VNU Journal of Science: Comp. Science & Com. Eng, Vol. 34, No. 2 (2018) 44-60

VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Abbreviation Detection in Vietnamese Clinical Texts

Chau Vo[1,*], Tru Cao[1], Bao Ho[2,3]

[1]*Ho Chi Minh City University of Technology, Vietnam National University, Ho Chi Minh City, Vietnam*
[2]*Japan Advanced Institute of Science and Technology, Japan*
[3]*John von Neumann Institute, Vietnam National University, Ho Chi Minh City, Vietnam*

Abstract

Abbreviations have been widely used in clinical notes because generating clinical notes often takes place under high pressure with lack of writing time and medical record simplification. Those abbreviations limit the clarity and understanding of the records and greatly affect all the computer-based data processing tasks. In this paper, we propose a solution to the abbreviation identification task on clinical notes in a practical context where a few clinical notes have been labeled while so many clinical notes need to be labeled. Our solution is defined with a semi-supervised learning approach that uses level-wise feature engineering to construct an abbreviation identifier, from using a small set of labeled clinical texts and exploiting a larger set of unlabeled clinical texts. A semi-supervised learning algorithm, Semi-RF, and its advanced adaptive version, Weighted Semi-RF, are proposed in the self-training framework using random forest models and Tri-training. Weighted Semi-RF is different from Semi-RF as equipped with a new weighting scheme via adaptation on the current labeled data set. The proposed semi-supervised learning algorithms are practical with parameter-free settings to build an effective abbreviation identifier for identifying abbreviations automatically in clinical texts. Their effectiveness is confirmed with the better Precision and F-measure values from various experiments on real Vietnamese clinical notes. Compared to the existing solutions, our solution is novel for automatic abbreviation identification in clinical notes. Its results can lay the basis for determining the full form of each correctly identified abbreviation and then enhance the readability of the records.

Received 26 August 2018, Revised 09 November 2018, Accepted 07 December 2018

Keywords: Electronic medical record, Clinical note, Abbreviation identification, Semi-supervised learning, Self-training, Random forest.

1. Introduction

In recent years, electronic medical records (EMRs) have become increasingly popular and significant in medical, biomedical, and healthcare research activities because of their advantages and the problems of the traditional medical records discussed in Shortliffe (1999) [21]. Experienced along the time, their successful adoption has been encouraged for their benefits in quality and patient care improvements in Cherry et al. (2011) [4]. These facts lead to a growing need for their sharing and utilization worldwide. Amenable for both human and computer-based understanding and

* Corresponding author. Email: chauvtn@hcmut.edu.vn
https://doi.org/10.25073/2588-1086/vnucsce.211

processing, the EMR contents must be clear and unambiguous. Nevertheless, free text in their clinical notes, called clinical text, often contains spelling errors, acronyms, abbreviations, synonyms, unfinished sentences, etc. described as explicit noises in Kim et al. (2015) [12].

Among these explicit noise types, abbreviations are pervasive for writing-time saving and record simplification. Unfortunately mentioned in Collard and Royal (2015) [5] and Shilo and Shilo (2018) [20], they result in misinterpretation and confusion of the content in the EMRs. They also greatly affect all the computer-based processing tasks. Therefore, identifying and replacing abbreviations with their correct long forms are necessary for enhancing the readability and shareability of the EMRs.

Many works have considered different tasks and purposes related to abbreviations. The Berman's list of 6 nonexclusive abbreviation groups in English medical records in Berman (2004) [3] has been widely used for clinical text processing. The abbreviation normalization and enhancing the readability of discharge summaries have been studied in Adnan et al. (2013) [1] and Wu et al. (2013) [30], respectively. Furthermore, Wu et al. (2012) [28] has examined three natural language processing systems (MetaMap, MedLEE, cTAKES) for handling abbreviations in English discharge summaries. Especially, the authors have confirmed that "accurate identification of clinical abbreviations is a challenging task". Indeed, in their most recent CARD framework in Wu et al. (2017) [31], abbreviation identification results in English clinical texts have been achieved with not very high F-measure: 0.755 on VUMC corpus and 0.291 on SHARe/CLEF one.

Certainly, it is more difficult to handle abbreviations in clinical texts than those in biomedical literature articles. In clinical texts, no long form of an abbreviation exists in the same text. In literature articles, however, the long form is typically provided next to the abbreviation (in parentheses) after which the abbreviation is used. In addition, more abbreviations with no convention are widely used in clinical texts.

Aware of the aforesaid necessity and challenges of abbreviation identification in clinical texts, many researchers have investigated several methods: word lists and heuristic rules in Xu et al. (2007) [32], supervised learning in Wu et al. (2017) [31], Kreuzthaler and Schulz (2015) [14], Wu et al. (2011) [29], and Xu et al. (2007) [32], and unsupervised approaches in Kreuzthaler et al. (2016) [13] including a statistical approach, a dictionary-based approach, and a combined one with decision rules.

Among these methods, the rule-based approaches cannot cover the ambiguity between abbreviations and non-abbreviations well. They also cannot thoroughly capture the surrounding context of each abbreviation in clinical texts. Machine learning-based approaches become advanced solutions to abbreviation identification. In Wu et al. (2011) [29] and Xu et al. (2007) [32], supervised learning has been utilized for abbreviation identification with decision trees C4.5, random forest models, support vector machines, and their combinations. Nevertheless, stated in Kreuzthaler et al. (2016) [13], it is not convenient for the supervised learning approach as this approach required clinical texts to be annotated. This requirement is costly in terms of effort and time.

In our view, semi-supervised learning is preferred in practice because a semi-supervised learning process can start with a smaller labeled data set and then iteratively exploit a larger unlabeled data set. Nevertheless, a semi-supervised learning approach has not yet been considered for abbreviation identification in any existing related works.

In this paper, we propose a new adaptive semi-supervised learning approach as an effective and practical solution to automatic abbreviation identification in clinical texts of EMRs. The proposed solution has the following key contributions.

The first contribution is level-wise feature engineering for a vector representation of each abbreviation or non-abbreviation, in a vector space. In particular, each token in clinical texts is comprehensively characterized at multiple levels of detail: token, sentence, and note.

The second one is the first semi-supervised learning method for abbreviation identification in clinical texts. Our method includes an appropriate semi-random forest algorithm, named Semi-RF, and its weighted semi-random forest version, named Weighted Semi-RF. These algorithms are defined with a parameter-free self-training mechanism, using random forest models in Breiman (2001) [3] and Tri-training in Zhou and Li (2005) [35].

As the third contribution, to the best of our knowledge, this is the first abbreviation identification work on Vietnamese EMRs. From the linguistic perspectives, the support of our work to the Vietnamese language of EMRs is adaptable and portable to other languages.

Experimental results on various real clinical note types have shown that our solution can produce the better Precision and F-measure values on average than the existing ones. Besides, all the differences in F-measure between Weighted Semi-RF and the other methods are statistically significant at the 0.05 level.

2. Related works

In this section, we introduce several existing works such as the works in Kreuzthaler et al. (2016) [13], Kreuzthaler and Schulz (2015) [14], Wu et al. (2011) [29], and Xu et al. (2007) [32] on abbreviation identification, and the works in Moon et al. (2014) [19], Xu et al. (2007) [32], and Xu et al. (2009) [33] on sense inventory construction for abbreviations.

Compared to the related works, our work aims at a more general solution to abbreviation identification. Indeed, Kreuzthaler et al. (2016) [13] and Kreuzthaler and Schulz (2015) [14] connected their solution to German abbreviation writing styles. Henriksson (2014) [10] considered the abbreviations with at most 4-letter lengths. Different from these works, our work has no limitation on either abbreviation writing styles or various lengths.

Besides, our work constructs a feature vector space from the inherent characteristics of each token in all the clinical notes at different levels: token, sentence, and note. Such level-wise feature engineering provides a comprehensive vector representation of each token. Moreover, a feature vector space is defined in our work, while Xu et al. (2007) [32] was not based on a vector space model, leading to different representations for clinical notes.

Furthermore, Wu et al. (2011) [29] used a local context based on the characteristics of the previous/next word of each current word and Xu et al. (2009) [33] used word forms of the surrounding words in a window size at the sentence level. Particularly for abbreviation identification, Wu et al. (2011) [29] formed several local context features in a single sentence. These local context features did not reflect the relationship between two consecutive words all over the notes. For sense inventory construction in Xu et al. (2009) [33], each feature word was associated with the modified Pointwise Mutual Information, representing a co-occurrence-based association between the feature word and its target abbreviation.

Different from the works in Wu et al. (2011) [29] and Xu et al. (2009) [33], our work handles the global context of each token additionally at the note level. The global context is represented by our *cross-document* features. The *cross-document* features are captured to represent a word based on its context words. Both syntactic relatedness and semantic relatedness between a word and its context words are achieved in a distributed representation of each word, from all the sentences in a note set using a continuous bag-of-words model in Mikolov et al. (2013) [18].

Regarding abbreviation identification, the work inXu et al. (2007) [32] used word lists and heuristic rules. Some works followed a

supervised learning approach in Wu et al. (2017) [31], Kreuzthaler and Schulz (2015) [14], Wu et al. (2011) [29], and Xu et al. (2007) [32] using decision trees C4.5, random forest, support vector machines, and their combination. A more recent work in Kreuzthaler et al. (2016) [13] proposed an unsupervised learning approach such as a statistical approach, a dictionary-based approach, and a combined one with decision rules. None of the aforementioned works was based on a semi-supervised learning approach. By contrast, our work defines a semi-supervised learning approach for constructing an abbreviation identifier on clinical texts.

Above all, each related work conducted evaluation experiments using its own data set. Kreuzthaler et al. (2016) [13] and Kreuzthaler and Schulz (2015) [14] used German clinical texts while Wu et al. (2012) [28], Wu et al. (2011) [29], and Xu et al. (2007) [32] used English ones. None of them is an available benchmark clinical data set for abbreviation identification. Therefore, it is difficult for empirical comparisons on different clinical texts in other languages.

In summary, our work is the first one that proposes a semi-supervised learning approach to abbreviation identification in clinical texts with two new semi-supervised learning algorithms, Semi-RF and Weighted Semi-RF, using level-wise feature engineering for a more comprehensive representation.

3. The proposed method for abbreviation identification in clinical texts

In this section, we define an abbreviation identification task along with level-wise feature engineering for clinical texts. After that, we propose an adaptive semi-supervised learning approach to abbreviation identification in clinical texts with two semi-supervised learning algorithms, Semi-RF and Weighted Semi-RF. Their discussions are also given.

3.1. Task definition

In this work, we formulate the abbreviation identification task as a binary classification task on free texts in the clinical notes. Given a set of labeled clinical texts and another one of unlabeled clinical texts, the task first builds an abbreviation identifier and then uses this identifier to identify each token in the given unlabeled set as abbreviation (class = 1) or non-abbreviation (class = 0).

For illustration, one sentence from a treatment order of a doctor for a patient written in a Vietnamese clinical note is given below:

(Tiêm TM) – TD: M – T – HA – NT 3h/lần.

The sentence is rewritten in English as follows:

(Inject into a vein) – Track: Pulse – Temperature – Blood Pressure – Breath Speed 3 hours/time.

It is realized that in this treatment order, the sentence is not a complete standard one and includes many abbreviations. Also, there are abbreviations of both medical and non-medical terms. The abbreviations for medical terms are "TM", "M", "T", "HA", "NT" and those for non-medical terms are "TD" and "3h".

If this sentence is in a set of labeled clinical texts, their tokens are labeled as shown in Figure 1.

If the sentence is in a set of new (unlabeled) clinical texts, its tokens need to be identified as 0 or 1, for non-abbreviation or abbreviation, respectively.

To be processed in the task, each token must be represented in a computational form. In our work, a vector space model is used. Each token is characterized by a vector of p features corresponding to p dimensions of the space.

A vector corresponding to a token in the labelled set is used in abbreviation identifier construction.

Token	(	Tiêm	TM	)	-	TD	:	M	-	T	-	HA	-	NT	3h	/	lần	.	CSC2	.
Label	0	0	1	0	0	1	0	1	0	1	0	1	0	1	1	0	0	0	1	0

Figure 1. A sample treatment order sentence with tokens and their labels.

On the other hand, a vector corresponding to a token in the unlabeled set has no class value. Its class value needs to be predicted by an abbreviation identifier.

If at the beginning, a labeled set is available, the task can be performed in a supervised learning or semi-supervised learning mechanism. In practice, a semi-supervised learning mechanism is preferred in the following conditions. An available labeled set is small and thus, might not be sufficient for an effective supervised learning process. Meanwhile, there exists a larger unlabeled set. It would be helpful if this unlabeled set can be exploited for more effectiveness.

In our work, we approach this abbreviation identification task in a semi-supervised learning mechanism with our semi-supervised learning algorithms. These algorithms can facilitate the task in a parameter-free configuration scheme.

3.2. Level-wise feature engineering for clinical texts in a vector space

In this subsection, we first design the vector structure of each token and then process the clinical texts to generate its vector by extracting and calculating its feature values. Figure 2 depicted these consecutive steps as (1). Unsupervised Feature Vector Space Building and (2). Feature Value Extraction.

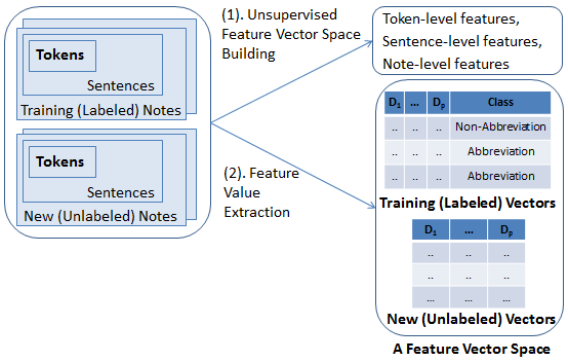

Figure 2. Representing clinical notes in electronic medical records in a vector space.

In step (1), we consider the features at the token, sentence, and note levels because clinical notes include sentences each of which contains many tokens attained with tokenization. In such a multilevel view, level-wise feature engineering captures many different aspects of each token from the finest token and sentence levels to the coarsest note one.

In step (2), each element of the vector is determined according to the characteristics of the token at these levels. A vector corresponding to a labeled token is annotated additionally.

Formally, a token in a clinical note is represented in the form of a vector:

$$X = (x^t_1, \ldots, x^t_{tp}, x^s_1, \ldots, x^s_{sp}, x^n_1, \ldots, x^n_{np}) \quad (1)$$

in a vector space of p dimensions where x^t_i is a value of the i-th feature at the token level for $i = 1..tp$, x^s_j is a value of the j-th feature at the sentence level for $j = 1..sp$, and x^n_k is a value of the k-th feature at the note level for $k = 1..np$; and tp is the number of token-level features, sp is the number of sentence-level features, and np is 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: word form with orthographic properties, word length, and semantics (e.g. being a medical term or an acronym of any medical term). The corresponding token-level features include: *AllAlphabeticChars*, *AnyAlphabeticChar*, *AnyAlphabeticCharAtBeginning*, *AllDigits*, *AnyDigit*, *AnyDigitAtBeginning*, *AnySpecialChar*, *AnyPunctuation*, *AllConsonants*, *AnyConsonant*, *AllVowels*, *AnyVowel*, *AllUpperCaseChars*, *AnyUpperCaseCharAtBeginning*, *Length*, *inDictionary*, *isAcronym*.

At the *sentence* level, many contextual features are defined from the surrounding words of each token in its sentence. We also used the local contextual features of the previous and next tokens in a 3-token window proposed in Wu et al. (2011) [29].

At the *note* level, occurrence of each token in clinical notes is considered as a note-level feature. We use a term frequency *TermFrequency* to capture the number of its occurrences. Additionally mentioned in Long (2003) [17], many abbreviations have been commonly used but many are dependent on

context, leading to the importance of capturing the surrounding context of each abbreviation. In our work, we enrich the context of each token by our *cross-document* features for its global context. Consistent with the local context, the global context is defined by the *cross-document* features of the previous, current and next tokens in a 3-token window.

To obtain the values for the *cross-document* features, we use a word embedding vector of each token. Indeed, their values stem from a distributed representation of a token in Mikolov et al. (2013) [18] based on their surrounding tokens in all the given texts, as a vector using a continuous bag-of-words model.

3.3. The proposed semi-supervised learning algorithm

3.3.1. Algorithm characteristics

Defined in Breiman (2001) [3], random forest is a well-known ensemble algorithm. One of its improved versions was defined in González et al. (2015) [9] for more effectiveness with monotonicity constraints. Meanwhile, Tri-training in Zhou and Li (2005) [35] is an advanced parameter-free co-training style algorithm. Introduced in Yarowsky (1995) [34], the self-training approach is one of the simplest semi-supervised learning algorithms. Nevertheless, the users must set a "correct" value to the probability threshold for newly labeled instance selection.

Bringing random forest and Tri-training to the self-training approach, our work proposes a new adaptive semi-supervised learning approach with two algorithms: Semi-RF and Weighted Semi-RF. Semi-RF combines Tri-training and a random forest in a self-training style, while Weighted Semi-RF is its adaptive version with a weighting scheme for proper treatment of the labeled instances in the learning process. They inherit the strengths of random forest and Tri-training and overcome the weaknesses of the self-training approach. Different from the existing algorithms such as Dong et al. (2016) [6], Joachims (1999) [11], Li and Zhou (2007) [16], Tanha et al. (2015) [22], and Triguero et al. (2015) [24], our algorithms are developed with the following foundations:

• The resulting algorithms are parameter-free based on Tri-training, effective based on random forest models, but simple in the self-training style.

• The final classifier is in fact a random forest model with its inherent effective, robust, and non-overfitting advantages.

• For Weighted Semi-RF, differentiating between the instances in both labeled and unlabeled sets is maintained in the learning process by favoring the truly labeled instances over those wrongly labeled instances in a weighting scheme.

Specifically, the algorithms are proposed in the form of self-training, using the random forest model of three random trees with $(\lfloor \log(p) \rfloor + 1)$ random features. This feature number is based on the study of Breiman (2001) [3]. Three random trees play the role of three classifiers in Tri-training so that the probability threshold can be automatically defined to select the most confidently predicted instances from a current unlabeled set.

Compared to Tri-training, our algorithms are different in the following instance selection. Each instance is considered to be correctly predicted and then selected if the agreement of these three random trees is achieved at the highest level. It can contribute to the learning process of each random tree if included in bootstrap sampling. Therefore, bootstrap sampling is retained 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 labeled set including both truly labeled and newly labeled instances at the beginning of each round. This weighting scheme makes the current labeled set adaptive to such truly labeled and newly easily predicted instances. Further, it will shift the prediction of our final classifier towards these instances and constrain the hard newly predicted instances that might be wrongly labeled.

Moreover, the optimization of our algorithms is based on the generalization of the final random forest model over the original labeled set containing true labels that are certainly known. This forms the stable convergence of our algorithms.

3.3.2. Algorithm details

For details, the pseudo-code of our Weighted Semi-RF algorithm is given in Figure 3. Its original Semi-RF algorithm is a simpler version without the weighting scheme via adaptation on the labeled set. Details of the weighting scheme are given in Figure 4 and details of the selection scheme of the most confidently predicted instances from the current unlabeled set are given in Figure 5.

In Figure 3 in an iterative manner, our Weighted Semi-RF algorithm performs below.

In line (5), the 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. This current classifier is then evaluated on the original set of labeled data. If its error rate is less than the previous error rate set previously, i.e. its prediction power is better, the previous error rate and the previous classifier will be updated with the new current ones. Otherwise, the previous classifier has been the best so far and thus will be returned as a resulting classifier *C*.

If improvement is found, exploiting unlabeled data is considered from line (11) to line (18). If the current set of unlabeled data is not empty, we use the current classifier to predict the 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 Figure 3, a resulting classifier *C* is obtained with two termination conditions: no element in the current set of unlabeled data in line (17) or no improvement on the prediction power of the resulting classifier on the original set of labeled data in line (20). The first termination condition is based on the general rationale behind the 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 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.

Shown in Figure 3, the entire learning process of our algorithms is in a self-training mechanism, but the use of the 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 weighting scheme via adaptation on the current labeled data set. As two main advantages, our weighting and selection schemes are discussed.

(i). Weighting Scheme

First, our *weighting* scheme makes *adaptation* on the current labeled set in the *k*-fold cross validation style by weighting each instance in favor of its being truly labeled. For example, to make *adaptation* on the current set of labeled instances into 5 similarly-sized folds (*k*=5), 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 correctly predicted instances of the remaining fold are added into the adapted current set of labeled instances, returned as a result of the weighting scheme.

Weighting is different for an instance that has a true label given in the original labeled set 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.

As the weighting scheme considers truly labeled instances, it is questionable that overfitting occurs in our learning process. This is not a fact in Weighted Semi-RF due to the characteristics of random forest models. Mentioned in Li and Zhou (2007) [16], the diversity of the random trees in the random forest is maintained even if their training data sets are similar. As a result, only truly labeled instances have mainly contributed to our learning process, while probably wrongly labeled instances that have been added into the training data set would have had less.

(ii). Selection scheme

Second, the most confidently predicted instance selection scheme is described.

Let us denote m be the number of classes and t be the number of random trees in the random forest model. The prediction score of a current instance X^* is calculated below:

Weighted Semi-RF: The proposed adaptive semi-supervised learning algorithm on both labeled and unlabeled data in the *p*-dimension vector space
Input:
lSet: a labeled set which is originally given in the *p*-dimension vector space
uSet: an unlabeled set which is originally given in the *p*-dimension vector space
Output:
C: a resulting classifier
Process:
(1). Set a previous error rate *Previous_error_rate* to 0.5
(2). Assign *lSet* as a current set *clSet* which contains all instances with known labels
(3). Assign *uSet* as a current set *cuSet* which contains all instances with unknown labels
(4). Repeat until the termination conditions are met:
(5). Weighting the labeled instances via *adaptation* on the labeled set *clSet* to obtain an adaptive labeled set *clSet_a*
(6). Build a current random forest *Current_RF* of three random trees with ($\lfloor \log(p) \rfloor + 1$) random features on *clSet_a*
(7). Compute a current error rate *Current_error_rate* by evaluating Current_RF on *lSet*
(8). If *Previous_error_rate* > *Current_error_rate* then
(9). *Previous_error_rate* = *Current_error_rate*
(10). Save the current random forest *Current_RF* as a previous random forest *Previous_RF*
(11). If *cuSet* is not empty then
(12). Predict a label of each instance in *cuSet* using *Current_RF*
(13). Select *a set sSet of the most confidently predicted instances* from *cuSet*
(14). Update *clSet_a* to *clSet* by including *sSet*
(15). Update *cuSet* by excluding *sSet*
(16). Else
(17). Return the current random forest *Current_RF* as a resulting classifier *C*
(18). End If
(19). Else
(20). Return the previous random forest *Previous_RF* as a resulting classifier *C*
(21). End If
(22). End Repeat

Figure 3. Weighted Semi-RF - the proposed adaptive semi-supervised learning algorithm.

• 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:

$$P_j(C_i|X^*) = \frac{k}{N} \quad (2)$$

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.

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

$$Score(X^*) = max\ \{P(C_i|X^*) \text{ for } i=1..m\} \quad (3)$$

$$Class(X^*) = argmax_{Ci}\ \{\ P(C_i|X^*) \text{ for } i=1..m\ \} \quad (4)$$

Where a class distribution score of a class C_i for X^* by the random forest model is calculated as $P(C_i|X^*) = \Sigma_{j=1..t}P_j(C_i|X^*)$ and normalized as:

$\forall i$=1..m, $0 \le P(C_i|X^*) \le 1$ and $\Sigma_{i=1..m}P(C_i|X^*) = 1$.

In the selection scheme, if the prediction score of the instance X^* is 1, then X^* is selected.

Weighting Scheme: *Weighting* the labeled instances via *adaptation* on a current set *clSet* of labeled instances in the 5-fold cross validation scheme
Input:
clSet: a current set which contains all instances with known labels in the *p*-dimension vector space
Output:
clSet_a: a current set which contains all instances with known labels after adaptation in the *p*-dimension vector space
Process:
(1). *clSet_a* = *clSet*
(2). Do stratified random sampling without replacement on *clSet* into 5 folds that have similar size (almost the same size)
(3). For each fold *f* do
(4). 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*
(5). Evaluate *aRF* on the current fold *f*
(6). Update *clSet_a* with the instances of the current fold *f* correctly recognized by *aRF*
(7). End For
(8). Return *clSet_a*

Figure 4. Weighting Scheme - *weighting* the labeled instances via *adaption* on a current set *clSet* of labeled instances

Selection Scheme: *Selecting* a set *sSet* of the *most confidently predicted instances* from the current set *cuSet* of unlabeled instances
Input:
cuSet: a current set which contains all instances with unknown labels in the *p*-dimension vector space
Output:
sSet: a selected set of the most confidently predicted instances in the *p*-dimension vector space
Process:
(1). For each instance X^* in *cuSet* do
(2). Calculate a prediction score for the current instance X^*
(3). If its prediction score = 1 then
(4). Add this current instance X^* into *sSet*
(5). End If
(6). End For
(7). Return *sSet*

Figure 5. Selection Scheme - *selecting* a set *sSet* of the *most confidently predicted instances* from the current set *cuSet* of unlabeled instances.

Its predicted label is now considered true. The reason for the threshold value of 1 is reducing a chance of selecting a wrongly predicted instance. Indeed, a wrong prediction occurs only if at least one of the random trees misclassifies the instance.

3.3.3. Discussions

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. Weighted Semi-RF is its adaptive version, which enhances the training set with the weighting scheme. Compared to Semi-RF, Weighted Semi-RF has reduced the influence of the selected wrongly predicted instances in the learning process. Besides, these algorithms are applicable to classifier construction from a small labeled set in practice. Above all, they are parameter-free with no restriction on parameter configurations.

4. Empirical evaluation

4.1. Data sets

In our work, all the experiments were conducted on three clinical note sets including Care and Treatment clinical notes in Table 1. Thanks to Hospital in Vietnam (Hospital (2016) [25]), these clinical notes are provided from real EMRs written in Vietnamese with some English medical terms.

After a tokenization process is performed with the separators such as space and tab, these clinical notes are manually annotated. Furthermore, we randomly select only 565 distinct sentences for each type in one processing batch. Besides, we made 30 random selections to avoid randomness. Thus, every measure value in our results is an average of the corresponding results from 30 executions. Their information is described in Table 2.

4.2. Experiment settings

The program is written in Java using Weka 3 (Weka3 (2016) [26]). For feature extraction, the word embedding library in Word2VecJava (Word2VecJava (2016) [27]) is used. In addition, a hand-coded dictionary including 1995 English/Vietnamese medical terms is prepared and used. From the linguistic perspectives, the support of our work to Vietnamese can be adaptable and portable to other languages with their own dictionaries.

For evaluation, a full set of features at all the three levels of details was used. Random Forest in Breiman (2001) [3], C4.5, Self-training in Yarowsky (1995) [34], Tri-training in Zhou and Li (2005) [35], Co-Forest in Li and Zhou (2007) [16], Semi-RF_2/3, Semi-RF, and Weighted Semi-RF are examined.

Among these algorithms, Random Forest and C4.5 are included because they are base models in the semi-supervised learning algorithms in our experiments. Tri-training with C4.5, Self-Training with C4.5, and Co-Forest are selected according to the empirical study of Triguero et al. (2015) [23]. We also record the performance of Semi-RF_2/3 which is Semi-RF using the threshold of 2/3 to check how effective our most confidently predicted instance selection scheme is.

Regarding performance measures, Precision, Recall, and F-measure are used to record the effectiveness of each method and show how well abbreviations can be identified. The higher measure value implies the better method. Besides, One-Way ANOVA in Fisher (1934) [8] has been done to determine if there exist significant differences in F-measure among compared groups at the 0.05 level of significance. In addition, Bonferroni post-hoc test in Dunn (1961) [7] with Levene's test in Levene (1960) [15] for equal variances at the 0.05 level of significance has been used for specific significant differences. In the following Tables 3, 4, and 5, the averaged results were reported. A summary of statistical test results is given in Table 6 to compare the averaged F-measure values of Weighted Semi-RF and those of the others. In Table 6, we used "Weighted Semi-RF>Y" to denote that Weighted Semi-RF outperformed the "Y" methods with significantly better F-measure values.

For reliable accuracy estimation, we use the *k*-fold cross validation scheme in the context of semi-supervised learning. In particular, *k* is 2, 4, 5, 10, or 20 corresponding to 50%, 75%, 80%, 90%, or 95% unlabeled data.

Table 1. Details about all the clinical note sets and abbreviations.

Note Type	Care	Treatment Order	Treatment Progress
Number of patients	2,000	2,000	2,000
Number of records	12,100	4,175	4,175
Number of sentences	8,978	39,206	13,852
Number of tokens	52,109	325,496	138,602
Number of abbreviations	3,031	24,693	7,641
Percentage of abbreviations (%)	5.82	7.59	5.51

Table 2. Details about the selected clinical note sets.

Note type	Care	Treatment Order	Treatment Progress
Averaged number of sentences	565	565	565
Averaged number of tokens	4119	6954	8002
Averaged number of tokens per sentence	7.29	12.31	14.16
Averaged number of abbreviations per sentence	1.11	2.70	2.16
Number of distinct abbreviations	49	117	199
Averaged percentage of non-abbreviations	83.74 %	78.08 %	84.71 %
Averaged percentage of abbreviations	16.26 %	21.92 %	15.29 %

4.3. Experimental results and an evaluation for the proposed method

Via the experimental results, our methods always outperform the others with the best Precision and F-measure values for all the clinical texts. Nevertheless, our methods produced the best Recall values for the Care and Order clinical texts and just the second best Recall values for the Progress clinical texts when there is about less than 90% unlabeled data. In those cases, Tri-training or Self-training got the best Recall values for the Progress clinical texts. As there are 90% and 95% unlabeled data, our methods can obtain the best Precision and F-measure values and almost the second best Recall values consistently for all the clinical texts while the best Recall values come from Tri-training. This is understandable as Tri-training handled the number of the instances added into the training set nicely based on the learning from noisy examples. In contrast, our methods selected all the instances based on the probability threshold, leading to an imbalance in the added instance set including more non-abbreviations and fewer abbreviations.

Indeed, Weighted Semi-RF can produce from 0.26% to 1.52% better Precision values than the highest ones by the others and from 2.37% to 9.06% better Precision values than the lowest ones by the others. As for Recall, they are from -2.12% to 0.99% compared to the highest ones by the others and from 0.4% to 4.68% compared to the lowest ones by the others. For F-measure, they are from 0.33% to 1.36% compared to the highest ones by the others and from 1.51% to 6.53% compared to the lowest ones by the others. On balance, our methods outperform the others with the better F-measure values in all the cases.

In Table 6, almost all the differences in F-measure between Weighted Semi-RF and the others are significant at the level of 0.05. It is confirmed that Weighted Semi-RF is effective for abbreviation identification in the clinical texts.

Among our methods, Weighted Semi-RF outperforms Semi-RF and Semi-RF outperforms Semi-RF-2/3 in almost all the

cases. In Table 6, statistical test results confirmed the effectiveness of Weighted Semi-RF compared to Semi-RF_2/3 with better F-measure values in almost all the cases. These facts imply appropriate design of our algorithms. In particular, the probability threshold setting based on the agreement of all the base learners is more stable than the one with the agreement in Tri-training or user-specified in Self-training. In addition, consideration on the influences of each instance in the training set is important and our weighting scheme is effective in that regard.

In short, our work has provided an effective solution to automatic abbreviation identification with Semi-RF and Weighted Semi-RF. It has been examined on the various real clinical texts and produced promising results to lay the foundations for determining the appropriate long forms of each correctly identified abbreviation.

Table 3. Averaged results for method evaluation on care notes

Note Type	Unlabeled Data	Method	Precision	Recall	F-measure
Care	50%	C4.5	98.48	96.6	97.53
		Random Forest	98.8	96.97	97.88
		Self-training	98.6	96.66	97.61
		Co-Forest	97.15	96.51	96.82
		Tri-training	98.3	97.13	97.71
		Semi-RF_2/3	98.91	96.96	97.92
		Semi-RF	99.14	97.33	98.23
		Weighted Semi-RF	99.24	97.49	98.36
	75%	C4.5	97.78	95.59	96.67
		Random Forest	97.86	95.89	96.86
		Self-training	97.73	95.64	96.68
		Co-Forest	95.23	94.22	94.72
		Tri-training	96.97	96.43	96.7
		Semi-RF_2/3	97.94	95.92	96.92
		Semi-RF	98.62	96.33	97.46
		Weighted Semi-RF	98.71	96.48	97.58
	80%	C4.5	97.46	95.23	96.33
		Random Forest	97.58	95.26	96.4
		Self-training	97.49	95.33	96.4
		Co-Forest	94.37	94.31	94.34
		Tri-training	96.44	96.24	96.34
		Semi-RF_2/3	97.67	95.33	96.48
		Semi-RF	98.38	96.1	97.23
		Weighted Semi-RF	98.43	96.34	97.37
	90%	C4.5	95.81	94.06	94.92
		Random Forest	96.17	93.32	94.72
		Self-training	95.62	94.49	95.05
		Co-Forest	91.98	91.28	91.62
		Tri-training	94.83	94.8	94.81
		Semi-RF_2/3	96.26	93.38	94.79
		Semi-RF	97.21	94.6	95.89
		Weighted Semi-RF	97.33	95.01	96.15
	95%	C4.5	94.29	91.62	92.93
		Random Forest	94.39	90	92.13
		Self-training	94.35	91.61	92.95
		Co-Forest	88	87.98	87.98
		Tri-training	93.41	92.71	93.05
		Semi-RF_2/3	94.5	90.03	92.2

Note Type	Unlabeled Data	Method	Precision	Recall	F-measure
		Semi-RF	96.38	91.75	94
		Weighted Semi-RF	96.43	92.66	94.51
	Average	C4.5	96.36	94.25	95.29
		Random Forest	96.72	93.75	95.21
		Self-training	96.33	94.5	95.4
		Co-Forest	92.67	92.21	92.43
		Tri-training	95.54	95.08	95.3
		Semi-RF_2/3	96.8	93.79	95.27
		Semi-RF	97.7	94.88	96.27
		Weighted Semi-RF	97.75	95.27	96.49

Table 4. Averaged results for method evaluation on treatment order notes

Note Type	Unlabeled Data	Method	Precision	Recall	F-measure
Treatment Order	50%	C4.5	98.17	98.24	98.2
		Random Forest	98.42	98.06	98.24
		Self-training	98.14	98.32	98.23
		Co-Forest	97.22	97.33	97.27
		Tri-training	98.08	98.31	98.2
		Semi-RF_2/3	98.49	98.1	98.29
		Semi-RF	98.79	98.5	98.64
		Weighted Semi-RF	98.74	98.54	98.64
	75%	C4.5	97.12	96.98	97.05
		Random Forest	97.21	96.8	97
		Self-training	97.12	97.06	97.09
		Co-Forest	95.4	95.71	95.55
		Tri-training	97.03	97.3	97.16
		Semi-RF_2/3	97.28	96.87	97.07
		Semi-RF	97.96	97.45	97.7
		Weighted Semi-RF	98.1	97.63	97.86
	80%	C4.5	96.85	96.55	96.7
		Random Forest	97.12	96.3	96.7
		Self-training	96.84	96.63	96.74
		Co-Forest	94.71	95.35	95.03
		Tri-training	96.61	96.86	96.73
		Semi-RF_2/3	97.19	96.34	96.76
		Semi-RF	97.75	96.92	97.33
		Weighted Semi-RF	97.88	97.26	97.57
	90%	C4.5	95.17	95.12	95.14
		Random Forest	95.74	94.14	94.93
		Self-training	95.28	95.03	95.15
		Co-Forest	92.23	93.09	92.65
		Tri-training	94.94	95.42	95.18
		Semi-RF_2/3	95.83	94.18	95
		Semi-RF	96.94	94.82	95.87
		Weighted Semi-RF	96.99	95.28	96.13
	95%	C4.5	92.79	92.76	92.77
		Random Forest	94.07	91.42	92.72
		Self-training	92.7	93.12	92.91
		Co-Forest	88.51	89.64	89.07
		Tri-training	92.56	93.39	92.97
		Semi-RF_2/3	94.16	91.45	92.78

Note Type	Unlabeled Data	Method	Precision	Recall	F-measure
		Semi-RF	96	92.14	94.03
		Weighted Semi-RF	96.15	92.87	94.48
	Average	C4.5	96.02	95.93	95.97
		Random Forest	96.51	95.34	95.92
		Self-training	96.02	96.03	96.02
		Co-Forest	93.61	94.22	93.91
		Tri-training	95.84	96.26	96.05
		Semi-RF_2/3	96.59	95.39	95.98
		Semi-RF	97.49	95.97	96.72
		Weighted Semi-RF	97.57	96.31	96.94

Table 5. Averaged results for method evaluation on treatment progress notes

Note Type	Unlabeled Data	Method	Precision	Recall	F-measure
Treatment Progress	50%	C4.5	98.35	98.25	98.3
		Random Forest	98.52	97.73	98.12
		Self-training	98.33	98.31	98.32
		Co-Forest	96.72	96.86	96.79
		Tri-training	98.35	98.28	98.31
		Semi-RF_2/3	98.6	97.83	98.21
		Semi-RF	98.95	98.1	98.53
		Weighted Semi-RF	99.06	98.25	98.65
	75%	C4.5	97.49	97.19	97.34
		Random Forest	97.69	96.5	97.09
		Self-training	97.53	97.24	97.39
		Co-Forest	95.05	95.23	95.14
		Tri-training	97.24	97.43	97.33
		Semi-RF_2/3	97.8	96.56	97.17
		Semi-RF	98.48	97.05	97.76
		Weighted Semi-RF	98.48	97.39	97.93
	80%	C4.5	97.25	96.72	96.98
		Random Forest	97.54	96.01	96.77
		Self-training	97.17	96.84	97
		Co-Forest	94.77	94.72	94.74
		Tri-training	97.09	97.03	97.06
		Semi-RF_2/3	97.67	96.05	96.85
		Semi-RF	98.41	96.71	97.55
		Weighted Semi-RF	98.46	96.96	97.7
	90%	C4.5	95.66	95.69	95.68
		Random Forest	96.41	93.42	94.89
		Self-training	95.61	95.86	95.74
		Co-Forest	91.87	91.73	91.8
		Tri-training	95.25	95.91	95.57
		Semi-RF_2/3	96.54	93.48	94.98
		Semi-RF	97.63	94.27	95.92
		Weighted Semi-RF	97.75	94.68	96.19
	95%	C4.5	93.42	91.64	92.52
		Random Forest	94.34	89.13	91.66
		Self-training	93.46	92.06	92.75
		Co-Forest	87.83	87.76	87.79
		Tri-training	92.87	92.74	92.8
		Semi-RF_2/3	94.52	89.15	91.75

Note Type	Unlabeled Data	Method	Precision	Recall	F-measure
		Semi-RF	96.75	89.82	93.15
		Weighted Semi-RF	96.89	90.62	93.65
	Average	C4.5	96.43	95.9	96.16
		Random Forest	96.9	94.56	95.71
		Self-training	96.42	96.06	96.24
		Co-Forest	93.25	93.26	93.25
		Tri-training	96.16	96.28	96.22
		Semi-RF_2/3	97.03	94.61	95.79
		Semi-RF	98.04	95.19	96.58
		Weighted Semi-RF	98.13	95.58	96.83

Table 6. Statistical test results for method evaluation at the 0.05 significance level with respect to F-measure

Note Type	Unlabeled Data	Weighted Semi-RF	Semi-RF	The Others
Care	50%	> The Others	No	No
	75%	> The Others	No	No
	80%	> The Others	No	No
	90%	> The Others	No	No
	95%	> The Others	No	No
Treatment Order	50%	> The Others	No	No
	75%	> The Others	No	No
	80%	> The Others	No	No
	90%	> The Others	No	No
	95%	> Semi-RF > The Others	No	No
Treatment Progress	50%	> The Others	No	No
	75%	> The Others	No	No
	80%	> The Others	No	No
	90%	> The Others	No	No
	95%	> Semi-RF > The Others	No	No

5. Conclusions

In this paper, we consider the abbreviation identification task on free texts of the clinical notes in EMRs. The task is formulated as a binary classification task in a semi-supervised learning mechanism. In order to perform this task, we do level-wise feature engineering to represent each token in clinical notes in a vector space by examining the different aspects at token, sentence, and note levels. Using this feature vector representation, a novel adaptive semi-supervised learning approach is proposed. A new adaptive semi-supervised learning algorithm, Weighted Semi-RF, and its traditional semi-supervised learning algorithm, Semi-RF, are defined by combining the random forest model and Tri-training in a self-training manner along with a new weighting scheme via adaptation.

These algorithms are simple, parameter-free, and practical by utilizing a current larger set of unlabeled data in constructing a classifier. The experimental results have confirmed that our solution is effective with the better Precision and F-measure values on average compared to some existing ones. This shows that abbreviation identification can be tackled well in our approach.

In practice, the proposed solution is the first attempt to deal with abbreviation identification for real Vietnamese EMRs. Our method has processed the clinical texts of three different structure kinds in those records. The outcome of our method is very promising with high accuracy.

In the future, determining long forms of the identified abbreviations is our next step to prepare EMRs for further data processes. Besides, we plan for a new optimized stratified sampling scheme to maintain and enhance the prediction power of the final classifier.

Acknowledgements

This work is funded by Vietnam National University at Ho Chi Minh City under the grant of the research program on Electronic Medical Records (2015-2020).

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 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 in [16, 35] very much for the source code of their algorithms in Java available on their website.

References

[1] M. Adnan, J. Warren, and M. Orr, "Iterative refinement of SemLink to enhance patient readability of discharge summaries," In Health Informatics: Digital Health Service Delivery - The Future is Now! H. Grain and L.K. Schaper (Eds.), pp. 128-134, 2013.

[2] J.J. Berman, "Pathology abbreviated: a long review of short terms. Arch Pathol Lab Med, vol. 128, pp. 347-352, 2004.

[3] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5-32, 2001.

[4] B.J. Cherry, E.W. Ford, and L.T. Peterson, "Experiences with electronic health records: early adopters in long-term care facilities," Health Care Manage Rev, vol. 36, no. 3, pp. 265-274, 2011.

[5] B. Collard and A. Royal, "The use of abbreviations in surgical note keeping," Annals of Medicine and Surgery, vol. 4, pp. 100-102, 2015.

[6] A. Dong, F. Chung, and S. Wang, "Semi-supervised classification method through oversampling and common hidden space," Information Sciences vol. 349-350, pp. 216-228, 2016.

[7] O.J. Dunn, "Multiple comparisons among means," J Amer Statist Assoc, vol. 56, no. 293, pp. 52-64, 1961.

[8] R.A. Fisher, "Statistical Methods for Research Workers," 5th. ed., Oliver and Boyd Ltd., Edinburgh, 1934.

[9] S. González, F. Herrera, and S. García, "Monotonic random forest with an ensemble pruning mechanism based on the degree of monotonicity," New Generation Computing, vol. 33, pp. 367-388, 2015.

[10] A. Henriksson, H. Moen, M. Skeppstedt, V. Daudaravičius, and M. Duneld, "Synonym extraction and abbreviation expansion with ensembles of semantic spaces," Journal of Bio-medical Semantics, vol. 5, no. 6, pp. 1-25, 2014.

[11] T. Joachims, "Transductive inference for text classification using support vector machines," In Proc. the 16th International Conference on Machine Learning, pp. 200-209, 1999.

[12] M.Y. Kim, Y. Xu, O.R. Zaiane, and R. Goebel, "Recognition of patient-related named entities in noisy telehealth texts," ACM Transactions on Intelligent Systems and Technology, vol. 6, no. 4, pp. 59:1-59:23, 2015.

[13] M. Kreuzthaler, M. Oleynik, A. Avian, and S. Schulz, "Unsupervised abbreviation detection in clinical narratives," In Proc. the Clinical Natural Language Processing Workshop, pp. 91-98, 2016.

[14] M. Kreuzthaler and S. Schulz, "Detection of sentence boundaries and abbreviations in clinical narratives," BMC Medical Informatics and Decision Making, vol. 15, pp. 1-13, 2015.

[15] H. Levene, "Robust tests for equality of variances," In Ingram Olkin; Harold Hotelling; et al. Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling. Stanford University Press, pp. 278–292, 1960.

[16] M. Li and Z.H. Zhou, "Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples," IEEE Trans Syst Man Cybern A Syst Hum, vol. 37, no. 6, pp. 1088-1098, 2007.

[17] W.J. Long, "Parsing free text nursing notes," In AMIA Annu Symp Proc., pp. 917, 2003.

[18] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," In the Workshop Proc. the

International Conference on Learning Representations, 2013.

[19] S. Moon, S. Pakhomov, N. Liu, J.O. Ryan, and G.M. Melton, "A sense inventory for clinical abbreviations and acronyms created using clinical notes and medical dictionary resources," J Am Med Inform Assoc, vol. 21, pp. 299-307, 2014.

[20] L. Shilo and G. Shilo, "Analysis of abbreviations used by residents in admission notes and discharge summaries," QJM: An International Journal of Medicine, vol. 111, no. 3, pp. 179-183, 2018.

[21] E.H. Shortliffe, "The evolution of electronic medical records," Acad Med, vol. 74, no. 4, pp. 414-419, 1999.

[22] J. Tanha, M. Someren, and H. Afsarmanesh, "Semi-supervised self-training for decision tree classifier," Int J Mach Learn & Cyber, pp. 1-16, 2015.

[23] I. Triguero, S. García S, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," Knowl Inform Syst, vol. 42, no. 2, pp. 245-284, 2015.

[24] I. Triguero, S. García, and F. Herrera, "SEG-SSC: A framework based on synthetic examples generation for self-labeled semi-supervised classification," IEEE Trans Cybern, vol. 45, no. 4, pp. 622-634, 2015.

[25] Hospital, A Set of Electronic Medical Records, Hospital, 24/02/2016.

[26] Weka 3, Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka, 2016, Accessed on 22/02/2016.

[27] Word2VecJava, https://github.com/medallia/Word2VecJava, 2016, Accessed on 22/02/2016.

[28] Y. Wu, J.C. Denny, S.T. Rosenbloom, R.A. Miller, D.A. Giuse, and H. Xu, "A comparative study of current clinical natural language processing systems on handling abbreviations in discharge summaries," In AMIA Annu Symp Proc., pp. 997-1003, 2012.

[29] Y. Wu, S.T. Rosenbloom, J.C. Denny, R.A. Miller, S. Mani, D.A. Giuse, and H. Xu, "Detecting abbreviations in discharge summaries using machine learning methods," In AMIA Annu Symp Proc., pp. 1541-1549, 2011.

[30] Y. Wu, B. Tang, M. Jiang, S. Moon, J.C. Denny, and H. Xu, "Clinical acronym/abbreviation normalization using a hybrid approach," In Proc. CLEF, pp. 1-9, 2013.

[31] Y. Wu, J.C. Denny, S.T. Rosenbloom, R.A. Miller, D.A. Giuse, L. Wang, C. Blanquicett, E. Soysal, J. Xu, and H. Xu, "A long journey to short abbreviations: developing an open-source framework for clinical abbreviation recognition and disambiguation (CARD)," J Am Med Inform Assoc, vol. 24, no. e1, pp. e79-e86, 2017.

[32] H. Xu, P.D. Stetson, and C. Friedman, "A study of abbreviations in clinical notes," In AMIA Annu Symp Proc., pp. 822-825, 2007.

[33] H. Xu, P.D. Stetson, and C. Friedman, "Methods for building sense inventories of abbreviations in clinical notes," J Am Med Inform Assoc, vol. 16, no. 1, 103-108, 2009.

[34] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," In Proc. the 33rd Annual Meeting of the Association for Computational Linguistics, pp. 189-196, 1995.

[35] Z.H. Zhou and M. Li, "Tri-Training: exploiting unlabeled data using three classifiers," IEEE Trans Knowl Data Eng, vol. 17, pp. 1529-1541, 2005.