# DEEP LEARNING BASED DATA GOVERNANCE FOR CHINESE ELECTRONIC HEALTH RECORD ANALYSIS

Junmei Zhong[1], Xiu Yi[2], Jian Wang[2], Zhuquan Shao[2], Panpan Wang[2], and Sen Lin[2]

[1]Inspur USA Inc
2010 156th Ave NE Bellevue, WA 98052
[2]Inspur Software Group, Technology Center
1036 Langchao Rd., Jinan, China

## ABSTRACT

*Electronic health record (EHR) analysis can leverage great insights for improving the quality of human health care. However, the low data quality problems of missing values, inconsistency, and errors in the data columns hinder building robust machine learning models for data analysis. In this paper, we develop a methodology of artificial intelligence (AI)-based data governance to predict the missing values or verify if the existing values are correct and what they should be when they are wrong. We demonstrate the performance of this methodology through a case study of patient gender prediction and verification. Experimental results show that the deep learning algorithm works very well according to the testing performance measured by the quantitative metric of F1-Score, and it outperforms support vector machine (SVM) models with different vector representations for documents.*

## KEYWORDS

*EHR Analysis, Data Governance, Vector Space Model, Word Embeddings, Machine Learning, Convolutional Neural Networks.*

## 1. INTRODUCTION

Electronic health record (EHR) analysis can leverage great insights for improving the quality of human health care and it is one of the approaches to accomplishing precision medicine. However, there are a lot of challenges in analyzing such massive data set. One of the most challenging problems for our big Chinese EHR data analysis is its low data quality. The typical issues include missing values, inconsistency, and errors in the data columns, which hinder building robust machine learning models for data analysis. Since for such massive data set it is impossible to do data correction in a manual way, it is very desirable to develop some automatic algorithms which can make corrections and verifications for the individual problems in the big data. In this paper, we develop an artificial intelligence (AI)-based data governance strategy trying to leverage the power of AI algorithms in analyzing big data of patient clinic profiles to predict the missing values or verify if the existing values are correct and what they should be when they are wrong. Particularly we develop natural language processing (NLP), traditional machine learning and deep learning techniques for gender prediction and verification as a case study. Although it is only used for patient gender prediction here, the underlying fundamental principles of this methodology can be applied for the prediction of other kinds of missing values and verification of other kinds of existing values. Also, gender prediction from modeling user behavior profile plays a significant role in targeting and personalized product recommendation in the e-Commerce of digital advertising. For EHR analysis, patients' gender information plays a very important role in referring some useful information. However, in the Chinese EHRs we obtained, the gender information in many cases is either missed or not correct. Even if we can use the patients' Chinese identity (ID) information to extract the gender information, however, for privacy consideration, the ID information is usually hidden and not available for data analysis tasks. So, we need to develop an effective solution without using the patients' ID information. We also agree the point that it is possible to guess the patient's gender information with a pretty high accuracy from his/her name without resorting to the sophisticated AI algorithms, but we would like to emphasize that we are trying to develop a general methodology for data governance with AI algorithms for predicting all missing-values or verifying all existing values, not only for the specific gender prediction problem. For example, we are using this strategy to predict the category of each patient's diseases according to the Chinese national medical coding standard.

Through information fusion, we first construct each patient's clinic profile from the heterogeneous tables of symptom description, medical treatment process, lab tests, together with the doctor's prescription data, and take it as a document, trying to leverage some insights from such kind of clinic information by using supervised machine learning to predict and verify the patient's gender information. There are two basic components in the system. The first one is NLP for tokenization and the feature engineering for document's vector representation, and the second one is supervised machine learning with both traditional machine learning and/or deep learning algorithms. When using machine learning, we need to get the document's vector representation and we have tried 4 different representations: the bag of words (BOW) vectors with both TF-IDF weighting and binary representation for tokens, the averaged word embeddings of words in the document pretrained with theword2Vec [2] algorithm, and the document vector obtained with the doc2Vec [3] algorithm. For supervised machine learning algorithms, we have tried the multi-class support vector machine (SVM) [4, 5], one of the most efficient traditional machine learning algorithms for classification, and the convolutional neural networks (CNN) [6] of deep learning. For multi-class SVM, we have tried different vector representations for documents as the feature vectors for SVM, trying to get its best classification performance. Experimental results show that the CNN with inputs of word embeddings works best according to the testing performance based on the quantitative metric of F1-Score. It outperforms support vector machine (SVM) with different document vector representations. To our best knowledge, this is the first work in using NLP and deep learning for data governance in EHR analysis, especially for patient gender prediction from clinic profile.

Our contributions are as follows:

- We develop AI-based data governance strategy for EHRs. The AI algorithms include NLP, machine learning and deep learning algorithms.

- We construct patients' clinic profiles by using information fusion methodology from heterogeneous tables and records in the EHRs according to the domain knowledge of medical informatics for gender prediction. This makes it possible to leverage insights from the big EHR data using AI algorithms. The success of this methodology can be applied to the prediction of other missing values.

The rest of the paper is organized as follows. In Section 2, we discuss the methodology of feature extraction and vector representations for documents. Section 3 talks about training the SVM mode of traditional machine learning algorithms, and CNN of deep learning in text classification. In Section 4, we present the experimental results. We conclude the paper with discussions and future work in section 5.

## 2. METHODOLOGY

The system of AI-based data governance for gender prediction and verification consists of 3 components. Data preparation through information fusion, vector representation for documents with NLP algorithms and word embeddings for tokens, and traditional machine learning and deep learning algorithms for gender prediction and verification.

### 2.1 Data Preparation Through Information Fusion

The Chinese EHRs we are using for our projects consist of 179 heterogeneous tables. However, since creating EHRs is still at the beginning stage in China and in most of the time, doctors are more willing to write the paper notes for their patients rather than leave electronic notes in the computer system. So, in the 179 tables, many of them do not have much useful information for the empty columns in the tables. After preprocessing, we get 85 tables, but even in these 85 tables, many records still have missing values and we need to do additional filtering process to remove the useless records according to our application. We analyze these tables with the domain knowledge of medical informatics to construct individual complete and meaningful clinic profiles about each patient's individual symptom descriptions, lab tests, doctor's treatment plans, and prescriptions for medications. Then we concatenate the text data in a few columns from each patient's clinical profile and take the concatenated text data as a document for NLP and machine learning. Our motivation is to make full use of the massive clinic data for AI algorithms to predict the patients' gender information.

### 2.2 Vector Representation for Documents

Machine learning algorithms take individual documents as the inputs, so we first tokenize each document into a collection of terms. For this, we use Han LP[1], an open source software, for the Chinese document tokenization. Since there are no spaces between the Chinese words in each sentence, which is totally different from the English texts which can be separated by spaces between the individual words, the tokenization of Chinese texts needs specific algorithms and it is another hot research topic, here we only use the available tool for this task and focus our efforts on other things based on the tokenization result. Then we generate a vector representation for each document with different ways, which include the TF-IDF weighting method for tokens, binary representation for tokens, the averaged word embeddings of word2Vec[2],doc2Vec [3]for individual documents, and the vectors of words as the inputs of CNN [6] to get the vector representation for documents.

### 2.2.1The Bag of Words (BOW) Method

The BOW method makes use of tokenized individual words and/or N-Grams (N>=1)in a corpus as features for a document's vector representation, which is usually called a feature vector in machine learning and pattern recognition. All N-Grams constitute the vocabulary of the corpus. If N is equal to 1, the N-Gram is called unigram. For individual tokens, we usually have both binary representation and TF-IDF weighting representation to get the feature values. The binary representation does not count the number of occurrences of the tokens but only considers their presence and absence in the individual documents. If a token is present in the document, the vector's corresponding feature value is 1, otherwise 0. On the other hand, the TF-IDF weighting method takes the product of two statistics, the term frequency and inverse document frequency. The term frequency is simply the number of times that the term t appears in a document d. It assumes that the more frequent a token appears in the document, the more important it is for the topics of the document and it is usually calculated in the following augmented way [8]:

$$\boldsymbol{tf(t,d) = 0.5 + 0.5 * \frac{f_{t,d}}{\max\{f_{t',d}: t' \in d\}}} \tag{1}$$

where $\boldsymbol{f_{t,d}}$ denotes the frequency of term *t* in document *d*.At the same time, the inverse document frequency is calculated in the following way [8]:

$$idf(t,D) = log(\frac{N}{|\{d \in D: t \in d\}|} + 1) \tag{2}$$

With

- N the total number of documents in the corpus D, N= |D|
- The denominator $|\{\boldsymbol{d} \in \boldsymbol{D}: \boldsymbol{t} \in \boldsymbol{d}\}|$ is the number of documents where the term *t* appears. If the term *t* does not occur in the corpus, the denominator needs to be adjusted into $|\{\mathrm{d} \in \boldsymbol{D}: \boldsymbol{t} \in \boldsymbol{d}\}|+1$

The inverse document frequency is used to offset the impact of common words without having specialty. But the BOW method usually suffers from the following issues:

- Sparsity, most of the documents usually have only a small fraction of the vocabulary, and most of the words in the vocabulary are absent from individual documents, resulting in the term-document matrix with a lot of unwanted zeros.
- Does not take the word order information into account and only considers the occurrence of a word independent of the others, which is not true from both semantic and syntactic point of view. So, documents with different semantic meanings may be taken to be similar only if they contain the same words.

High dimensionality. Corpora generally have at least thousands of words (features). In addition to this, if the 2-grams and 3-grams are included, the number of features per document increases significantly. It could generate an even more sparse term-document matrix leading to insufficient RAM problem when we try to hold the entire matrix into the RAM. Not all features are important and modeling the data with such features needs a huge number of annotated samples for training, and it tends to lead to the overfitting problem for supervised learning when no sufficient annotated samples are provided. The high dimensionality of data challenges very much supervised machine learning algorithms for the curse of dimensionality, and in most of the time, we need to do dimensionality reduction for the BOW vector representations. But it is very critical in dimensionality reduction to preserve the structural information of the data

### 2.2.2 Word2Vec

Since the above BOW-based vector representation is not efficient to capture the semantic information from documents with the limitations of high dimensionality and sparsity, researchers have proposed different methods to represent documents and words in an embedded low-dimensional continuous vector space and the word2Vec [2] generates the state-of-the-art results. The Word2vec algorithm is such a distributed representation learning method to extract both semantic and syntactic information for individual words in a sentence. It consists of a bunch of related models that are used to produce the distributed representation of word embeddings. These models are the continuous bag-of-words (CBOW) and the skip-gram as shown in Figure 1. The CBOW model predicts the current word from its surrounding context words within a window centered at the current word, while for the skip-gram model, given the current word, it predicts the surrounding context words within a window for this current word. The Word2vec model is an unsupervised learning algorithm which can be trained with the hierarchical softmax and/or negative sampling method to reduce the computational complexity and make the learning process practical. In the hierarchical softmax method, a binary Huffman tree is constructed for the terms in a corpus according to their frequencies to reduce the computational complexity in updating the vectors of the terms. The benefits of using the Huffman tree is that all words are the leaf nodes in the tree, and high frequency words will have short paths from the root of the tree to such leaf nodes and low frequency words will have long paths. In the iterative backpropagation process, for updating each word's vector, we do not need to update the vectors of all other words in the vocabulary, but only need to update the vectors of the nodes in the path from the root of the tree to the leaf node in the tree. If the word is a frequent word, its corresponding path will be short and the further reduction of the computation complexity is accomplished. On the other hand, for the negative sampling method, it accomplishes this goal and improves the vector quality of low-frequency words byonly sampling a few negative samples from the vocabulary for updating their vectors. For this end,in the negative sampling method, the high-frequency words are down-sampled and the low-frequency words are up-sampled by lifting the low-frequencies.

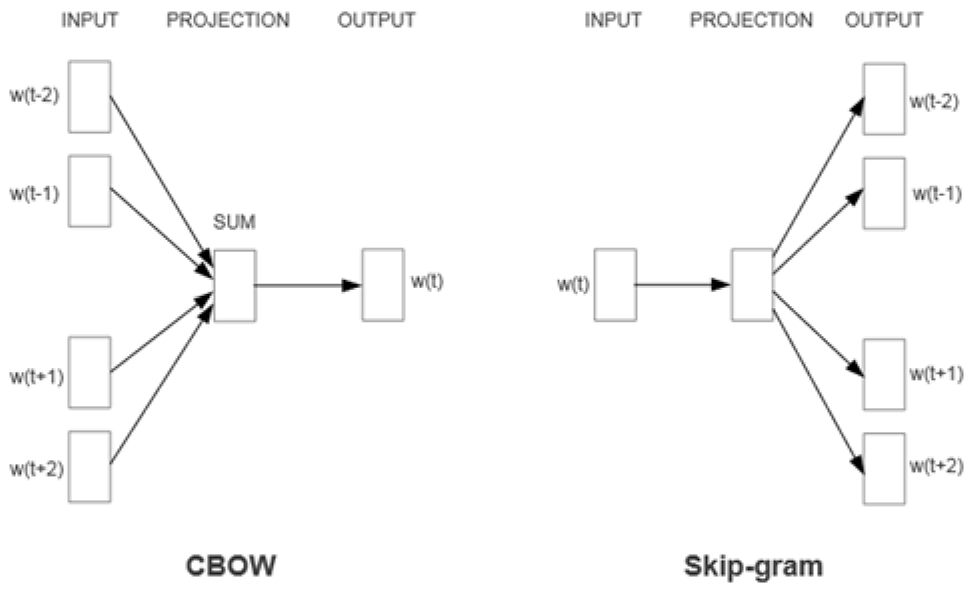

Figure 1. The illustration of CBOW and Skip-gram models in Word2Vec with courtesy of Mikolov etc. [2]

These models are the two-layer shallow neural networks. Word2vec takes as its inputs the high dimensional one-hot vectors of the words in the corpus and produces a vector space of several hundred dimensions which are much smaller than the size of the vocabulary, such that each unique word in the corpus is represented by a continuous dense vector in the embedded vector space. A very salient feature of this kind of vector representation with the word embeddings is that word vectors are such points in the vector space that for semantically similar words, their vectors are close to each other. This offers great benefit that we can infer the semantically similar words from the vector space if one word's vector is known, and it hence has been attracted with tremendous attention in text analysis. However, the word embeddings still have a limitation for representing documents. The usual way of using the word vectors is to take the averaged word embeddings of words in a document for document classification, sentiment analysis for movie reviews and customer service reviews, and text clustering analysis, but for most of the documents composed of syntactic sentences, this kind of average operation will introduce noise to the average result, making it deviate from the actual topic of the document.

### 2.2.3 Doc2Vec

The Doc2Vec algorithm [3] is an extension of Word2Vec for representing a document or a paragraph with a single unique real-valued dense vector learned together with the process of generating the vectors for individual words in the corpus. This is accomplished by assigning a unique document tag to each document, and the vector of this added document tag is learned together with all other words in the same document. When the learning process is done, the vector of the document tag is obtained and it is used for document analysis. Also, this model can be used to infer the document vectors for new documents. Just like word vectors generated by word2Vec, which provide semantic inference for individual words, a document vector generated by doc2Vec as shown in Figure 2, can be thought of reflecting some semantic and topic information of the document. As a result, the document vectors of similar documents tend to be close to each other in the vector space and they are very useful for document classification or clustering analysis.

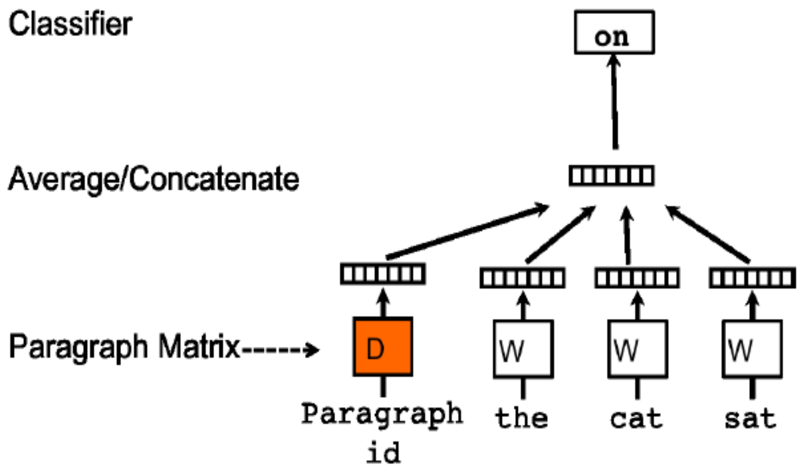

Figure 2. A framework for learning paragraph vector with courtesy of Quoc Le, etc. [3]

### 2.2.4 The CNN Architecture for Text Classification

CNN is one of the deep learning algorithms and it integrates feature extraction, feature selection and classification in a single architecture. It has been widely used for computer vision [7] and text classification [6]. As outlined by Figure 3, for computer vision, different channels of the image, like the R, G, B colors of the image, can be used as the inputs of CNN architecture for convolutional feature extraction. For text classification, the CNN usually takes as inputs the word embeddings of the sentence by stacking the words' vectors as a matrix according to the order of the words in the sentence. The embeddings can be either from word2Vec, one-hot representation or other vector representations of words in a sentence, forming different channels for representing the text data. With the CNN architecture, each channel of the texts can be represented as a matrix, in which, the rows represent the sequence of tokens or words in a sentence, and each row is a word's embeddings. The matrix is convolved with some filters of different sizes such as 3, 4, 5, but with the same dimension as the words' embedding vectors.

The main idea of CNN for text classification with different sizes of filters is to extract the semantic features of the N-Grams with the filters. The different filter sizes correspond to different numbers of the N in N-Gram. The words' vectors can be either from the pre-trained word embeddings such as those of word2Vec from large corpus, or randomly initialized. For the latter case, the word vectors are iteratively updated by backpropagation during the training process until the model is learned and the word vectors become the side effects of the CNN. Let's assume the filter size is $m$, sentence length is $l$, dimensionality of word embeddings is $d$, then the sentence matrix $x \in R^{l \times d}$and the filter can be represented as a matrix $w \in R^{m \times d}$. During the convolution process, each of the filters gradually moves down one word at a time along the sequence of words and at each position, the filter covers a portion of the words' vector matrix, i.e., m words' vectors in the matrix, and the point wise multiplication of the filter with the covered vector matrix is taken, and the multiplication results are summed up. This sum is then taken by the rectified linear unit (Relu) activation function together with a biased term $b \in R$to generate a feature value:

$$c_i = f(w \cdot x_{i:i+m-1} + b) \tag{3}$$

After the convolution process is done, a list of feature values is obtained like$c = [c_1, c_2, c_3, \ldots, c_{l-m+1}]$, which is regarded as the feature map of the filter. Finally,the max-pooling operation continues to take the maximum value from the feature map as the feature value of the filter's convolution result with the sentence. When all filters are applied for convolution with the sentence's vector matrix, we can get a list of feature values as the feature vector representation for the input sentence data. This feature extraction process with the max-pooling operation makes the length of the final feature vector independent of the input sentence length and the filter sizes.The length of the final feature vector is only dependent on the number of filters used for convolution. The final step in the CNN architecture is a full connection including the dropout strategy, regularization, and the Relu activation function from the final feature vectorwith the output layer and this full connection layer is fundamentally like the conventional neural network. The classification result of a sample is obtained by the softmax function applied to the output layer. The number of neurons in the output layer depends on the number of classes for the output.

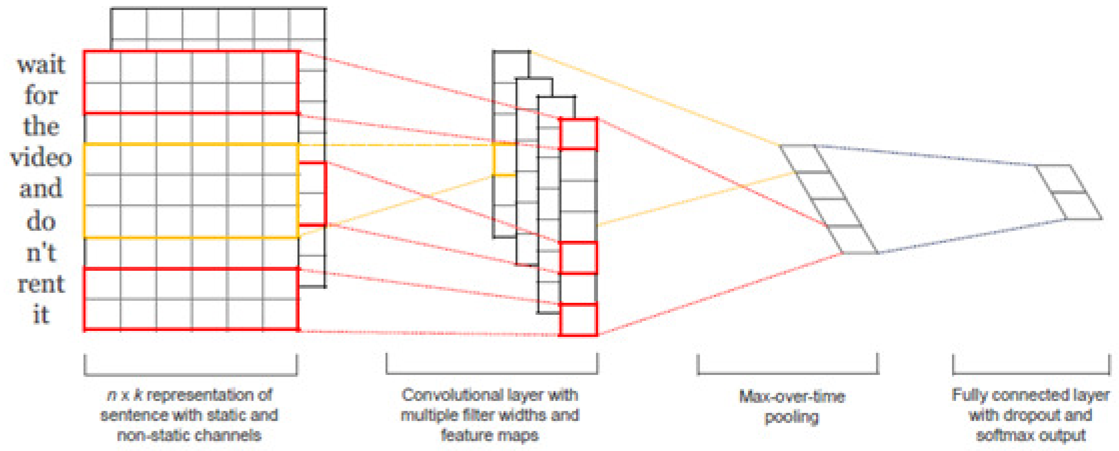
wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 3. The CNN architecture for text classification with courtesy of Yoon Kim [6].

## 3. TRAIN MACHINE LEARNING MODELS

We use machine learning algorithms to model patients' clinical profile for gender prediction. In the EHRs, many patients' clinical profiles do not have clear gender implications, it is very hard to predict the patients to be male to female. As a result, we add one more class of "unknown" in addition to the two classes of "male" and "female", forming a 3-class classification problem.

### 3.1 Training the CNN Model

The CNN is used for a 3-class classification problem. The Tensorflow's Python implementation of the CNN is used in this work. We use the pretrained word embeddings of word2Vec as the inputs of CNN. We use 100 filters for each filter size for feature extraction in the convolutional process. The corpus for training the word2Vec algorithm to get the word embeddings is obtained from the crawled Chinese medical documents. For the hyperparameters of CNN, through grid search, we set batch size 20, epochs size 40, the dimension of word embeddings200, dropout 0.5, and$l_2$-norm is used as the regularization. Additionally, a threshold is set 3 for clipping the gradient magnitude. We shuffle the samples in each epoch.

### 3.2 Training Multi-Class SVM

For performance comparison, the traditional machine learning algorithm of multi-class SVM is used as the baseline model. Although SVM belongs to the traditional machine learning models, it is very different from many of the other models like artificial neural network, decision tree, and Bayesian models for which the goal is to minimize the empirical learning risk, represented by the mean squared error of the training samples with respect to their predictions by the corresponding model. But for SVM, its goal is to minimize the structural risk by maximizing the margin between the two classes of the training samples. Minimizing the empirical learning risk does not have the guarantee for the model to generalize well to the unseen samples, but as shown in Figure 4, minimizing the structural risk by maximizing the margin between the two classes can guarantee for SVM to generalize well to the unseen samples, and this is what we are pursuing in training a machine learning model.

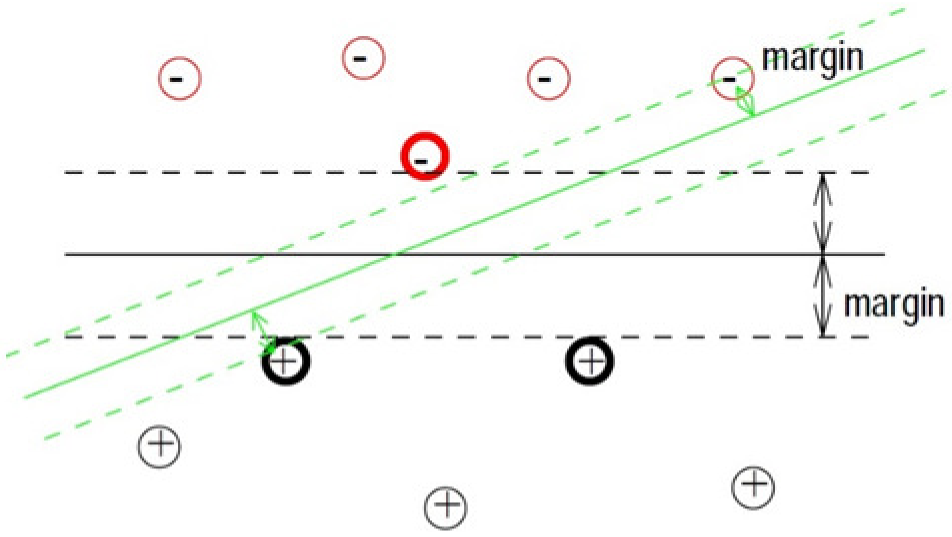

Figure 4. The illustration of the maximum margin for SVM

When training machine learning models, we do not want our model to work only well on the training data, but we want the model to be able to work well on the unseen data in the future, so the SVM model matches our goal of training a machine learning model. In Figure 4, we have two classes of samples denoted by empty red circles, and black circles, respectively, for classification and there are two possible margins for the hyperplane to be placed to separate the two classes of samples, one is labeled with the green color and the other is labeled with the black color. We can clearly see that the hyperplane marked with the black color can generalize better than the one with the green color because it has a wider margin. To train the SVM model, its input feature vectors of documents are from either doc2Vec, or the averaged word embeddings of all words in the document with word2Vec, or the TF-IDF vectors, or the binary vectors of tokens, respectively. Furthermore, in our experiment, only linear SVM is used and no kernel SVM is used for the fact that the dimensionality of the feature vector is already high. For the implementation of the multi-class SVM, we use the Python tool of Scikit-learn, which is a free software of machine learning library in Python programming language. We use the grid search method to optimize the hyperparameters of SVM.

## 4. EXPERIMENTAL RESULTS AND ANALYSIS

In the experiment, we compare CNN with SVM, one of the most popular traditional machine learning algorithms. Also, 4 different vector representations for documents are used as the input feature vectors for SVM. When obtaining the pre-trained word vectors with word2Vec, we have tried 4 models:

- CBOW + hierarchical softmax,
- CBOW + negative sampling,
- Skip-Gram +hierarchical softmax, and
- Skip-Gram + negative sampling.

For our dataset, the CBOW+negative sampling works best and it is selected to generate the word embeddings for CNN's inputs, and the averaged word embeddings for the SVM. In this paper, the quantitative metrics used for measuring the performance of the models are the precision (accuracy), recall, and F1-score and they are calculated in the following way:

$$\boldsymbol{Precision = \frac{tp}{tp+fp}} \tag{4}$$

$$\boldsymbol{Recall = \frac{tp}{tp+fn}} \tag{5}$$

$$\boldsymbol{F_1\_score = 2 \cdot \frac{precision \cdot recall}{precision + recall}} \tag{6}$$

Where *tp* denotes true positives, *fp* denotes false positives, and *fn* denotes false negatives. The prediction results of CNN with word embeddings as inputs is listed in Table 1.

For training the SVM model, we have compared the binary BOW vector with the TF-IDF BOW vector. However, it is found out that the performance of SVM with the TF-IDF vector representation is far below that with the binary BOW vector representation, so its result is not listed in Table 2 and only the binary vector representation is used as the method of BOW vectors. Our analysis figures out that the vector representation based on TF-IDF weighting method calculated with formula (1) and (2) is suitable for long documents with many words because it introduces some additional "smoothing" effect, but for our short documents, the binary vector representation makes more sense to represent them. The experimental results with comparative studies show that the CNN works best and it outperforms the SVM model with different feature vectors for document representation. We think this is mainly for the following reasons. The vector representation of a document with the doc2Vec is trying to summarize the document's topic, but for our documents, it cannot extract very much topic information because the documents are very short and they are composed of only a list of words about the symptoms of the disease from different aspects without containing any syntactic information. The fact that the averaged word embeddings ofword2Vec accomplishes better performance than the doc2Vec for SVM, is that the words in each document are related to each other about the disease, and to some extension, they can be regarded as the synonyms of each other in our EHR corpus, and their vector representations in the embedded vector space are close to each other. As a result, when each of our documents does not have any syntactic information and is only composed of such words, the averaged vector of these words' vectors will still be close to the vectors of these words and it can be roughly taken to be the "center" of these words' vectors. So, for our documents, the averaged word embeddings represents each document better than the document vector obtained with the doc2Vec algorithm. But since we still do not have very sufficient data to train the word2Vec algorithm for generating high-quality word embeddings, its performance with SVM is still lower than that of SVM with the binary BOW vector. It is reported that for generating the high-quality word embeddings with word2Vec [2], a very huge corpus of 100 billion words from Google news is used for training, so, their obtained word vectors can sufficiently capture the semantic information of words. Furthermore, for the deep learning structure CNN, it further extracts effective features from the input word embeddings for classification by using different sizes of filters to extract the N-Gram semantic information in the texts, nonlinear activation functions, max-pooling operations, dropout sampling for preventing the correlations in features, and an additional learning layer through the full connection with the output layer. As a result, even if the deep learning architecture in this paper is only the shallow CNN, it could accomplish the best performance among all these models. But for the multi-class SVM with different document's vector representations as inputs, it simply takes what is provided as the feature vector for learning, and there is no additional feature engineering work as done in CNN to re-extract and re-select the most effective features from the input features for classification.

Table 1. The prediction results of CNN with word embeddings as inputs.

| Input Vectors | F1-Score | Accuracy | Recall |
|---|---|---|---|
| Word2Vec | 0.96 | 0.94 | 0.97 |

Table 2. The prediction results of SVM with different feature vectors for document representation.

| Input Vector(s) | F1-Score | Accuracy | Recall |
|---|---|---|---|
| Binary BOW | 0.93 | 0.97 | 0.90 |
| Avg. Word2Vec | 0.91 | 0.97 | 0.86 |
| Doc2Vec | 0.66 | 0.60 | 0.74 |

## 5. CONCLUSION AND FUTURE WORK

In this paper, we develop NLP and supervised machine learning based data governance strategy for EHR analysis and demonstrate its effectiveness in gender prediction. Two kinds of machine learning algorithms are investigated. They are the traditional machine learning algorithm SVM and deep learning algorithm CNN. Four kinds of vector representations are investigated for document's vector representation for the multi-class SVM. From our experimental results, the deep learning CNN algorithm outperforms the state-of-the-art traditional machine learning algorithm SVM. This AI-based data governance strategy can be applied to any other data prediction and verification problem to improve the data quality and leverage great insights from the big data. In the next step, we will continue to investigate the deep learning classification algorithm by either adding more layers of convolutions into the current shallow CNN structure or using other deep learning architectures such as the LSTM and bi-directional LSTM networks for document representation. Also, we hope to be able to get more data from the hospitals to completely demonstrate the power of AI in big EHR analysis.

## ACKNOWLEDGEMENTS

The authors would like to thank the support from both the Technology Research Centerand the Business Unit of Medical Health in the Software Business Group of Inspur Inc, and the Health Planning Commission of Shandong Province, China.

## REFERENCES

[1] HanLP,https://datascience.shanghai.nyu.edu/hanlp.

[2] Mikolov Tomas, Chen Kai, Corrado Greg, Dean Jeffrey (2013) "Efficient Estimation of Word Representations in Vector Space", Advances in neural information processing systems, pp3111-3119.

[3] Le Quoc, Mikolov Tomas (2014)"Distributed Representations of Sentences and Documents", Proceedings of 31 International conference on machine learning, pp1188-1196.

[4] V. N. Vapnik (1999)"An overview of statistical learning theory", IEEE Trans. Neural Network, vol. 10, pp988-999.

[5] Yoon Kim (2014)“Convolutional Neural Networks for Sentence Classification”, Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp1746-1751.

[6] Alex Krizhevsky, Ilya Sutskever, Geoffery Hinton (2012) “ImageNet Classification with deep convolutional neural networks”,NIPS'1, Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, pp1097-1105.

[7] https://en.wikipedia.org/wiki/Tf–idf

## AUTHORS

**Junmei Zhong** received the B.Sc. degree in Computer Science from Dalian University of Technology, China, in 1988, the Master’s degree in Computer Science, Nankai University, Tianjin, China, in 1993, where he received the “Excellent Thesis Award”,the Ph.D. degree from Electrical & Electronic Engineering, The University of Hong Kong in 2000, where he received the prize of "Certificate of Merits for Excellent Paper", Awarded by IEEE Hong Kong Section and Motorola Inc, Dec. 1998.

He has been the Chief Data Scientist at Inspur USA Inc since March 2017. His R&D interests include machine learning, data mining, NLP, text mining, digital advertising, graph theory, knowledge graph, deep learning, signal processing, wavelets, image analysis, pattern recognition, and computer vision.

Before joining Inspur USA Inc, He was the Senior Principal Data Scientist at Spectrum Platform Company and Twelvefold Media Inc for content-based display advertising, Principal Data Scientist at Pitchbook Data Inc about NLP and text mining. Dr. Zhong was the research faculty in University of Rochester, NY, USA from 2002 to 2004, and Assistant Professor in Cincinnati Children's Hospital Medical Center, Ohio, USA from 2004 to 2006. He has generated many scientific papers published on prestigious journals and top conference proceedings.

**Xiu Yi** received the B.Sc. degree from Dept. of Computer Science and Technology, Harbin Engineering University, Harbin, China, in 2009, the Master’s degree in Computer Application Technology, Harbin Engineering University, Harbin, China, in 2012.

Since 2014, she has been a software engineer in the Technology Research Center of Inspur Software Business Group Co. Ltd., Jinan, Shandong Province, China, for machine learning, data mining, NLP, deep learning, text mining, and computer vision in OCR, face recognition and auto license plate recognition. From 2012 to 2014, she was a software engineer in Baidu, Beijing for NLP, NER and relation prediction with knowledge graph. She is proficient in C, Java and Python programming.

**Jian Wang** received the B.Sc. degree in Applied Mathematics from Shandong University, China in 2014, the Master’s degree of Science from Shandong University,Jinan, Shandong Province, China in 2017.

Since 2017, he has been an assistant engineer in the Technology Research Center of Inspur Software Group Co. Ltd. His current work is mainly aboutNLP, data mining, machine learning,deep learning and software development in Python for big data analysis.

**Zhuquan Shao** received B.Sc. degree in Applied Mathematics from Dezhou University, Shandong Province, China in 2014, the Master's degree of Science from Dalian Maritime University, Dalian, China in 2017.

Since 2017, he has been an assistant engineer in the Technology Research Center of Inspur Software Business Group Co. Ltd. His current work is mainly in machine learning, big data analysis, data warehouse, and software development in SQL and Python.

**Panpan Wang** received the B.Sc. degree with the major of Statistics from QuFu Normal University, Shandong Province, China in 2014, the Master's degree of Science from Dalian University of Technology, Dalian, China in 2017.

Since 2017, She has been working in the Technology Research Center of Inspur Software Business Group Co. Ltd., Jinan, Shandong Province, China. Her current work is mainly about deep learning and machine learning in the field of medical healthcare.

**Sean Lin** received the B.Sc. degree in Environment and Design from University of Jinan, Jinan, Shandong Province, Chinain 2007.

Since 2007, he has been the principal engineer in user experience design, big data analysis, project management, visualization and data governance. He worked in Guiyuan Tech Ltd. in Jinan, China, for project management and UI design from 2007 to 2010. He received the Certificate of System Integration and Project Management Engineer in 2014.