-text of abbreviation usage where an abbreviation writing habit is formed, agreed, and maintained in a group of people for certain note types. Therefore, sentence-level and note-level features get important for abbreviation determination while token-level features are remained to characterize each token. Third, there is no restriction on abbreviation writing styles as our feature engineering does not make use of abbreviation writing styles. Above all, our level-wise feature engineering has no restriction on note structures as only token occurrence is examined at the note level. Specific note structures were not encoded into the resulting features so that many various clinical types could be supported over time. Simply, we examine two groups of tokens: one for abbreviations and another one for non-abbreviations. This means that there is no distinguishing between abbreviations and acronyms, leading to the capability to identify a large set of so-called short forms, i.e. abbreviations, in clinical notes.

4. AN ADAPTIVE SEMI-SUPERVISED LEARNING APPROACH TO AUTOMATIC ABBREVIATION IDENTIFICATION IN CLINICAL NOTES

In this section, we first propose two semi-supervised learning algorithms named Semi-RF and Adaptive Semi-RF. Semi-RF is defined as a semi-supervised one in a combined self-training and tri-training style of a random forest while Adaptive Semi-RF is an adaptive version of Semi-RF with a weighting scheme for proper treatment of the labeled instances in the learning process. Their analysis is also discussed. A solution to the abbreviation identification task is then highlighted in a vector space by means of the proposed algorithms.

4.1. The Proposed Adaptive Semi-Supervised Learning Approach Based on Random Forest Models

Bringing the advantages of random forest models [Breiman 2001] and Tri-training [Zhou and Li 2005] to the self-training approach, our work proposes a new adaptive semi-supervised learning approach. Random forest is a well known ensemble learning algorithm that is based on bootstrap sampling and random trees while Tri-training is an advanced parameter-free co-training style algorithm that did not ask for two sufficient and redundant views; instead, used three classifiers not required to be different partitioning the data space into a set of equivalence classes. As for the self-training approach originally introduced in [Yarowsky 1995], self-training is one of the simplest semi-supervised learning algorithms that needed neither two sufficient and redundant views nor many different classifiers that could partition the space into a set of equivalence classes. Nevertheless, the simplicity of the self-training approach places a burden on its users because the users have to determine an appropriate base clas-

sifier wrapped by self-training, define the way to select the so-called most confidently predicted instances from the set of unlabeled instances to enlarge the set of labeled instances in each round, and above all, set a correct value to the probability threshold for newly labeled instance selection. Different from the existing semi-supervised learning algorithms in [Li and Zhou 2007], [Tanha et al. 2015], [Yarowsky 1995], [Zhou and Li 2005], our semi-supervised learning algorithms are defined with the following foundations:

— Combining a random forest model and Tri-training in a self-training approach so that the resulting semi-supervised learning algorithms can be parameter-free and simple, but effective for the classification task.
— Maintaining the advantages of a random forest model in the finalized classifier which can be effective, robust, and un-overfitting.
— Differentiating between the instances in both sets of labeled and unlabeled data in the learning process because they are different from each other in many aspects such as their labels (true or predicted) and recognition (easy or hard). Such a treatment can make the resultant algorithm more effective by favoring the truly labeled instances over those wrongly labeled instances in a weighting scheme.

As a result, the proposed algorithms are shaped in the form of self-training but more flexible and effective by using a random forest model of three random trees with $(\lfloor log(\boldsymbol{p}) \rfloor + 1)$ random features. Three random trees play the role of three classifiers in Tri-training so that the probability threshold can be automatically established to select the most confidently predicted instances from a current set of unlabeled data. The number of the random features of each of these three random trees is based on the theory of random forest models in [Breiman 2001]. Such a design results in our algorithms free of parameter settings. In addition, bootstrap sampling is remained in random forest construction in each round and so is the diversity of the three random trees. This maintained diversity is significant for a majority voting scheme in classification by an ensemble model. Besides, a weighting scheme that favors truly labeled instances and easily predicted instances is introduced via adaptation on a current set of labeled data including both truly labeled and newly labeled instances at the beginning of each round. This weighting scheme makes the current set of labeled data adaptive to such truly labeled and newly easily predicted instances. Further, it will shift the prediction of our finalized classifier towards these instances and constrain the hard newly predicted instances which might be wrongly labeled. Thus, those perhaps wrongly predicted instances have less impact on the prediction power of the finalized classifier in this weighting scheme. Moreover, the optimization of our algorithms is based on the generalization of the finalized random forest model over the original set of labeled data containing true labels that we certainly know and the finalized classifier does too. These points form the stability of our algorithms in their halting.

For details, the pseudo-code of our adaptive semi-supervised learning algorithm Adaptive Semi-RF is given in Algorithm 1 and its traditional semi-supervised learning algorithm Semi-RF is a simpler version without the weighting scheme via adaptation on the set of labeled data. Along with the Adaptive Semi-RF algorithm, details of the weighting scheme are given in Algorithm 2 and details of the selection of the most confidently predicted instances from the current set of unlabeled data are given in Algorithm 3.

Described in Algorithm 1 in an iterative manner, our Adaptive Semi-RF algorithm performs in each iteration as follows. A weighting scheme is invoked on the current set of labeled instances to provide another adaptive set which will be later used in constructing a current random forest model of three random trees with $(\lfloor log(\boldsymbol{p}) \rfloor + 1)$ random features. This current classifier is then evaluated on the original set of

ALGORITHM 1: The Proposed Adaptive Semi-supervised Learning Algorithm Adaptive Semi-RF on both Labeled and Unlabeled Data in the p-dimension Vector Space

Input: *lSet* which is a labeled set and *uSet* which is an unlabeled set, both originally given in the p-dimension vector space.
Output: C which is a resulting classifier.
Set a previous error rate $Previous\_error\_rate$ to 0.5;
Assign *lSet* as a current set $clSet$ which contains all instances with known labels;
Assign *uSet* as a current set $cuSet$ which contains all instances with unknown labels;
repeat
  Weighting the labeled instances via *adaptation* on the labeled set $clSet$ to obtain an adaptive labeled set $clSet\_a$;
  Build a current random forest Current_RF of three random trees with $(\lfloor log(p) \rfloor + 1)$ random features on $clSet\_a$;
  Compute a current error rate $Current\_error\_rate$ by evaluating Current_RF on *lSet*;
  if *Previous_error_rate* $>$ *Current_error_rate* **then**
    $Previous\_error\_rate$ = $Current\_error\_rate$;
    Save the current random forest Current_RF as a previous random forest Previous_RF;
    if *cuSet is not empty* **then**
      Predict a label of each instance in $cuSet$ using Current_RF;
      Select a set $sSet$ of the *most confidently predicted instances* from $cuSet$;
      Update $clSet\_a$ to $clSet$ by including $sSet$;
      Update $cuSet$ by excluding $sSet$;
    end
    else Return the current random forest Current_RF as a resulting classifier C;
  end
  else Return the previous random forest Previous_RF as a resulting classifier C;
until *the termination conditions are met*;

labeled data. If its error rate is less than the previous error rate set previously, i.e. its prediction power has got better, the previous error rate and a previous classifier are updated with the new current ones. Otherwise, the previous classifier has been the best so far and thus, returned as a resulting classifier C. If improvement is found, exploiting unlabeled data is considered. If the current set of unlabeled data is not empty, we use the current classifier to predict a label of each instance in this set. After that, the most confidently predicted instances are selected from this unlabeled set and added into the current set of labeled instances to enlarge the training set in the next iteration. The current unlabeled set is also updated by removing those chosen instances. If the current unlabeled set is empty, the learning process will stop and return the current classifier as a resulting classifier C. As specified in Algorithm 1, a resulting classifier C is obtained with two termination conditions: no element in the current set of unlabeled data and no improvement on the prediction power of the resulting classifier on the original set of labeled data. The first termination condition is based on the general rationale behind a semi-supervised learning approach which aims to exploit unlabeled instances in the learning process to enhance the learnt classifier when there are a few labeled instances. If there is no unlabeled instance for the exploitation, the learning process will end. As for the second one, if the exploitation is not positive for enhancing the current classifier which has been the best one so far, the learning process will end as well so that the current prediction power of this classifier can be kept for use. These two termination conditions ensure the convergence of our proposed algorithms.

The entire learning process of our algorithms is in a self-training mechanism but the use of a random forest model of three random trees and the selection of the most confidently predicted instances have turned our algorithms in a tri-training mechanism. On the other hand, the learning process is enhanced with the aforementioned

ALGORITHM 2: *Weighting* the labeled instances via *adaptation* on a Current Set *clSet* of Labeled Instances in the 5-fold Cross Validation Scheme

Input: *clSet* which is a current set which contains all instances with known labels in the p-dimension vector space.
Output: *clSet_a* which is a current set which contains all instances with known labels after adaptation in the p-dimension vector space.
$clSet\_a$ = $clSet$;
Do stratified random sampling without replacement on $clSet$ into 5 folds that have similar size (almost the same size);
for *each fold* f **do**
  Build a random forest aRF of three random trees with $(\lfloor log(p) \rfloor + 1)$ random features on a set which is $clSet$ excluded the current fold f;
  Evaluate aRF on the current fold f;
  Update $clSet\_a$ with the instances of the current fold f correctly recognized by aRF;
end
Return $clSet\_a$;

weighting scheme via adaptation on the current set of labeled instances. These two main advantages are discussed as follows:

First, our *weighting* scheme in Algorithm 2 is done with stratified random sampling without replacement to make *adaptation* on the current set of labeled instances into 5 similarly-sized folds. In a 5-iteration loop of the k-fold cross validation style, four out of 5 folds form a training set to build a random forest model of three random trees with $(\lfloor log(p) \rfloor + 1)$ random features which will be then used to predict the remaining fold. The instances of the remaining fold correctly predicted are added into the adapted current set of labeled instances which will be returned as a result of the weighting scheme. Performed at the beginning of each iteration in the semi-supervised learning process of the proposed algorithm Adaptive Semi-RF sketched in Algorithm 1, adaptation on the current set of labeled data helps us weight each instance in favor of its being truly labeled. Indeed, weighting is different for an instance that has a true label given in the original set of labeled data and another one that has a predicted label given in the semi-supervised learning process. It is also different for an instance that has a truly predicted label and another one that has a wrongly predicted label, both given and selected in the semi-supervised learning process. We formally present the weighting differences for each instance in the weighting scheme via adaptation in detail as follows.

Let us denote $lSet$ be an original set of labeled data, $clSet$ be a current set of labeled data, $clSet\_a$ be an adaptive set of labeled data after performing the weighting scheme on $clSet$, and $cuSet$ be a current set of unlabeled data. At the instance level, let us denote O_1^l and O_2^l be instances in $lSet$, O_1^u and O_2^u in $cuSet$ which are both selected as the most confidently predicted instances. Among these instances, it is supposed that O_1^l and O_1^u are all the instances that are always correctly recognized and in contrast, O_2^l and O_2^u are all the instances that are always wrongly recognized by our current classifier. Let us consider the weighting scheme via adaptation in the first two iterations of our semi-supervised learning process.

In the first iteration, there are two O_1^ls and one O_2^l after adaptation on $clSet$ which is $lSet$ previously and is now $clSet\_a$. The set $clSet\_a$ is then updated with the most confidently predicted instances. At the end of the first iteration, a current set of labeled data $clSet$ is:

$$clSet = lSet \cup \{O_1^l\} \cup \{O_1^u, O_2^u\} = \{..., O_1^l, O_2^l, ...\} \cup \{O_1^l\} \cup \{O_1^u, O_2^u\}$$

In the second iteration, similarly $clSet$ is now updated to $clSet\_a$ as:

$$clSet\_a = lSet \cup \{O_1^l, O_1^l, O_1^l\} \cup \{O_1^u, O_1^u, O_2^u\}$$
$$clSet\_a = \{..., O_1^l, O_2^l, ...\} \cup \{O_1^l, O_1^l, O_1^l\} \cup \{O_1^u, O_1^u, O_2^u\}$$

After weighting, the semi-supervised learning process continues as planned.

After i iterations, there are 2^i O_1^ls and 2^{i-1} O_1^u showing our learning process is performing in favor of O_1^l more than in favor of O_1^u, and certainly much more than in favor of the other instances considered to be perhaps wrongly predicted like O_2^u.

By conducting the weighting scheme via adaptation, it appears that our learning process might consider the truly labeled instances so much that it can be easily leading to overfitting. This is not a fact in our proposed algorithm, Adaptive Semi-RF, due to the characteristics of random forest models. Indeed, the diversity of the random trees in the random forest is maintained even if their training data sets are similar as mentioned in [Li and Zhou 2007]. As a result, only truly labeled instances have mainly contributed to our semi-supervised learning process while probably wrongly labeled instances that have added into the training data set would have had less impact on the learning process. A resulting adaptive current set of labeled instances after weighted will be used to build a random forest model in the self-training approach.

Second, we consider each instance from the current set of unlabeled instances for the *most confidently predicted instance selection* scheme in Algorithm 3. Let us denote m be the number of classes and t be the number of random trees in the random forest model which is 3 in the case of our algorithms. The prediction score of a current instance X^* is calculated as follows:

— Each random tree *j* performs a prediction on X^* and provides a class distribution score of each class C_i for i=1..*m* for X^* which is:

$$P_j(C_i|X^*) = \frac{k}{N}$$

where *k* is the number of instances in class C_i out of *N* instances in the training set of the tree *j* at the leaf node at prediction by the tree *j*.

— Summing all the class distribution scores of each class C_i for i=1..*m* by all *t* trees provides a class distribution score of each class C_i at prediction for X^* by the random forest model which is:

$$\text{P}(C_i|X^*) = \sum_{j=1..t} P_j(C_i|X^*)$$

These class distribution scores P($C_i|X^*$) for i=1..*m* are then normalized into the range [0, 1] such that:$\forall i = 1..m, 0 \leq P(C_i|X^*) \leq 1$ and $\sum_{i=1..m} P(C_i|X^*) = 1$.

— Based on the majority voting scheme, a final prediction score of X^* is determined as the maximum prediction score P($C_i|X^*$) for i=1..*m* and its predicted class is C_i corresponding to the maximum prediction score P($C_i|X^*$):

$$\text{Prediction score } (X^*) = \max \{\text{P}(C_i|X^*) \text{ for i=1..}m\}$$
$$\text{Predicted class } (X^*) = \text{argmax}_{Ci} \{ \text{P}(C_i|X^*) \text{ for i=1..}m\}$$

Once a prediction score is computed for the instance X^*, it will be considered with the following selection scheme. If the prediction score of X^* is 1, X^* is selected and added into the resulting set. Its predicted label is now considered true label. The reason for a threshold value of 1 in our selection scheme stems from the assumption about the prediction of each random tree in an ensemble and also about the final prediction of that ensemble. This is because our classifier is a random forest model of three random trees. First, consider a binary classification task where there are two classes (0 and 1), it is well known that an ensemble makes a wrong prediction only if more than half its base classifiers make wrong predictions in a majority voting scheme. Thus, in order to reduce a chance of selecting an instance wrongly predicted, we set the

ALGORITHM 3: *Selecting* a Set $sSet$ of the *Most Confidently Predicted Instances* from the Current Set $cuSet$ of Unlabeled Instances

Input: $cuSet$ which is a current set which contains all instances with unknown labels in the p-dimension vector space.
Output: $sSet$ which is a selected set of the most confidently predicted instances in the p-dimension vector space.
for *each instance* X^* *in* $cuSet$ **do**
  Calculate a *prediction score* for the current instance X^*;
  if *its prediction score = 1* **then**
    Add this current instance X^* into $sSet$;
  end
end
Return $sSet$;

threshold to 1 to make sure that all random trees agree on the predicted class value. If the threshold were set to a smaller value (< 1), all the predicted instances in the current set of unlabeled data would be selected because there is always such a case that at least two random trees agree on the same class. This fact would lead to a non-significant selection. Second for a general multi-class classification task with the number of classes greater than 2, we consider the instances correctly predicted with an agreement on the same class from all random trees, leading to a threshold of 1. It is worth noting that a prediction score of an instance in our classifier is not simple vote counts from each random tree in the random forest model. Instead, it is a distribution score from each random tree based on the purity of the instance set at the leaf node where prediction is made. This score is a finer value in range [0, 1] and then aggregated for the entire forest. The selection criterion is valid for selecting the most confidently predicted instances. The resulting set of selected instances will be used to enlarge the current set of labeled instances which is originally small in our semi-supervised learning process.

In short, Semi-RF is our semi-supervised learning algorithm using random forest models as its base model in a combined self-training and tri-training manner. Adaptive Semi-RF is its extended version, which enhances the training set of labeled data with the more likely correct instances by memorizing them time after time in a weighting scheme via adaptation. These two algorithms are applicable to classifier construction from a small labeled data set in practice and able to provide a resulting random forest model effective for label prediction based on the majority voting scheme. It is also worth noting that they have no restriction on the number of classes as well as parameter configurations.

4.2. Automatic Abbreviation Identification in Clinical Notes in a Semi-Supervised Learning Paradigm

In this subsection, an abbreviation identification process on electronic medical records is examined in a semi-supervised learning paradigm and based on the abbreviation identification task definition and level-wise feature engineering in the previous section. As sketched in Figure 3, an abbreviation identification process is conducted with two subprocesses in sequence: *A. Identifier Construction* and *B. Abbreviation Identification*. With their following details, we show how the proposed semi-supervised learning algorithms help performing this abbreviation identification process.

A. Identifier Construction

In our current work, the abbreviation identification task is regarded a classification task and thus, a collection of training vectors needs to be ready as an input for identifier construction. In addition to training vectors which are labeled, this subprocess

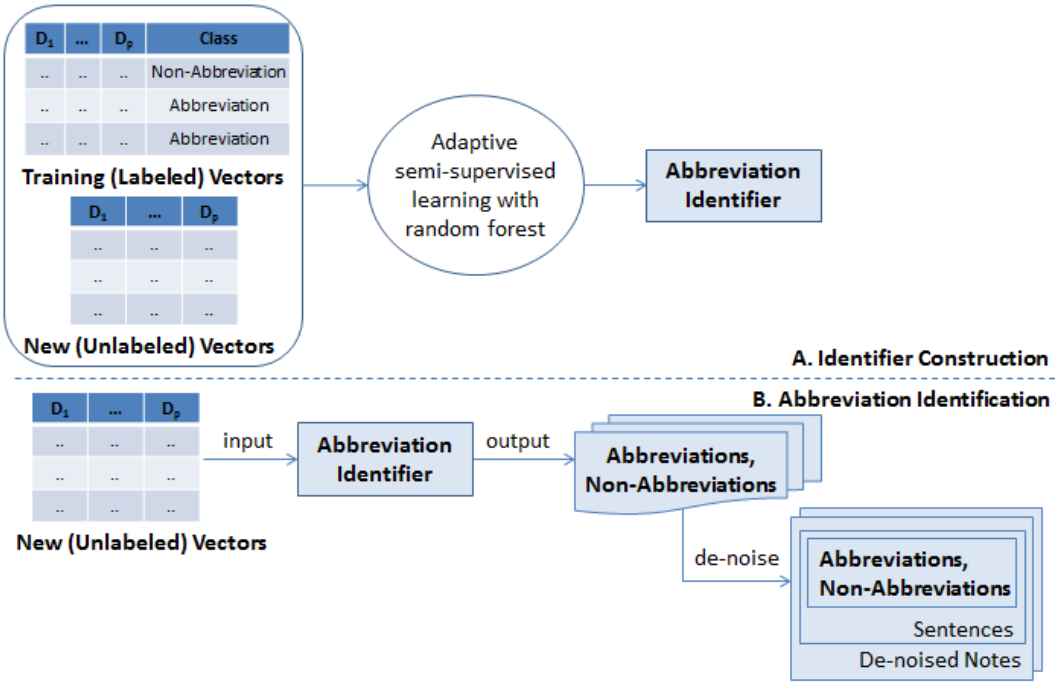

Fig. 3. The proposed abbreviation identification process using an adaptive semi-supervised learning approach on electronic medical records.

makes the most of unlabeled vectors to enlarge the training set of labeled vectors and enhance the prediction power of the resulting abbreviation identifier. Thus, the input of this subprocess also includes a set of unlabeled vectors. After that, our adaptive semi-supervised learning algorithm, Adaptive Semi-RF, is invoked using random forests to process these vectors with no need of parameter configuration. A classifier which is our abbreviation identifier is finally returned. Before this abbreviation identifier is shifted to the next subprocess for use, an evaluation is made by maximizing its performance for abbreviation identification on the original set of labeled vectors.

B. Abbreviation Identification

The second subprocess comes after the previous one gets done. It will make use of the abbreviation identifier to decide which unlabeled token is an abbreviation and which is not by classifying a corresponding token-level vector. The resulting abbreviations and non-abbreviations are then marked for the corresponding tokens in the clinical notes so that long forms of the significant abbreviations can be resolved in the next de-noising phase, abbreviation disambiguation. How correctly a token is marked with abbreviation or non-abbreviation relies on how effective the abbreviation identifier is. For more certainty, human interaction might be considered to check the predicted abbreviations and non-abbreviations in practice in the future. It is also worth noting cost of time and effort for human interaction in this task. Therefore, automatic abbreviation identification is significant for processing a large number of clinical notes in electronic medical records worldwide nowadays.

Before going into detail over an empirical study, we would like to emphasize the convenience and applicability of our solution to the abbreviation identification task on clinical text. First of all, the abbreviation identification task is carried out in a parameter-free scheme in our work. Indeed, all abbreviations in clinical text can be

identified automatically with no parameter setting effort from the users side. Second, our solution is not specific for languages, note structures, and note types used for clinical text from feature engineering to process implementation. Third, it is also general for all types of abbreviations and their lengths because our task does not consider abbreviations as individuals and instead, simply identifies new unlabeled tokens to be abbreviations or non-abbreviations. In short, our solution is simple and practical to be deployed in practice and lay the basis for abbreviation resolution.

5. EXPERIMENTAL RESULTS

In this section, an evaluation of our proposed semi-supervised learning approach is made from the empirical point of view. For generalization, we conducted two experiment groups: one on 10 benchmark data sets from UCI Archives [Archives 2016] described in Table I and another on three clinical note sets [Hospital 2016] in Table III. The program is written in Java using the supervised learning algorithm implementation in Weka 3 [Bouckaert et al. 2016]. For comparison, our result tables in Table II, Table IV, Table V, and Table VI show the number of instances incorrectly recognized by Random Forest [Breiman 2001], Self-training [Yarowsky 1995], Tri-training [Zhou and Li 2005], Co-Forest [Li and Zhou 2007], Semi-RF_2/3, Semi-RF, and Adaptive Semi-RF. Random Forest is included because it is a base model in our proposed semi-supervised learning algorithms with 3 random trees. In addition, Tri-training with C4.5 is selected because [Triguero et al. 2015b] has performed an empirical study on semi-supervised learning algorithms and found that Tri-training with C4.5 is often among outstanding methods in their experiments. Self-Training with C4.5 and a confident threshold equal to 0.75 is also among the outstanding methods mentioned in [Triguero et al. 2015b]. Co-Forest with 6 random trees and a confident threshold also equal to 0.75 is used for comparison because our proposed semi-supervised learning algorithms utilized Random Forest as a base model similarly to Co-Forest. Beside Semi-RF and Adaptive Semi-RF, we record the performance of Semi-RF_2/3 which is our Semi-RF algorithm using a threshold of 2/3 like the agreement check among the three classifiers in Tri-training. Semi-RF_2/3 is used for checking how effective our most confidently predicted instance selection scheme is. More recently several new semi-supervised learning approaches have been proposed in [Tanha et al. 2015], [Triguero et al. 2015a]. However, we did not include these approaches in our comparison because their approaches are not in the same direction as ours. In particular, [Triguero et al. 2015a] aimed at a framework wrapping several existing semi-supervised learning algorithms. It prepares a better starting set of labeled data for the semi-supervised learning process by generating synthetic labeled instances using the idea of the synthetic minority over-sampling technique (SMOTE) for all classes. As for [Tanha et al. 2015], a self-training approach is examined for decision trees with the new prediction score definitions (called probability estimates in [Tanha et al. 2015]). In our work, we use the simplest definition to define a prediction score from a random tree based on the class distribution at the leaves of each tree. Furthermore, our self-training mechanism is modified with random forest models in a Tri-training style so that the resulting algorithms can be parameter-free. Our semi-supervised learning algorithm is then developed more with our new weighting scheme via adaptation. These characteristics have not yet been considered in [Tanha et al. 2015], [Triguero et al. 2015a]. Above all, the algorithms in [Tanha et al. 2015], [Triguero et al. 2015a] require parameter settings while ours are parameter-free. Thus, we think that a comparison should not be made to avoid any bias. Regarding performance measures, each value in the result tables is the number of instances incorrectly recognized by each algorithm as previously mentioned. The smaller number is the better. For reliable accuracy/error estimation, we use the k-fold cross validation scheme in the context of semi-supervised learning where a data set

Table I. Details about Benchmark Data Sets

Data Sets	Description	Instance#	Attribute#	Class#
Iris	Iris plants database	150	4	3
Drift	Gas sensor array drift data set at different concentrations	459	128	6
Gesture	Gesture phase segmentation	1743	32	5
Wine_quality_white (WQ_white for short)	Wine quality using white wine samples	4898	11	7
Wine_quality_red (WQ_red for short)	Wine quality using red wine samples	1599	11	6
Credit_card	Default payments of credit card clients in Taiwan	30000	23	2
Sensorless_drive_diagnosis (Diagnosis for short)	Sensorless drive diagnosis data whose features are extracted from motor current	58509	48	11
EEG_eye_state (EEG for short)	EEG eye state data from one continuous EEG measurement with the Emotiv EEG Neuroheadset	14980	14	2
Aggressive_1	Physical action data that include 10 aggressive actions measuring the human activity	20400	27	10
Normal_2	Physical action data including 5 normal actions (bowing, clapping, handshaking, hugging, jumping)	8768	27	5

is divided into k folds in a stratified sampling scheme with random sampling without replacement and one fold is used as a labeled data set and the rest (k-1) folds are used as an unlabeled data set. The number of instances incorrectly recognized is the total sum after k iterations in the validation scheme. Due to space limitation, the total sum of instances incorrectly identified over k folds for k=2..10 is provided for each benchmark data set and the total sum of instances incorrectly identified for each case of k folds for k=2..12, 14, 16, 20 is recorded for each clinical note set. The total sum for each algorithm is given at the last row in each result table where the best results are written in bold and the second best ones are underlined. Besides, $\%\Delta\_min$ and $\%\Delta\_max$ are calculated to show how much improvement is gained by comparing the results from Adaptive Semi-RF with the best and worst results, respectively, from Random Forest, Self-training, Tri-training, and Co-Forest. If either $\%\Delta\_min$ or $\%\Delta\_max$ is a negative number, our proposed algorithm is not better than the others. Otherwise, our algorithm outperforms the others. In order to avoid randomness in experimental results, Paired-Samples T test with confidence interval percentage of 95% is also used to compare the results between Adaptive Semi-RF with each of the others. All statistical tests have found that the mean difference between Adaptive Semi-RF and each of the others is statistically significant at the 0.05 level of significance with Sig.(2-tailed) = .000. That is the number of instances incorrectly recognized by Adaptive Semi-RF is significantly smaller than that of each of the others on average. Generally speaking, Adaptive Semi-RF outperforms the others on the data sets on average in these two experiment groups. Detailed experimental results are discussed below for each group.

5.1. Experiments on Benchmark Data Sets

In Table I, ten benchmark data sets are introduced and their descriptions are briefly given. They are different from each other in terms of the number of instances Instance#, the number of attributes Attribute#, and the number of classes Class#.

Shown in Table II, our proposed algorithm Adaptive Semi-RF outperforms the others with the lowest number of instances incorrectly recognized in the data sets except Credit_card data set. It has the highest difference on Aggressive_1 and Normal_2, which consist of the highly non-overlapping classes. It is also interesting to note that

Table II. The Number of Instances Incorrectly Recognized in Benchmark Data Sets

Data Sets	**Random Forest**	**Self-training**	**Tri-training**	**Co-Forest**	**Semi-RF_2/3**	**Semi-RF**	**Adaptive Semi-RF**	$\%\Delta\_min$	$\%\Delta\_max$
Iris	827	720	792	721	823	679	**627**	12.92	24.18
Drift	926	1393	1402	1244	914	830	**547**	40.15	60.98
Gesture	30555	32617	33110	36162	30231	27813	**27566**	8.82	23.77
WQ_white	116556	116270	116802	117599	116316	109367	**101976**	12.29	13.28
WQ_red	35457	35003	35265	36222	35356	33272	**31994**	8.6	11.67
Credit_card	314753	**275600**	331382	372450	307515	287365	277398	-0.65	25.52
Diagnosis	64881	78620	81720	138264	60301	30498	**20297**	66.34	85.32
EEG	145414	153755	156268	169597	143103	128492	**117570**	17.84	30.68
Aggressive_1	819	1820	1662	2624	811	681	**345**	57.46	86.85
Normal_2	12	215	175	12	12	12	**0**	100	100
Total	710200	696013	758578	874895	695382	619009	**578320**	–	–

our Adaptive Semi-RF algorithm can improve the performance of its base model Random Forest and that of its semi-supervised version Semi-RF. In case of Normal_2, there is no error from Adaptive Semi-RF. Unfortunately, the others incorrectly identified a few instances in Normal_2 data set. This implies the appropriateness of the design choices of our algorithm such as the automatic determination of the prediction threshold to select the most confidently predicted instances from the unlabeled data set, the weighting of the distinct instances that is often recognized correctly, and the termination conditions for the iterative self-training mechanism of our algorithm.

Compared to Semi-RF_2/3, both Adaptive Semi-RF and Semi-RF can make better prediction with the lower number of incorrectly recognized instances in these data sets on a consistent basis. Such a result shows the importance of setting an appropriate value to the selection threshold so that a truly predicted data set can be selected and used for enhancing the training set. Otherwise, a cumulative error would occur and affect the effectiveness of the learnt classifier.

Compared to Semi-RF, Adaptive Semi-RF is also always better on these data sets. This fact confirms the effectiveness of our weighting scheme for the instances easily classified, i.e. often recognized correctly. By making the labeled data set adaptive to its different instances in the vector space, the influence of the instances wrongly selected from the unlabeled set and then added into the current labeled set is controlled and diminished. In general, Adaptive Semi-RF is effective for the multi-class classification task in a vector space.

5.2. Experiments on Clinical Notes

For an evaluation of the proposed approach to automatic abbreviation identification in clinical notes using our Semi-RF and Adaptive Semi-RF algorithms, several experiments and their results are discussed below. For feature extraction, the word embedding implementation in Word2VecJava [Medallia 2016] and a hand-coded dictionary composed of 1995 medical terms are used. Thanks to VanDon Hospital in Vietnam [Hospital 2016], clinical notes are provided from electronic medical records written in Vietnamese and English for medical terms.

Furthermore, we prepare three different types of clinical notes including care notes, treatment order notes, and treatment progress notes. The note types are different from each other in the number of records Record#, the number of sentences Sentence#, the number of tokens Token#, and the number of abbreviations Abbreviation#. Details about these clinical notes are given in Table III.

In the following tables, experimental results are displayed for the number of both abbreviation and non-abbreviation tokens incorrectly recognized in clinical notes. The experimental results on care notes are given in Table IV, the one on treatment order notes in Table V, and the one on treatment progress notes in Table VI.

Table III. Details about Clinical Notes and Abbreviations

Clinical Note Types	Care	Treatment Order	Treatment Progress
Patient#	2,000	2,000	2,000
Record#	12,100	4,175	4,175
Sentence#	8,978	39,206	13,852
Token#	52,109	325,496	138,602
Abbreviation#	3,031	24,693	7,641
Abbreviation%	5.82	7.59	5.51

As performing the abbreviation identification task with the proposed algorithms Semi-RF and Adaptive Semi-RF in a 12-dimension vector space, we examined more k folds for our clinical data sets, in particular, k = 2..12, 14, 16, 20 corresponding to 50% down to 5% labeled data used as an original labeled data set and the rest as an original unlabeled data set.

As recorded in Table IV, the experimental results on Care notes have shown the effectiveness of the proposed algorithms Adaptive Semi-RF and Semi-RF in comparison with that of the other algorithms. Adaptive Semi-RF outperforms the others at k=2, 4, 5, 9, 10, and 20 folds corresponding to 50%, 25%, 20%, 11%, 10%, and 5% of labeled data while Semi-RF provides the best results at k=3, 7, 12, 14, and 16 folds corresponding to 33%, 14%, 8%, 7%, and 6% of labeled data. It seems that a weighting scheme in Adaptive Semi-RF would work better when a set of labeled data is originally larger. For k=6, 8, and 11 folds corresponding to 17%, 13%, and 9% of labeled data, our proposed algorithms do not provide the best results. These cases stem from the imbalance of the added instances in the weighting and selection schemes. At this moment, our approach does not limit the number of added instances in these two schemes, leading to a large number of the instances selected and added into the training set of the first iteration. This number decreases over the time. Such a way results in two unfortunate factors: a skewed distribution of the enhanced training set and an early convergence of the learning process, which make the learnt classifier less effective. Indeed, for k=6, 8, and 11, the number of the selected non-abbreviations in these two schemes is much more than that of the selected abbreviations and most of the runs of Adaptive Semi-RF are performed in only 1 or 2 itera-tions. In contrast to our algorithms for k = 6, 8, and 11, Tri-training gets better due to a nice manner in handling the number of the added instances based on learning from noisy examples. In the future, we will take this problem into consideration. Nevertheless, Adaptive Semi-RF outperforms the others and especially improves Random Forest and Semi-RF_2/3 very much on average.

As for Treatment Order notes in Table V, Adaptive Semi-RF can provide the best results in most of the cases from 5% to 50%, in particular at k = 2, 3, 5, 7..10, 12, 14, 16, 20. It can also provide the second best result at k=11 while the best result comes from Semi-RF. Although not the best results at k=4, 6, the results from Semi-RF_2/3, Semi-RF, and Adaptive Semi-RF are close to the best results from Random Forest. In these two cases, it seems to be that semi-supervised learning algorithms are not applicable as Self-training (C4.5), Tri-training, Co-Forest, and our algorithms cannot correctly recognize so many tokens as possible. Like that on Care notes, the results produced by Adaptive Semi-RF are the best on average as compared to that by the others.

More stable on Treatment Progress notes in Table VI, our algorithms can recognize the tokens correctly in almost all the cases for k=2..12, 14, 16, 20. The best results are with Adaptive Semi-RF for k=2..6, 8, 10..12, 14, 16, 20 and with Semi-RF for k=7 and 9. However, the differences between the results from these two algorithms are little. Similarly to the results with Care notes and Treatment Order notes, the results from Adaptive Semi-RF are the best ones on average in these cases as compared to that

Table IV. The Number of Tokens Incorrectly Recognized in Care Notes

Fold#	Random Forest	Self-training	Tri-training	Co-Forest	Semi-RF_2/3	Semi-RF	Adaptive Semi-RF	$\%\Delta\_min$	$\%\Delta\_max$
2	36	32	35	34	34	35	**25**	21.88	30.56
3	70	83	74	104	67	**63**	72	-7.46	30.77
4	146	141	159	187	140	147	**116**	17.14	37.97
5	206	241	185	305	205	206	**161**	12.97	47.21
6	265	348	**232**	362	263	245	294	-26.72	18.78
7	311	424	356	499	309	**288**	347	-12.3	30.46
8	474	502	**364**	776	472	428	426	-17.03	45.1
9	591	653	607	795	591	537	**441**	25.38	44.53
10	723	742	635	1022	721	606	**476**	25.04	53.42
11	824	823	**601**	1477	821	680	713	-18.64	51.73
12	848	952	832	1388	845	**760**	804	3.37	42.07
14	1045	1337	1187	1851	1045	**943**	1002	4.11	45.87
16	1702	1630	1533	2268	1701	**1506**	1552	-1.24	31.57
20	2184	2411	2348	3016	2178	1975	**1722**	20.94	42.9
Total	9425	10319	9148	14084	9392	8419	**8151**	–	–

Table V. The Number of Tokens Incorrectly Recognized in Treatment Order Notes

Fold#	Random Forest	Self-training	Tri-training	Co-Forest	Semi-RF_2/3	Semi-RF	Adaptive Semi-RF	$\%\Delta\_min$	$\%\Delta\_max$
2	1108	1127	1130	1118	1105	1099	**1092**	1.18	3.36
3	2268	2391	2317	2273	2265	2244	**2209**	2.47	7.61
4	**3420**	3671	3665	3538	3422	3421	3477	-1.67	5.28
5	4582	4845	4854	4758	4582	4628	**4580**	0.04	5.64
6	**5827**	6102	6181	6163	5828	5850	5858	-0.53	5.23
7	7126	7418	7402	7192	7120	6944	**6907**	2.99	6.89
8	8516	8828	8886	8655	8515	8431	**8371**	1.69	5.8
9	9634	10338	10301	9876	9629	9629	**9424**	2.13	8.84
10	11456	11672	11707	11474	11452	10929	**10897**	4.85	6.92
11	12537	13064	12998	13004	12535	**12188**	12380	1.24	5.24
12	14209	14895	14778	14597	14205	13828	**13623**	4.1	8.54
14	16859	18044	17813	17374	16845	16608	**16472**	2.21	8.71
16	19428	21196	20970	21058	19426	19411	**19064**	1.86	10.06
20	26930	27897	27689	27145	26912	25503	**25320**	5.92	9.24
Total	143900	151488	150691	148225	143841	140713	139674	–	–

from the other algorithms on Treatment Progress notes. Generally speaking, Adaptive Semi-RF has improved Semi-RF_2/3, Semi-RF, and Random Forest very much.

In short, our work has provided an effective solution to automatic abbreviation identification with a comprehensive representation of each token in clinical notes and the two semi-supervised learning algorithms Semi-RF and Adaptive Semi-RF. This solution has been examined on the various real clinical note types and produced promising results in order to lay the foundations for determining the appropriate long forms of each correctly identified abbreviation. Regarding the effectiveness of the proposed semi-supervised learning algorithms, an intensive empirical study with the experiments on both benchmark data sets and clinical notes is conducted. Experi-mental results have shown that Adaptive Semi-RF is more effective than Semi-RF, Semi-RF_2/3, Random Forest, Self-training (C4.5), Tri-training, and Co-Forest on average.

6. CONCLUSIONS

Abbreviations have a variety of uses in taking clinical notes worldwide. As soon as clinical notes are archived along with their electronic medical records, abbreviations are regarded as explicit noises in the electronic medical records that need to be de-noised for more readability and sharability and further for computer-based data processing and analysis tasks. In this paper, we pay attention to the abbreviation identification

Table VI. The Number of Tokens Incorrectly Recognized in Treatment Progress Notes

Fold#	Random Forest	Self-training	Tri-training	Co-Forest	Semi-RF_2/3	Semi-RF	Adaptive Semi-RF	$\%\Delta\_min$	$\%\Delta\_max$
2	212	214	240	239	211	206	**191**	9.48	20.42
3	478	480	485	503	480	425	**411**	14.02	18.29
4	810	895	944	893	810	802	**761**	6.05	19.39
5	1207	1397	1460	1383	1190	1151	**1113**	6.47	23.77
6	1616	1649	1623	1831	1605	1503	**1397**	12.96	23.7
7	1890	2224	2154	2548	1888	**1780**	1785	5.46	29.95
8	2683	3088	3093	3142	2679	2368	**2366**	11.68	24.7
9	3073	3393	3323	3502	3063	**2607**	2749	10.25	21.5
10	3615	4295	4063	4199	3606	3278	**3257**	9.68	24.17
11	4362	4851	4617	4931	4336	3977	**3684**	15.04	25.29
12	4753	5669	5304	5894	4746	4227	**3897**	17.89	33.88
14	5937	6952	6466	7326	5910	5518	**5267**	10.88	28.11
16	7774	8922	8155	8801	7751	6663	**6659**	14.09	25.36
20	10228	12882	11977	12992	10211	9173	**9132**	10.57	29.71
Total	48638	56911	53904	58184	48486	43678	**42669**	–	–

task on electronic medical records to cope with the first phase of abbreviation resolution. The abbreviation identification task is formulated as a binary token-level classification task in a semi-supervised learning mechanism to identify abbreviations in free text of clinical notes in electronic medical records automatically.

In our proposed solution to this task, we do level-wise feature engineering in an unsupervised manner to represent each token in clinical notes in a vector space by making the most of several different aspects at token, sentence, and note levels. A resulting feature set forming the vector space is composed of orthographic, word length, and semantic features at the token level; contextual features at the sentence level using the continuous bag-of-word model; and occurrence feature of each token in a given note set at the note level using term frequency. We believe that a comprehensive set of level-wise features can help us distinguish the instances of abbreviations from the others of non-abbreviations. In addition to this feature vector representation, we define a novel adaptive semi-supervised learning approach using random forest models. In our approach, we make the most of random forest models and Tri-training in a self-training manner along with a new weighting scheme via adaptation to define a new adaptive semi-supervised learning algorithm called Adaptive Semi-RF and its traditional semi-supervised learning algorithm called Semi-RF. These resulting algorithms are simple and effective. Above all, they are not only parameter-free but also practical by using a current set of unlabeled data for advantage in constructing a classifier. For an empirical evaluation, many various experiments on both benchmark data sets and real clinical note sets have been conducted to compare our resulting algorithms (Semi-RF and Adaptive Semi-RF) with Random Forest, Self-training (C4.5), Tri-training (C4.5), and Co-Forest. The experimental results have shown that our solution is effective with the smallest number of instances incorrectly recognized on average. This implies that abbreviation identification can be tackled well for de-noising clinical notes with abbreviation resolution in our adaptive semi-supervised learning approach.

In the future, cleaning abbreviations from clinical notes by determining their correct long forms is one of our next steps to prepare electronic medical records for further computer-based data analysis and knowledge discovery. Parallel processing for our solution to abbreviation resolution is also regarded to speed up the task. Regarding the proposed adaptive semi-supervised learning algorithm, extending the space where the classification model is finalized is also of our interest. In addition, we plan to a new optimized stratified selection scheme for selecting the most confidently predicted

instances from the unlabeled data set in order to not only enlarge the training set but also maintain and enhance the prediction power of the final classifier.

ACKNOWLEDGMENTS

This work is funded by Vietnam National University at Ho Chi Minh City under the grant number B2016-42-01. In addition, we would like to thank John von Neumann Institute, Vietnam National University at Ho Chi Minh City, very much for providing us with a very powerful server machine to carry out the experiments. Moreover, this work was partially completed when the authors were working at Vietnam Institute for Advanced Study in Mathematics, Vietnam. Besides, our thanks go to Dr. Nguyen Thi Minh Huyen and her team at Hanoi University of Science, Vietnam National University, Hanoi, Vietnam, for external resources used in the experiments and also to the administrative board at VanDon Hospital for their real clinical data and support. Furthermore, the authors would like to thank the authors of the works [Li and Zhou 2007], [Zhou and Li 2005] very much for the source code of their algorithms in Java available on their website.

REFERENCES

Mehnaz Adnan, Jim Warren, , and Martin Orr. 2013. Iterative refinement of SemLink to enhance patient readability of discharge summaries. *Studies in Health Technology and Informatics* 188 (January 2013), 128–134.

UCI Archives. 2016. UCI Machine Learning Repository: Data Sets. (May 2016). Retrieved May 24, 2016 from http://archive.ics.uci.edu/

Jules J. Berman. 2004. Pathology abbreviated: a long review of short terms. *Archives of Pathology & Laboratory Medicine* 128, 3 (March 2004), 347–352.

Remco R. Bouckaert, Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann, Alex Seewald, and David Scuse. 2016. Weka 3: Data Mining Software in Java. (2016). Retrieved February 22, 2016 from http://www.cs.waikato.ac.nz/ml/weka

Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32.

Benjamin Collard and A. Royal. 2015. The use of abbreviations in surgical note keeping. *Annals of Medicine and Surgery* 4 (2015), 100–102.

Aron Henriksson, Hans Moen, Maria Skepstedt, Vidas Daudaravičius, and Martin Duneld. 2014. Synonym extraction and abbreviation expansion with ensembles of semantic spaces. *Journal of Biomedical Semantics* 5, 6 (2014), 1–25.

VanDon Hospital. 2016. A Set of Electronic Medical Records. (2016). Retrieved February 24, 2016 from Vietnam

Jong-Beom Kim, Heung-Seon Oh, Sang-Soo Nam, and Sung-Hyon Myaeng. 2013. Using candidate exploration and ranking for abbreviation resolution in clinical documents. In *Proceedings of the 2013 International Conference on Healthcare Informatics*. IEEE, 317–326.

Mi-Young Kim, Ying Xu, Osmar R. Zaiane, and Randy Goebel. 2015. Recognition of patient-related named entities in noisy tele-health texts. *ACM Transactions on Intelligent Systems and Technology* 6, 4, Article 59 (2015), 23 pages.

Seonho Kim and Juntae Yoon. 2015. Link-topic model for biomedical abbreviation disambiguation. *Journal of Biomedical Informatics* 53 (2015), 367–380.

Youngjun Kim, John Hurdle, and Stéphane M. Meystre. 2011. Using UMLS lexical resources to disambiguate abbreviations in clinical text. In *American Medical Informatics Association Annual Symposium Proceedings*. 715–722.

Markus Kreuzthaler and Stefan Schulz. 2015. Detection of sentence boundaries and abbreviations in clinical narratives. *BMC Medical Informatics and Decision Making* 15 (2015), 1–13.

Ming Li and Zhi-Hua Zhou. 2007. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man and Cybernetics - Part A: Systems and Humans* 37, 6 (2007), 1088–1098.

Yue Liu, Tao Ge, Kusum S. Mathews, Heng Ji, and Deborah L. McGuinness. 2015. Exploiting task-oriented resources to learn word embeddings for clinical abbreviation expansion. In *Proceedings of the 2015 Workshop on Biomedical Natural Language Processing*. 92–97.

William J. Long. 2003. Parsing free text nursing notes. In *American Medical Informatics Association Annual Symposium Proceedings*. 917.

Medallia. 2016. Word2VecJava. (2016). Retrieved February 22, 2016 from https://github.com/medallia/Word2VecJava

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Workshop Proceedings of the International Conference on Learning Representations*. 1–12.

Sungrim Moon, Bjoern-Toby Berster, Hua Xu, and Trevor Cohen. 2013. Word sense disambiguation of clinical abbreviations with hyperdimensional computing. In *American Medical Informatics Association Annual Symposium Proceedings*. 1007–1016.

Sungrim Moon, Bridget McInnes, and Genevieve B. Melton. 2015. Challenges and practical approaches with word sense disambiguation of acronyms and abbreviations in the clinical domain. *Healthcare Informatics Research* 21, 1 (2015), 35–42.

Sungrim Moon, Serguei Pakhomov, Nathan Liu, James O. Ryan, and Genevieve B. Melton. 2014. A sense inventory for clinical abbreviations and acronyms created using clinical notes and medical dictionary resources. *Journal of American Medical Informatics Association* 21 (2014), 299–307.

Serguei Pakhomov, Ted Pedersen, and Christopher G. Chute. 2005. Abbreviation and acronym disambiguation in clinical discourse. In *American Medical Informatics Association Annual Symposium Proceedings*. 589–593.

Jafar Tanha, Maarten van Someren, and Hamideh Afsarmanesh. 2015. Semi-supervised self-training for decision tree classifier. *International Journal of Machine Learning and Cybernetics* (January 2015), 1–16. DOI: http://dx.doi.org/10.1007/s13042-015-0328-7

Isaac Triguero, Salvador García, and Francisco Herrera. 2015a. SEG-SSC: A framework based on synthetic examples generation for self-labeled semi-supervised classification. *IEEE Transactions on Cybernetics* 45, 4 (April 2015), 622–634.

Isaac Triguero, Salvador García, and Francisco Herrera. 2015b. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems* 42, 2 (February 2015), 245–284.

Wilson Wong and David Glance. 2011. Statistical semantic and clinician confidence analysis for correcting abbreviations and spelling errors in clinical progress notes. *Artificial Intelligence in Medicine* 53 (2011), 171–180.

Yonghui Wu, Joshua C. Denny, S. Trent Rosenbloom, Randolph A. Miller, Dario A. Giuse, and Hua Xu. 2012. A comparative study of current clinical natural language processing systems on handling abbreviations in discharge summaries. In *American Medical Informatics Association Annual Symposium Proceedings*. 997–1003.

Yonghui Wu, S. Trent Rosenbloom, Joshua C. Denny, Randolph A. Miller, Subramani Mani, Dario A. Giuse, and Hua Xu. 2011. Detecting abbreviations in discharge summaries using machine learning methods. In *American Medical Informatics Association Annual Symposium Proceedings*. 1541–1549.

Yonghui Wu, Buzhou Tang, Min Jiang, Sungrim Moon, Joshua C. Denny, and Hua Xu. 2013. Clinical acronym/abbreviation normalization using a hybrid approach. In *ShARe/CLEF eHealth 2013 Challenge*. 1–9.

Yonghui Wu, Jun Xu, Yaoyun Zhang, and Hua Xu. 2015. Clinical abbreviation disambiguation using neural word embeddings. In *Proceedings of the 2015 Workshop on Biomedical Natural Language Processing*. 171–176.

Hua Xu, Peter D. Stetson, and Carol Friedman. 2007. A study of abbreviations in clinical notes. In *American Medical Informatics Association Annual Symposium Proceedings*. 822–825.

Hua Xu, Peter D. Stetson, and Carol Friedman. 2009. Methods for building sense inventories of abbreviations in clinical notes. *Journal of the American Medical Informatics Association* 16, 1 (2009), 103–108.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. 189–196.

Zhi-Hua Zhou and Ming Li. 2005. Tri-Training: exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering* 17, 11 (2005), 1529–1541.

Received February 2007; revised March 2009; accepted June 2009