DOI: 10.15625/vap.2017.00096

REPRESENTING CONTEXT IN ABBREVIATION EXPANSION USING MACHINE LEARNING APPROACH

Trieu Thi Ly Ly, Nguyen Van Quy, Ninh Khanh Duy, Huynh Huu Hung, Dang Duy Thang

Information Technology Faculty, University of Science and Technology, The University of Danang

lylytrieu95@gmail.com, quynguyen3490@gmail.com, nkduy@dut.udn.vn, hhhung@dut.udn.vn, ddthang@dut.udn.vn

ABSTRACT: *Text normalization is an essential problem in applications involving natural language processing since the input text often contains non-standard words such as abbreviations, numbers, and foreign words. This paper deals with the problem of normalizing abbreviations in Vietnamese text when there are several possible expansions for an abbreviation. To disambiguate the expansions for an abbreviation, a machine learning approach is proposed in which contextual information of the abbreviation is represented by either of the two models: Bag-of-words or Doc2vec. Experiments with Naïve Bayes classifier on a dataset of abbreviations collected by us shows that the average ratios of expanding correctly for Bag-of-words and Doc2vec are 86.0% and 79.7 %, respectively. Experimental results also show that information on the context plays an important role in the correct expansion of an abbreviation.*

Key words*: Text normalization, abbreviation expansion, context representation, Bag-of-words model, Doc2vec model, machine learning.*

I. INTRODUCTION

Text normalization is an essential problem in applications involving natural language processing because the input text often contains non-standard words such as numbers, dates, abbreviations, currency, and foreign words [1]. In many applications, we need to standardize these non-standard words by replacing them with contextually relevant words. However, this is not easy because non-standard words have a greater propensity than ordinary words to be ambiguous in term of their semantics or pronunciation. Therefore, it is necessary to develop intelligent algorithms to solve the problem of text normalization.

Recently there are several researches on Vietnamese text normalization, mainly in developing text-to-speech systems [2][3]. These studies have proposed some normalization approaches for all non-standard word categories in Vietnamese. However, the fact that there are too many different types of non-standard words to deal with in just one paper makes standardization methods and results for a particular word class not clearly and persuasively be presented. This issue occurs for abbreviations (hereafter refered to as ABB for short), which are popular in Vietnamese texts. In [2][3], the authors only proposed the methods for expanding abbreviations without describing the accuracy, advantages and disadvantages of these algorithms. In addition, the problem of disambiguation in expanding abbreviations has not been properly considered.

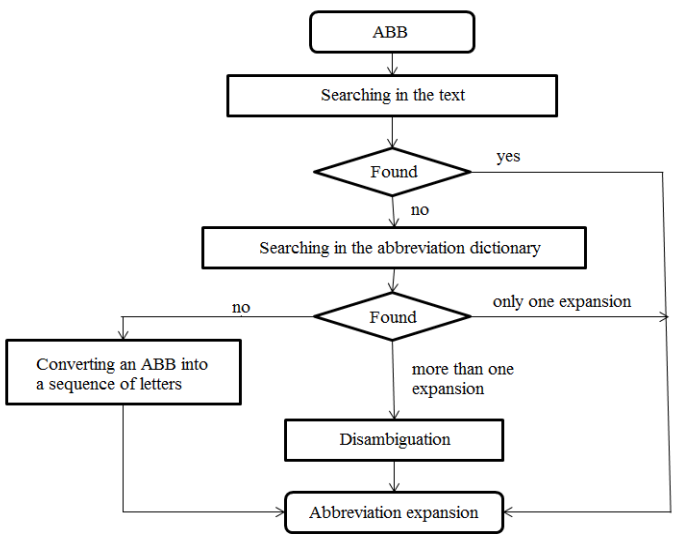

Figure 1. The algorithm for expanding abbreviations

Due to the above problems, it is necessary to have an in-depth study of normalizing abbreviations in documents. Based on the algorithm for expanding abbreviation described in [2], we propose another one illustrated in **Figure 1**. The idea of this algorithm is to search for the potential expansion in the surrounding text of the ABB first, if not any then search in an abbreviation dictionary. In case there are more than one expansion for an ABB in the dictionary, a

disambiguation algorithm will be used to find the correct expansion of the ABB. Since searching for the potential expansion in the surrounding text and the dictionary is a trivial task, we will focus on resolving the disambiguity problem when there are multiple expansions for an ABB in this paper. For example, an ABB such as "BHYT" can be expanded as "bài hát yêu thích" or "bảo hiểm y tế".

A typical approach to disambiguate abbreviation expansions involves sets of ad hoc rules based on the experience derived from a collection of ABBs. The advantage of this method is simple, however, the result obtained from a dataset are unlikely to be well generalized to another one. Therefore, we chose a machine learning approach to solve the problem of assigning an ABB to its correct expansion. This is a classification problem. By applying machine learning techniques, the classification model, which is estimated based on a sufficiently large amount of training data, is highly generalizable to any unseen dataset.

To disambiguate the expansions of an ABB in a text, contextual information of the ABB is used to decide the assigned class. In this paper, the context that we chose is the whole sentence containing an ABB which need to be expanded. Because the context of an ABB is the input to the classifier, context representation plays an important role, directly affecting the accuracy of the classifier. We have experimented with two models widely used to represent the contextual information in terms of text: Bag-of-words [4] and Doc2vec [5][6], and given the evaluations.

This paper is organized as follows. Section II describes abbreviation data collection. Section III presents two models for representing contextual information of ABBs. Experimental results with Naïve Bayes classifier is given in Section IV. In Section V, we discuss several remarks. The conclusion is given in Section VI.

II. ABBREVIATION DATA COLLECTION

A. ABB definition

Definition of ABB is quite inconsistent, depending on the author and the research [7]. Within the framework of a larger study on text normalization for text-to-speech applications [8], this paper defines a word in the text as ABB if it has two or more characters and is composed of the following elements:

- Uppercase letters: "A" to "Z", "Đ", "Ư";
- Notations: ".", "&", "-".

Some examples of ABBs are: "GS.TS" (Giáo sư Tiến sỹ), "BCHTƯ" (Ban chấp hành Trung Ương).

This article also defines the following two exceptions that are not considered as an ABB since our text normalization tool has classified these words into the "Roman Numerals" or "Currency" classes and has their own expansion methods:

- Roman numerals (For example: "IV", "XII").
- Currency (For example: "USD", "`VNĐ").

B. Some statistics

To ensure the diversity of our data sources, we collect about 100,000 articles from the 10 most popular Vietnamese electronic news website based on the rankings of alexa.com. To ensure the diversity of content, each page is divided into 20 major themes and the number of articles collected for each topic is approximately equal. Statistics the number of articles collected according to the topic that is shown in the **Figure 2**.

As a result, we have collected 1,011 ABBs with 159,050 different contexts from online newspaper sites. However, we only obtained 5 ABBs satisfying model training and testing conditions described in Section 4.2 to serve the research objectives of this paper.

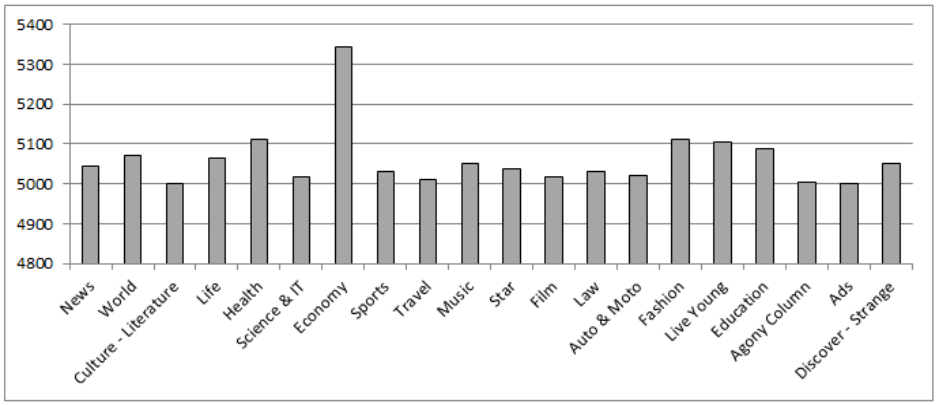

Figure 2. The number of articles collected by topics

III. TWO MODELS FOR REPRESENTING THE CONTEXT OF ABBREVIATIONS

To deal with the ambiguity when there are more than one possible expansion for an ABB, information on the context plays an important role in the correct expansion of an ABB. In this section, we present two models to represent the contextual information of ABBs: Bag-of-words and Doc2vec.

A. Bag-of-words model

The bag-of-words model (**Figure 3**) is a simplifying representation used in natural language processing and information retrieval. In this model, a text document is represented as if it were the bag of its words, disregarding grammar and word order but keeping only frequency of each word in the document. The bag-of-words model is commonly used in methods of document classification where the occurrence of each word is used as a feature for training a classifier [4].

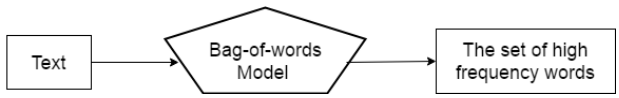

Figure 3. Bag-of-words model

When this model is applied to represent in the text, each word is expressed a binary number that depend on whether this word is belong to the sets of high frequency words or not. As a result, the input text is represented by the binary vector. The algorithm to determine of the binary feature of the text is show in the **Figure 4**.

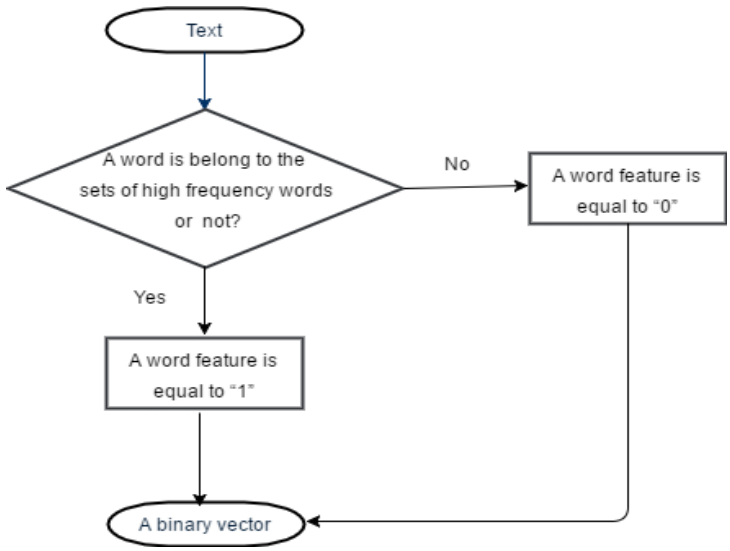

Figure 4. The algorithm to determine the binary feature of the text

Let consider an example with two sentences below: i) "Liveshow tháng 1/2016 cũng đồng thời là liveshow cuối cùng của chương trình BHYT khép lại sau 4 năm kiên trì tạo dựng một thói quen thưởng thức âm nhạc cho công chúng.", and ii) "Mặt khác, sẽ có rất nhiều trường hợp phải đăng ký khai sinh, nhập hộ khẩu và đề nghị cấp thẻ BHYT diễn ra trong ngày trong khi cán bộ, công chức phải thực hiện nhiều nhiệm vụ khác nhau.". Two sentences are the contextual information of the ABB "BHYT" which can be expanded to "bài hát yêu thích" and "bảo hiểm y tế" respectively. With the assumption that the set of high frequency words in the data include {liveshow, thẻ, khai, sinh, bệnh, nhân, âm, nhạc, ca, khúc, hộ, khẩu}, then the binary feature of the two sentences are respectively: i) [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0], and ii) [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]. It can be seen that Bag-of-words model loses the semantic information that can be deduced from the order of the words in the sentence and the feature vector expressing sentence is often sparse. It means that the feature vector has many elements equal to '0'.

B. Doc2vec Model

In 2013, researchers at Google proposed the Word2vec model [5] for learning distributed representations of a word in a vector space that can keep semantics of this word. After that, Doc2vec model [6] was developed from Word2vec model for unsupervised learning of distributed representations for larger blocks of text, such as sentences, paragraphs or entire documents. Doc2vec has outperformed the traditional methods for text representations in a research about text classification and sentiment analysis [6]. This model has been gaining the attention of the research community on natural language processing in recent years.

Word2vec used a distributed representation for each word. Word2vec takes as its input a large corpus of text and produces a vector space with several hundred dimensions. A word in the text is represented by a distribution of

weights corresponding with elements of vector. Therefore, the representation of a word is spread across all of the elements in the vector, and each element in the vector contributes to define many words. **Figure 5** illustrates a idea of Word2vec, in which the elements of the vector from the hypothesis have been labeled to be easy to understand (blue letters), although in the algorithm there was no such pre-assigned labels. It can be seen that each vector comes to represent a word(green letters) in some abstract way the 'meaning' of this word.

Figure 5. Word representation by vector in Word2vec [5]

Doc2vec modified Word2vec algorithm and add a paragraph matrix (**Figure 6**). Words are still mapped to unique vectors as before. Besides, every paragraph (or text, if working at the text level) is also mapped to an only vector. Word vectors and paragraph vectors are captured as columns in the matrix W and the matrix D respectively. The only change compared to word vector learning is that the paragraph vector is concatenated (or averaged) with the word vectors and these vector are trained to predict the next word in a context (in Figure 6, the context of three words "the", "cat", and "sat" is used to predict the fourth word "on"). Contexts are fixed length and sampling from a sliding window over a paragraph. Paragraph vectors are shared for all windows generated from the same paragraph, but not across paragraphs. On the other hand, word vectors are shared across all paragraphs.

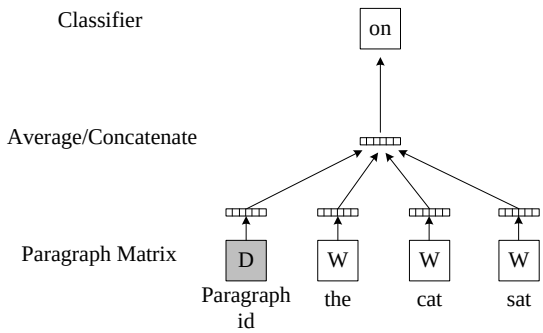

Figure 6. A framework for learning paragraph vector [6]

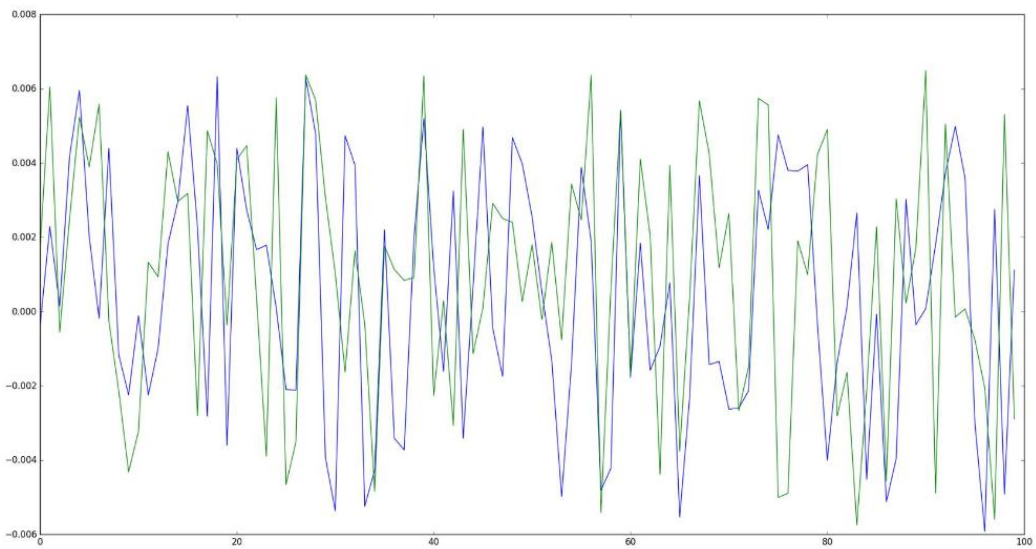

Figure 7. Graphical representation of feature vectors using Doc2vec model

In Doc2vec, each paragraph is assigned a paragraph id and is mapped to a paragraph vector via matrix D. If we apply at the sentence level, this paragraph vector can be considered as the feature vector of the sentence. With the same two sentences in the example in Section 3.1, we find two corresponding feature vectors with the components (coordinates) represented graphically, as shown in **Figure 7**. The sentence i) is a blue line and the sentence ii) is the

green line. It may be remarked that, contrary to Bag-of-words model, the feature vectors representing sentences using Doc2vec are usually quite dense, which means that there are not many components equal to "0". However, using Doc2vec model also makes the number of dimension of feature vectors which are rather large compared to Bag-of-words model. In this paper, we fix the number of dimension of the feature vectors when using Doc2vec model to 100.

IV. DISAMBIGUATION IN EXPANDING ABBREVIATIONS USING MACHINE LEARNING APPROACH

To disambiguate an ambiguous ABB, we employ a machine learning approach to select the most likely expansion in the set of possible expansions of the ABB. At this point, disambiguation problem can be seen as classification issue. The advantage of using machine learning approach is that: if a classification model is trained on a sufficiently large training set, it will be able to classify correctly with any new data (called test set) which is not included in the training set (also known as high generalizability). The disadvantage is that the training data must be large enough and have good coverage to produce a reliable classifier. Although there are many classification models, we choose the Naïve Bayes classifier for this research because of its popularity and ease of implementation. The following sections will present the machine learning approach to the Naïve Bayes classifier for disambiguation in abbreviation expansion and the experimental results with two context representation models described in Section 3.

A. Machine learning approach

Machine learning approach (particularly supervised learning) consists of two stages: training and classification. The training phase is shown in the **Figure 8**. For an ABB, each expansion of this ABB will have a corresponding classifier that needs to be estimated. In order to train the classifier for an expansion, all contexts of the ABB (i.e., the whole sentences containing the ABB) corresponding to this expansion is used as the training data.

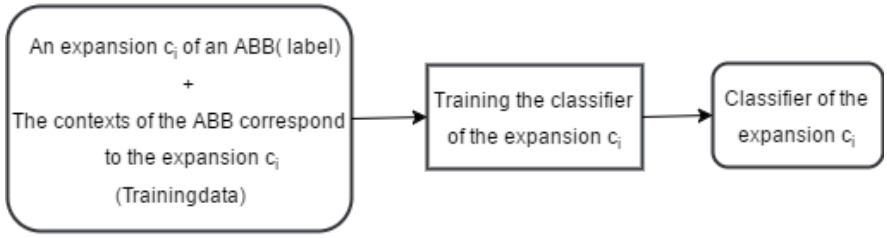

Figure 8. Training the classifier for each abbreviation expansion

Figure 9 presents the classification stage. Input data is a context of an ABB, which is called test data because it is not included in the training data. We need to find the most probable expansion for the ABB in this context. The most probable expansion is defined as the one posessing the highest score estimated in some way among the set of all possible expansions of the ABB. The score of each expansion is determined by the classifier of the expansion.

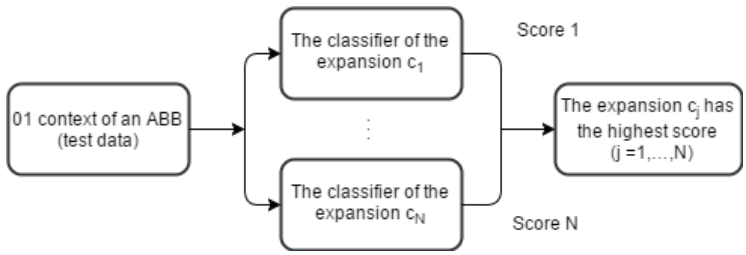

Figure 9. Classifying a context of the ABB into the most probable expansion

B. Naïve Bayes classifier

Naïve Bayes classifier is a probabilistic classifier based on Bayes' theorem. Disambiguation problem using Naïve Bayes classifier is expressed as follows: for an input document *d* consisting of the ABB and its context, the most probable expansion of the ABB is defined as the expansion $\hat{c}$ which has the maximum posterior probability given the document, which means

$$\hat{c} = \text{argmax}_{c \in C} P(c|d), \quad (1)$$

where c is an expansion in the set C of all possible expansions of the ABB. Therefore, the estimated score of the expansion c is the probability $P(c|d)$ provided by the Naïve Bayes classifier. In this paper, training and classification experiments with Naïve Bayes classifier is carried out by using the scikit-learn toolkit [9].

C. Data preparation

Before starting the experiment, we filtered out the ABBs whose data do not satisfy the conditions for training and testing the classification model as follows:

- The number of training data is less than or equal to 5 samples. This is because if the amount of training data is too small, it is impossible to train the classification model reliably using machine learning algorithm.
- Training data is too biased towards a certain expansion of an ABB, specifically an expansion has more than 20 times as many training samples as another expanson. This is to ensure that test results accurately reflect the disambiguation capacity of classifiers.

After filtering process, we obtained 5 ABBs satisfying the above two conditions, which are "BHYT", "NS", "PTTH", "THA", and "KH". This number is significantly less than 1,011 ABBs collected in Section 2.2. This is because, for most of ABBs, the collected contexts is either too few or unevenly distributed among expansions. **Table 1** shows the amount of data used to train the classifier for each ABB expansion. Note that the amount of data used for evaluation in the next section is also equal to the that used for training the classifiers.

Table 1. Statistics on training data for expansions (in descending order of the total number of training data)

No	Abbreviation	Expansion	Number of training data	Total number of training data
1	BHYT	bài hát yêu thích	52	295
		bảo hiểm y tế	243	
2	NS	nghệ sĩ	44	99
		nhạc sĩ	55	
3	PTTH	phát thanh truyền hình	26	49
		phổ thông trung học	23	
4	THA	thi hành án	17	29
		tăng huyết áp	12	
5	KH	khoa học	7	17
		kế hoạch	10	

D. Experimental results

We conducted training and testing experiments with Naïve Bayes classifiers using two context representation models: Bag-of-words and Doc2vec. The results in **Table 2** shows that Bag-of-words model gives a higher (or equal) rate of correct expansion than Doc2vec model in all cases. The average accuracy of Bag-of-words is 86,0%, while that of Doc2vec is 79,7%. On average, the accuracy of abbreviation expansion experiments using Naïve Bayes classifier is 82,9%, in which accuracy rates vary greatly depending on the ABB and its expansions as discussed below.

V. DISCUSSION

In machine learning approach based on a probabilistic model such as the Naïve Bayes classifier, it is often observed that the bigger the amount of training data we use, the higher accuracy the classification model achieves. In **Table 2**, it can be seen that, with the disambiguation problem in expanding ABBs using the statistical machine approach, the degree of closeness (or difference) of socio-economic fields to which the expansions belong also plays the role as important as the size of training data. If the fields are not much associated to each other, the disambiguation method gives a fairly high radio of expanding ABBs correctly (above 90%) whether more or less training data is used because the context of an ABB demonstrates its classification capability. The examples illustrate for this reasoning including the ABB "BHYT", which can be expanded as either "bài hát yêu thích" (music field) or "bảo hiểm y tế" (health-care field), or the ABB "THA", which can be expanded as either "thi hành án" (law field) or "tăng huyết áp" (health-care field). In contrast, if the fields of ABB expansions are interrelated or closed to each other, the context of the ABB no longer exhibits its power in the classification process. As a result, the ratio of accurate expansions is relatively low (only higher than 70%) whether more or less training data is used. This remark applies for the expansions "nghệ sĩ" and "nhạc sĩ" (in the case the ABB "NS"), or "khoa học" and "kế hoạch" (for the ABB "KH").

Table 2. Expansion accuracy of two context representation models: Bag-of-words and Doc2vec
(the number of data used for testing is equal to that used for training the classifiers in Table 1)

No	ABB	Expansion	Bag-of-words	Doc2vec	Average accuracy
1	BHYT	bài hát yêu thích	98,0%	98,0%	**98,0%**
		bảo hiểm y tế			
2	NS	nghệ sĩ	77,5%	74,5%	76,0%
		nhạc sĩ			
3	PTTH	phát thanh truyền hình	83,7%	69,4%	76,5%
		phổ thông trung học			
4	THA	thi hành án	93,3%	90,0%	**91,7%**
		tăng huyết áp			
5	KH	khoa học	77,8%	66,7%	72,2%
		kế hoạch			
Average			**86,0%**	**79,7%**	**82,9%**

VI. CONCLUSION

In this paper, we have proposed two methods of context representation for an abbreviation in order to disambiguating its expansions, a traditional method Bag-of-words and a recently proposed method Doc2vec. A statistical machine learning approach using Naïve Bayes classifier was also described and experimentally implemented to demonstrate the effectiveness of these two context representation methods. On average, Bag-of-words model gives a higher classification accuracy than Doc2vec model around 6%. The average accuracy of abbreviation expansion experiments using Naïve Bayes classifier is approximately 83%. In the future, we will expand the abbreviation dataset as well as perform experiments using other classification models to obtain a more comprehensive assessment of Bag-of-words and Doc2vec's performances in the context representation of abbreviations.

ACKNOWLEDGMENTS

This research is supported by the Ministry of Education and Training under project No. B2016-DNA-38-TT, and the University of Science and Technology - The University of Danang under project No. T2017-02-93.

REFERENCES

[1] Richard Sproat, Alan Black, Stanley Chen, Shankar Kumar, Mari Ostendorf, and Christopher Richards, "Normalization of Non-Standard Words", Computer Speech and Language, vol. 15, no. 3, pp. 287-333, 2001.

[2] Thu-Trang Thi Nguyen, Thanh Thi Pham, Do-Dat Tran, "A Method for Vietnamese Text Normalization to Improve the Quality of Speech Synthesis", Proceedings of International Symposium on Information and Communication Technology (SoICT), Vietnam, 2010.

[3] Dinh Anh Tuan, Phi Tung Lam, Phan Dang Hung, "A Study of Text Normalization in Vietnamese for Text-to-Speech System", Proceedings of Oriental COCOSDA Conference, China, 2012.

[4] Daniel Jurafsky, James H. Martin, Speech and Language Processing, 2nd edition, Prentice Hall, 2008.

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean, "Distributed Representations of Words and Phrases and their Compositionality", Proceedings of Conference on Neural Information Processing Systems (NIPS), USA, 2013.

[6] Quoc Le, Tomas Mikolov, "Distributed Representations of Sentences and Documents", Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014.

[7] Nguyen Nho Tuy, Phan Huy Khanh, "Developing Database of Vietnamese Abbreviations and Some Applications", Proceedings of Second International Conference on Nature of Computation and Communication, Rach Gia, Vietnam, 2016.

[8] Paul Taylor, Text-to-Speech Synthesis, Cambridge University Press, 2009.

[9] Fabian Pedregosa et al., "Scikit-learn: Machine Learning in Python", Journal of Machine Learning Research, vol. 12(Oct), pp. 2825-2830, 2011.