Efficient Autocoding Method in High Dimensional Space

Yukako Toko (ytoko@nstac.go.jp)
National Statistics Center, Japan

Mika Sato-Ilic (msato@nstac.go.jp, mika@risk.tsukuba.ac.jp)
National Statistics Center, Japan, / University of Tsukuba, Japan

ABSTRACT

In recent years, data handled in official statistics is getting large and complex. This paper proposes a new autocoding method utilizing a metric in high dimensional space to efficiently classify large and complex data. The proposed method is a hybrid method of Support Vector Machine (SVM) utilized Word2Vec and previously developed autocoding method based on reliability scores. Word2Vec was developed based on an idea of a neural probabilistic language model in which words are embedded in a continuous space using distributed representations of the words. SVM is a supervised machine learning algorithm for classification utilizing a metric in high dimensional space. It is known as high discrimination ability and generalization performance. In this paper, Word2Vec is used for notation from a word to a numerical vector, and SVM is used for classification based on the numerical vectors. In order to improve both ability of high classification accuracy and generalization performance, we combine classification by SVM that is known as classifying numerical vectors with high generalization performance and autocoding method based on the reliability score. Numerical examples show the efficiency of the proposed method. That is, the numerical examples show a better performance of the proposed hybrid method, which combines SVM and an autocoding method based on reliability scores, compared with the results of classification accuracy of cases when we apply either one of the methods. The proposed method is developed in R utilizing existing R packages for efficient development.

Keywords: *Coding, Machine learning, Word2Vec, Support Vector Machine, Reliability score*

JEL Classification: *C38*

1. INTRODUCTION

In official statistics, text response fields such as fields in *the family Income and Expenditure Survey* or occupation, industry, are found in survey forms. Those respondents' text descriptions are usually translated into corresponding codes for efficient data processing. Originally, coding tasks are performed manually, whereas the studies of automated coding have made progress with the improvement of computer technology in recent years. For example, Hacking and Willenborg (2012) introduced coding methods,

including autocoding. Gweon et al. (2017) illustrated methods for automated occupation coding based on statistical learning. However, as data is getting complex and large in recent years, there is a need for efficiently handling those data with higher accuracy of classification and generalization for obtaining robust classification results for various kinds of inputted text descriptions.

For this purpose, this paper proposes a new autocoding method for large amounts of complex data utilizing Support Vector Machine (SVM) (Cristianini and Shawe-Taylor, 2000), Word2Vec (Mikolov et al., 2013), and our previously developed autocoding method based on reliability score (Toko and Sato-Ilic, 2020).

It is known that a Bernoulli type simple Bayesian model-based autocoding method of Naïve Bayes (Toko et al., 2017) does not always perform well as it is known to have less classification accuracy for large amounts of complex data due to a lack of consideration of the relationship among words. Moreover, in this case the number of dimensions tends to be large; therefore, the curse of dimensionality problem tends to occur. Therefore, we have developed several autocoding methods based on reliability scores (Toko et al., 2018a, Toko et al., 2018b) considering humans' uncertainty of recognizing the autocoding of words for adjusting the complexity of data. In addition, considering the generalization for adapting to the high variability of obtained data, we have extended the simple reliability scores to generalized reliability scores considering robustness and generalization for adjusting various types of complex data (Toko et al., 2019, Toko and Sato-Ilic, 2020). However, these methods are not enough to obtain better classification accuracy for a large amount of data.

In order to solve this problem, this paper proposes a new autocoding method, which is a hybrid method of SVM utilizing Word2Vec and a previously developed autocoding method based on reliability score (Toko and Sato-Ilic, 2020) for large amounts of complex data to improve both the ability of high classification accuracy and generalization performance. We apply SVM for classification based on the numerical vectors with high generalization performance. Word2Vec, a well-known method to produce word embeddings, is utilized to obtain numerical vectors corresponding to words. In addition, the reliability score is applied to improve accuracy. The proposed autocoding system has been developed in R for efficient development. We utilize "wordVectors" package (Schmidt and Li, 2020) to train Word2Vec models and "e1071" package (Meyer et al., 2019) to train a support vector machine. Numerical examples show the efficiency of the proposed hybrid method.

The rest of this paper is organized as follows: Word2Vec and SVM are explained in sections 2 and 3. The method of autocoding based reliability

score is described in section 4. The hybrid method of autocoding in high dimensional space is proposed in section 5. The numerical examples are illustrated in section 6. Conclusions are described in section 7.

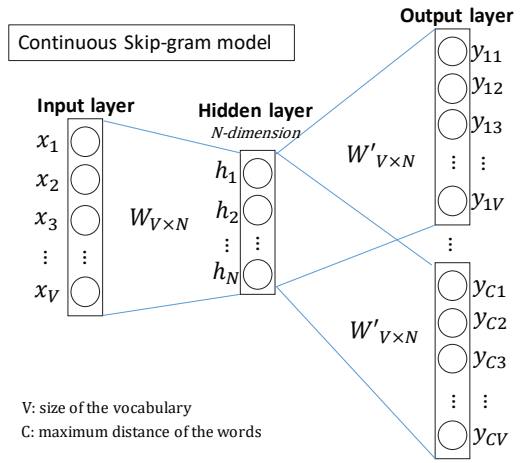
2. EFFICIENT ESTIMATION OF WORD REPRESENTATIONS IN VECTOR SPACE (WORD2VEC)

Word2Vec was developed based on an idea of a neural probabilistic language model in which words are embedded to a continuous space by using distributed representations of the words (Mikolov et al., 2013). The algorithm of Word2Vec learns word association from a given dataset utilizing a neural network model based on an idea of a neural probabilistic language model (Bengio et al., 2003). It produces a vector space and each word in the given dataset is assigned a corresponding numerical vector of a word in the produced vector space. The essence of the idea is to avoid the curse of dimensionality by distributed representations of words.

Word2Vec utilizes continuous bag-of-words (CBOW) model and continuous skip-gram model to distributed representation on words. The CBOW model predicts the current word based on the context. The skip-gram model uses each current word to predict words within a certain range before and after the current word. It gives less weight to the distant context words. In this study, we applied the skip-gram model. Fig. 1 visually shows the skip-gram model architecture of Word2Vec. It is a two-layer neural network, and it takes each word in a given dataset as an input to produce an N-dimensions vector space.

Continuous skip-gram model of Word2Vec

Fig. 1

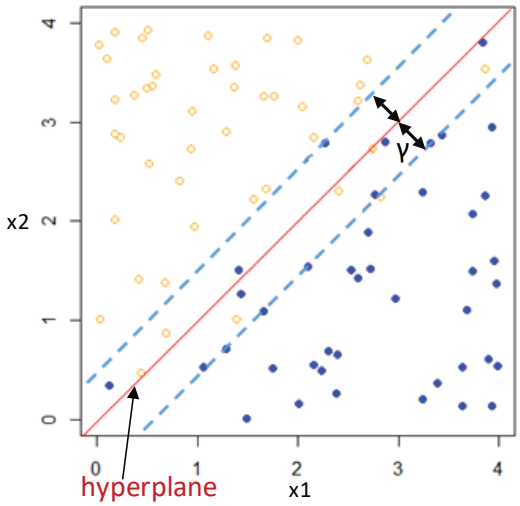


3. SUPPORT VECTOR MACHINE

Support Vector Machine (SVM) (Cristianini and Shawe-Taylor, 2000) is a supervised machine learning algorithm for classification and regression. Fig. 2 shows the basic idea of SVM. It finds the maximum-margin hyperplane in high dimensional space for classification.

Classification by SVM

Fig.2



If $\boldsymbol{w}$ is the weight vector realizing a functional margin of 1 on the positive point $\boldsymbol{x}^+$ and negative point $\boldsymbol{x}^-$, a functional margin of 1 implies

$$\langle \boldsymbol{w} \cdot \boldsymbol{x}^+ \rangle + b = +1,$$

$$\langle \boldsymbol{w} \cdot \boldsymbol{x}^- \rangle + b = -1,$$

while $\boldsymbol{w}$ is normalized. Then the margin γ is the functional margin of the resulting classifier

$$\begin{aligned} \gamma &= \frac{1}{2}\left(\langle \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2} \cdot \boldsymbol{x}^+ \rangle - \langle \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2} \cdot \boldsymbol{x}^- \rangle \right) \\ &= \frac{1}{2\|\boldsymbol{w}\|_2} \left(\langle \boldsymbol{w} \cdot \boldsymbol{x}^+ \rangle - \langle \boldsymbol{w} \cdot \boldsymbol{x}^- \rangle \right) \\ &= \frac{1}{\|\boldsymbol{w}\|_2}. \end{aligned}$$

Therefore, the resulting margin will be equal to $1/\|\boldsymbol{w}\|_2$ and the following can be written.

Given a linearly separable training sample

$$S = \left((\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_l, y_l) \right)$$

the hyperplane $(\boldsymbol{w}, b)$ that solves the optimization problem

$$\begin{aligned} &\min_{\boldsymbol{w},b} \langle \boldsymbol{w} \cdot \boldsymbol{w} \rangle, \\ &\text{subject to } y_i (\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b) \geq 1, i = 1, \dots, l, \end{aligned} \tag{1}$$

realizes the maximal margin hyperplane with geometric margin $\mathrm{M} = 1/\|\boldsymbol{w}\|_2$. Then, slack variables are introduced to allow the margin constraints to be violated

$$\begin{aligned} \text{subject to } y_i (\langle \boldsymbol{w} \cdot \boldsymbol{x}_i \rangle + b) &\geq 1 - \xi_i, i = 1, \dots, l, \\ \xi_i &\geq 0, i = 1, \dots, l. \end{aligned}$$

From the above, the optimization problem shown in (1) can be written as

$$\min_{\xi,w,b}\langle \boldsymbol{w}\cdot\boldsymbol{w}\rangle + C\sum_{i=1}^{l}\xi_i, \tag{2}$$

$$\text{subject to } y_i(\langle \boldsymbol{w}\cdot\boldsymbol{x}_i\rangle + b) \geq 1-\xi_i, \qquad i=1,\ldots,l,$$

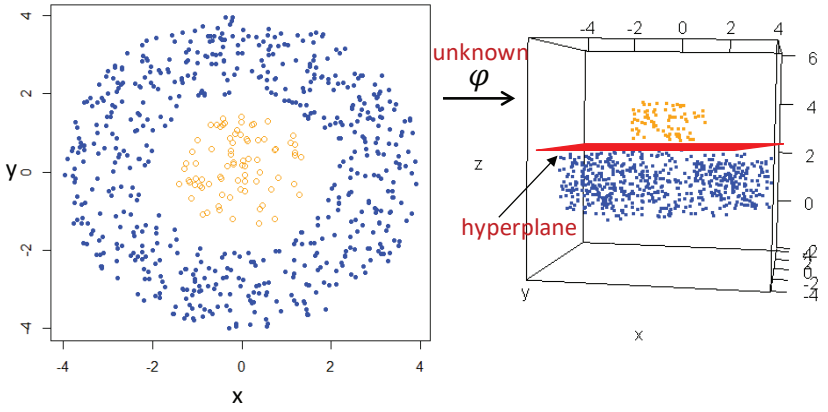
$$\xi_i \geq 0, i=1,\ldots,l.$$

where C is the cost parameter that will give the optimal bound as it corresponds to finding the minimum of $\|\xi\|_1$ in (2) with the given value for $\|\boldsymbol{w}\|_2$. This is soft-margin linear SVM.

Also, SVM performs a non-linear classification transforming input data into higher dimensional spaces and calculating the inner product between the data in higher dimensional space using kernel trick. Fig. 3 shows the transformation of input data into higher dimensional space. The left side picture in Fig. 3 shows that the discrimination of two-colored data is impossible in the two-dimensional space. However, in the right-side picture, the transformed space shows it will be possible to discriminate using the metric of product in the high dimensional space by using the kernel trick.

Transformation of input data into higher dimensional space

Fig. 3



SVM uses kernel functions to enable it to obtain the inner product of data in higher dimen sional space (kernel trick), which is represented as follows:

$$k(\boldsymbol{x}_i,\boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^T\varphi(\boldsymbol{x}_j),$$

where φ is a mapping function from an observational space to a higher-dimensional space. The conditions for $k(\boldsymbol{x}, \boldsymbol{x}')$ to be a kernel function are as follows:

· Symmetry: $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}', \boldsymbol{x})$.
· Gram matrix is Positive semi-definite.

There are many possible choices for the kernel function, such as

· Radial basis function kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{x}'\|^2), \qquad \left(\gamma = \frac{1}{2\sigma^2}\right), \tag{3}$$

· Polynomial kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = (c + \boldsymbol{x}^T \boldsymbol{x}')^d, \quad (c > 0, d \in \boldsymbol{N}),$$

· Sigmoid kernel

$$k(\boldsymbol{x}, \boldsymbol{x}') = \tanh(a\boldsymbol{x}^T \boldsymbol{x}' + c).$$

In this paper, the radial basis function is applied. The mapped feature space of this kernel function has an infinite number of dimensions.

For multiclass SVM, there are two approaches: one-versus-the-rest and one-versus-one. In one-versus-the-rest, SVM builds binary classifiers that discriminate between one class and the rest, whereas it builds binary classifiers that discriminate between every pair of classes in one-versus-one.

4. AUTOCODING BASED RELIABILITY SCORE

An autocoding method of a Bernoulli type simple Bayesian model-based autocoding method of Naïve Bayes (Toko et al., 2017) comprises the training and classification processes. In the training process, the extraction of objects and the creation of an object frequency table are performed. First, each text description in the training dataset is tokenized MeCab (Kudo et al., 2004), a dictionary-attached morphological Japanese text analyzer. Then, word-level N-grams from the word sequences of a text description are taken as objects. After the object extraction, the classifier tabulates all extracted objects based on their given codes into an object frequency table.

In the classification process, first, the classifier performs the extraction of objects and retrieval of candidate codes from the object frequency table

provided by using the extracted objects. Then, it calculates the relative frequency of j -th object to a code k defined as

$$p_{jk} = \frac{n_{jk}}{n_j}, \quad n_j = \sum_{k=1}^{K} n_{jk}, \quad j = 1, \dots, J, \qquad k = 1, \dots, K,$$

where n_{jk} is the number of occurrence of statuses in which an object *j* assigned to a code k in the training dataset. J is the number of objects and K is the number of codes.

However, this classifier has difficulty correctly assigning codes to text descriptions for complex data included uncertainty. To address this problem, we developed the overlapping classifier that assigns codes to each text description based on the reliability score (Toko and Sato-Ilic, 2020). Then, the classifier arranges $\{p_{j1}, \cdots, p_{jK}\}$ in descending order and creates $\{\tilde{p}_{j1}, \cdots, \tilde{p}_{jK}\}$, such as $\tilde{p}_{j1} \geq \cdots \geq \tilde{p}_{jK}, j = 1, \cdots, J$. After that, $\{\tilde{\tilde{p}}_{j1}, \cdots, \tilde{\tilde{p}}_{j\tilde{K}_j}\}$, $\tilde{K}_j \leq K$ are created. That is, each object has a different number of classes (or codes). Then, the classifier calculates the reliability score for each class (or code) of each object. The reliability score of j-th object to a code k is defined as

$$\bar{p}_{jk} = T\left(\tilde{\tilde{p}}_{jk}, 1 + \sum_{m=1}^{\tilde{K}_j} \tilde{\tilde{p}}_{jm} \log_K \tilde{\tilde{p}}_{jm}\right), \qquad j = 1, \dots, J, k = 1, \dots, \tilde{K}_j. \qquad (4)$$

$$\bar{p}_{jk} = T\left(\tilde{\tilde{p}}_{jk}, \sum_{m=1}^{\tilde{K}_j} \tilde{\tilde{p}}_{jm}^2\right), \qquad j = 1, \dots, J, k = 1, \dots, \tilde{K}_j.$$

These reliability scores were defined considering both probability measure and fuzzy measure (Bezdek, 1981, Bezdek et al., 1999). That is, $\tilde{\tilde{p}}_{jk}$ shows the uncertainty from the training dataset (probability measure) and $1 + \sum_{m=1}^{\tilde{K}_j} \tilde{\tilde{p}}_{jm} \log_K \tilde{\tilde{p}}_{jm}$ or $\sum_{m=1}^{\tilde{K}_j} \tilde{\tilde{p}}_{jm}^2$ shows the uncertainty from the latent classification structure in data (fuzzy measure). These values of the uncertainty from the latent classification structure can show the classification status of each object; that is, how each object is classified to the candidate classes (or codes). *T* shows *T*-norm in statistical metric space (Menger, 1942, Mizumoto, 1989, Schweizer and Sklar, 2005). We generalize the reliability score by using the idea of *T*-norm which is a binary operator in statistical metric space. *T*-norm satisfies the following four conditions:

- Boundary conditions

$$0 \le T(a,b) \le 1, \qquad T(a,0) = T(0,b) = 0, \qquad T(a,1) = T(1,a) = 1$$

- Monotonicity

$$a \le c\,, b \le d \;\rightarrow T(a,b) \le T(c,d)$$

- Symmetry

$$T(a,b) = T(b,a)$$

- Associativity

$$T(T(a,b),c) = T(a,T(b,c))$$

where $\forall a,b,c,d \in [0,1]$. Typical examples of *T*-norms are as follows:

- Algebraic product

$$T(x,y) = xy$$

- Sin-based *T*-norm

$$T(x,y) = \frac{2}{\pi}\sin^{-1}\left[\left(\sin\frac{\pi}{2}x + \sin\frac{\pi}{2}y - 1\right) \vee 0\right]$$

- Hamacher product

$$T(x,y) = \frac{xy}{p + (1-p)(x + y - xy)}, \quad p \ge 0$$

- Dombi product

$$T(x,y) = \frac{1}{1 + \sqrt[p]{\left(\frac{1-x}{x}\right)^p + \left(\frac{1-y}{y}\right)^p}}, \qquad p > 0$$

In this paper, we use a case of algebraic product.

5. HYBRID METHOD OF AUTOCODING IN HIGH DIMENSIONAL SPACE

Although we defined the reliability score to improve the classification accuracy and generalization performance for text descriptions by utilizing ideas of fuzzy measure and *T*-norms, it is not satisfactory enough for adjusting to a large amount of data.

The proposed method is a hybrid method of SVM utilizing Word2Vec and a previously developed autocoding method based on reliability score for a large amount of data to improve both ability of high classification accuracy and generalization performance. SVM is applied for classification based on the numerical vectors with high generalization performance. To perform classification by SVM, we apply Word2Vec to obtain numerical vectors corresponding to words, as it is a well-known technique to produce word embeddings. In addition, classification based on the reliability score is performed for improving accuracy.

First, the proposed method obtains numerical vectors corresponding words utilizing Word2Vec after tokenizing each text description by MeCab. Then, it produces sentence vectors for each text description based on the obtained numerical vectors and assigns corresponding codes by using a classifier obtained by SVM. After that, the proposed algorithm performs re-classification based on the previously defined reliability score to unmatched text descriptions at classifying by SVM.

The detailed algorithm of the proposed method is the following:

Step 1. The proposed algorithm tokenizes each text description into words by MeCab.

Step 2. It obtains numerical vectors corresponding to words utilizing Word2Vec: First, it produces a dataset concatenating all tokenized words consecutively. Then, it trains Word2Vec model using the produced dataset. Each unique word in the dataset will be assigned a corresponding numerical vector. We determine the followings by trial and error:

- Type of model architecture: CBOW model or skip-gram model
- The number of vector dimensions
- The number of training iterations
- Window size of words considered by the algorithm

Step 3. It produces sentence vectors for each text description based on the obtained numerical vectors at Step 2: First, it obtains a corresponding numerical vector for each word in each text description from the trained Word2Vec model. Then, it

calculates the sum of obtained numerical vectors for each text description as sentence vectors.

Step4. It assigns corresponding codes applying SVM: It trains a support vector machine and then predicts a corresponding code for each target text description. For training a support vector machine, we determine the followings:

- Cost parameter appeared in (2) as C
- Kernel function to be applied
- Gamma parameter appeared in (3) as γ if radial basis function kernel is applied as a kernel function
- Type of methods: one-versus-the-rest or one-versus-one

In this study, a radial basis function as a kernel function is applied. We apply the one-versus-one method as we use the "e1071" package that trains a support vector machin using a one-versus-one approach. Cost parameter C and gamma parameter γ are determined by grid search.

Step5. It extracts unmatched text descriptions in Step 4.

Step6. It implements re-classification based on the reliability score in (4) described in section 4.

6. NUMERICAL EXAMPLE

For the numerical example, the proposed hybrid method of autocoding is applied to the Stack overflow dataset (Xu et al., 2015). The Stack overflow dataset publicly available short text description dataset published in Kaggle. This dataset contains 20,000 instances, consisting of question titles in English and 20 different codes. We used randomly extracted 2,000 instances of the stack overflow dataset for evaluation and used the rest of the dataset for training.

The following data pre-processing is performed utilizing existing R packages before the implementation of classification by the proposed method.

- Replace punctuations with space using "gsub" function in the "base" package.
- Convert uppercase letters to low ercase using "tolower" function in the "base" package.
- Remove unnecessary space using "stripWhitespace" function in the "tm" package (Feinerer and Hornik, 2019) and "str_trim" function in the "stringr" package (Wickham, 2019).

We used "WordVector" package for training the word2vec model. We selected the skip-gram model as a type of model architecture and set the number of vector dimensions as 100, the number of training iterations as 10,

and the window size as 2. We used the "e1071" package for training a support vector machine. We set the cost parameter C appeared in (2) as 10, and the gamma parameter γ appeared in (3) as 0.001, and use radial basis function as the kernel function.

Table 1 compares the classification accuracy of the proposed hybrid autocoding method, a Bernoulli type simple Bayesian model based autocoding method (Toko et al., 2017), and the autocoding method based on reliability scores (Toko and Sato-Ilic, 2020), and SVM. From this table, it can be seen that the classification accuracy of the proposed method described in Section 5 is 0.910. A Bernoulli type simple Bayesian model-based autocoding method (Toko et al., 2017) is 0.652. An autocoding method based on reliability scores (Toko and Sato-Ilic, 2020) shown in (4) is 0.870. In addition, the classification accuracy by simply applying SVM is 0.824.

Therefore, it is found that the Bernoulli type simple Bayesian model-based autocoding method has the worst accuracy. So, we can see that this method cannot obtain a better classification accuracy for complex data, and our proposed methods based on reliability scores considering the uncertainty of human recognition and robustness of the solution performs better than the Bernoulli type simple Bayesian model.

And it can be seen that the proposed hybrid method in this paper has the best classification accuracy. This means that the proposed hybrid method that combines SVM and the autocoding method based on reliability scores successfully obtains better performance than when SVM is only applied or when the autocoding method based on reliability scores is only used. Therefore, we can show the efficiency of using the hybrid method of these two methods for classification.

Comparison of classification accuracy of the proposed hybrid autocoding method and conventional methods

Table 1

	Number of text descriptions			accurucy
	Training	Evaluation	Correctly assigned	
Classification by the proposed method	18,000	2,000	1,820	0.910
Classification by SVM			1,648	0.824
Classification based on the relative frequency			1,304	0.652
Classification based on the reliability score			1,740	0.870

7. CONCLUSION

This paper proposes a new autocoding method, a hybrid method of SVM utilizing Word2Vec, and a previously developed autocoding method based on reliability scores for complex and large amounts of data to improve both the ability of high classification accuracy and generalization performance. SVM is applied for classification based on the numerical vectors obtained by Word2Vec with high generalization performance, and autocoding method based on the reliability score is utilized for improving accuracy. The numerical examples show that the proposed hybrid method gives a better classification result compared with the ordinary Bernoulli type simple Bayesian model-based autocoding method. The proposed hybrid method that combines SVM and the previously developed autocoding method based on reliability scores obtains a better result compared with either one of the used cases. Therefore, we show the efficiency of the proposed hybrid method. The proposed method is developed in R utilizing existing R packages such as "wordVectors" and "e1071" for efficient development. This paper presents a numerical example based on open data; however, in the same logic, the proposed method can be applied to any data, including a survey for occupation and industry. However, of course, it is important to use real survey data, so it will be future work for this study. In addition, theoretically, any languages are adaptable for the proposed method; however, it might exist some pre-processing or ad-hoc language-specific adaptation depended on the kinds of languages. This will be a future problem to investigate. Moreover, the numerical example showed a better performance for the improved accuracy. However, in practical use, we need to improve accuracy.

Acknowledgements: We would like to thank Kaggle for making the Stack Overflow dataset available.

References

1. **Bengio, Y., Ducharme, R., Vincent, P., Jauvin, C**., 2003, *"A neural probabilistic language model"*, Journal of Machine Learning Research, 3, pp. 1137-1155.
2. **Bezdek, J.C.**, 1981, Pattern recognition with fuzzy objective function algorithms, Plenum Press.
3. **Bezdek, J.C., Keller J., Krisnapuram, R., Pal, N.R.**, 1999, *Fuzzy models and algorithms for pattern recognition and image processing*, Kluwer Academic Publishers.
4. **Cristianini, N., Shawe-Taylor, J.,** 2000, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge University Press.
5. **Feinerer, I., Hornik, K.**, 2019, *tm: Text Mining Package. R package version 0.7-7*, https://CRAN.R-project.org/package=tm.
6. **Gweon, H., Schonlau, M., Kaczmirek, L., Blohm, M., Steiner, S.**, 2017, *"Three*

methods for occupation coding based on statistical learning", Journal of Official Statistics, Vol. 33, No. 1, pp. 101-122.
7. **Hacking, W., Willenborg, L.**, 2012, *"Coding; interpreting short descriptions using a classification"*, Statistics Methods, Statistics Netherlands, The Hague, Netherlands, Available at: https://www.cbs.nl/en-gb/our-services/methods/statistical-methods/throughput/throughput/coding (accessed December 2020).
8. *Kudo, T., Yamamoto, K., Matsumoto, Y.*, 2004, *"Applying conditional random fields to Japanese morphological analysis"*, in the 2004 Conference on Empirical Methods in Natural Language Processing, pp. 230-237.
9. **Menger, K.**, 1942, *"Statistical metrics"*, in Proceedings of the National Academy of Sciences of the United States of America, Vol. 28, pp. 535-537.
10. **Meyer, D., Dimitriadou, E., Hornik, K., Weingessel, A., Leisch, F**., 2019, *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071)*, TU Wien, R package version 1.7-3, https://CRAN.R-project.org/package=e1071.
11. **Mikolov, T., Chen, K., Corrado, G., Dean, J**., 2013, *"Efficient estimation of word representations in vector space"*, arXiv preprint arXiv:1301.3781.
12. *Mizumoto, M*., 1989, *"Pictorical representation of fuzzy connectives, Part I: Cases of T-norms, t-Conorms and Averaging Operators"*, Fuzzy Sets and Systems, Vol. 31, pp. 217-242.
13. **Schmidt, B., Li, J.**, (2020), *wordVectors: Tools for creating and analyzing vector-space models of texts*, R package version 2.0. http://github.com/bmschmidt/wordVectors (accessed November 2020).
14. **Schweizer, S., Sklar, A**., 2005, *Probabilistic metric spaces*, Dover Publications.
15. **Toko, Y., Wada, K., Kawano, M.**, 2017, *"A supervised multiclass classifier for an autocoding system"*, Journal of Romanian Statistical Review, Vol. 4, pp. 29-39.
16. **Toko, Y., Wada, K., Iijima, S., Sato-Ilic, M.**, 2018a, *"Supervised multiclass classifier for autocoding based on partition coefficient"*, Czarnowski, I., Howlett, R.J., Jain, L. C., and Vlacic, L. (Eds.), Intelligent Decision Technologies 2018, Smart Innovation, Systems and Technologies, Springer, Vol. 97, pp. 54-64.
17. **Toko, Y., Iijima, S., Sato-Ilic, M**., 2018b, *"Overlapping classification for autocoding system"*, Journal of Romanian Statistical Review, Vol. 4, pp. 58-73.
18. *Toko, Y., Iijima, S., Sato-Ilic, M*., 2019, *"Generalization for improvement of the reliabil ity score for autocoding"*, Journal of Romanian Statistical Review, Vol. 3, pp. 47–59.
19. **Toko, Y., Sato-Ilic, M.**, 2020, *"Improvement of the training dataset for supervised multiclass classification"*, Czarnowski, I., Howlett, R.J., Jain, L. C. (Eds.), Intelligent Decision Technologies, Smart Innovation, Systems and Technologies, Springer, Singapore, Vol. 193, pp. 291-302.
20. **Wickham, H.**, 2019, *stringr: Simple, Consistent Wrappers for Common String Operations*, R package version 1.4.0, https://CRAN.R-project.org/package=stringr.
21 . **Xu, J., Wang, P., Tian, G., Xu, B., Zhao, J., Wang, F., Hao, H.**, 2015, *"Short text clustering via convolutional neural networks"*, In: NAACL-HLT on Proceedings, pp. 62–69.