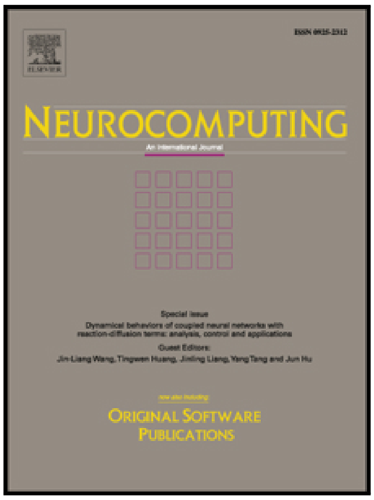
Accepted Manuscript



Discriminant document embeddings with an extreme learning machine for classifying clinical narratives

Paula Lauren, Guangzhi Qu, Feng Zhang, Amaury Lendasse

PII: S0925-2312(17)31415-7
DOI: 10.1016/j.neucom.2017.01.117
Reference: NEUCOM 18805

To appear in: *Neurocomputing*

Received date: 8 July 2016
Revised date: 31 December 2016
Accepted date: 3 January 2017

Please cite this article as: Paula Lauren, Guangzhi Qu, Feng Zhang, Amaury Lendasse, Discriminant document embeddings with an extreme learning machine for classifying clinical narratives, *Neurocomputing* (2017), doi: 10.1016/j.neucom.2017.01.117

Discriminant document embeddings with an extreme learning machine for classifying clinical narratives

Paula Lauren[a,], Guangzhi Qu[a], Feng Zang[b], Amaury Lendasse[c,d]

[a] Computer Science and Engineering Department, Oakland University, Rochester, USA
[b] China University of Geosciences, Wuhan, China
[c] Department of Mechanical and Industrial Engineering The University of Iowa, Iowa City, USA
[d] Department of Business Management and Analytics, Arcada University of Applied Sciences, Helsinki, Finland

Abstract

The unstructured nature of clinical narratives makes them complex for automatically extracting information. Feature learning is an important precursor to document classification, a sub-discipline of natural language processing (NLP). In NLP, word and document embeddings are an effective approach for generating word and document representations (vectors) in a low-dimensional space. This paper uses skip-gram and paragraph vectors-distributed bag of words (PV-DBOW) with multiple discriminant analysis (MDA) to arrive at discriminant document embeddings. A kernel-based extreme learning machine (ELM) is used to map the clinical texts to the medical code. Experimental results on clinical texts indicate overall improvement especially for the minority classes.

Keywords: Document classification, Feature learning, Word embeddings, Document embeddings, Skip-gram, PV-DBOW, Multiple discriminant analysis, Extreme learning machines, Clinical narratives

1. Introduction

Clinical narratives contain the clinical encounter as observed by the healthcare professional with a patient. The data from clinical narratives enable qual-

*Corresponding author
Email address: gqu@oakland.edu (Guangzhi Qu)

ity assessment programs [1], improve patient safety [2], support evidence-based medicine [3], improve surveillance of infectious diseases [4], support clinical trials [5], and assist with various other clinical research programs [6]. The unstructured nature of clinical narratives allow clinicians ease of input, but their inherent lack of structure makes it difficult to automatically extract knowledge [7].

Document classification is a sub-discipline of natural language processing (NLP) that pertains to a process for assigning one or more labels from an existing set of labels. A problem in document classification relates to the classification of unstructured text from the document [8]. For instance, in sentiment analysis the objective is to assign a label to denote the sentiment of the text as being being positive or negative [9]. Some other applications of document classification include language identification [10] and genre classification [11]. Identifying relevant features is an important precursor to accurate classification. In addition, a system that can automatically identify features from text is preferred over the cumbersome task of manually selecting the features.

Feature generation using the traditional bag-of-words (BoW) model [12] generates features from a document based on word frequencies. The BoW model has been successfully applied to various applications in the medical domain with automating medical image annotation [13], biomedical concept extraction [14], and recommender systems for medical events [15]. Another traditional approach is based on n-gram frequency statistics [16], for automatically rendering features. An n-gram is a sequence of n items from text, when n=3 (a trigram) the consecutive three words are considered a feature. The sequential aspect of n-grams permits the preservation of word order unlike the BoW model [12]. The n-gram approach has been successful in medical applications for identifying features in categorizing radiology reports [17], identifying novel synonyms or symptoms associated with a medical drug [18], and sentence sub-graph mining from pathology reports [19]. Latent semantic analysis (LSA) [20] is a feature extraction method that generates features by applying truncated singular value decomposition (SVD) [21] to a word co-occurrence matrix. In the medical domain, LSA has been successful in automating analysis of speech in psychiatric disorders [22], finding thematic correlations of patients with severe depression [23], and automatic grading of clinical case summaries [24]. Latent dirichlet allocation (LDA) [25] is a generative model that assigns topics to documents, rendering topic distributions over words with use in feature generation for document classification. A few medical applications using LDA include mining cancer clinical notes [26] and searching as well as creating clinical trials [27].

A word embedding is a learned distributed representation of a word, consisting of a vector of continuous real values that represent the word. The essential idea is that words that are used in similar contexts will be represented by similar vectors. Word embeddings generated from a neural network jointly represents the probability of word sequences from natural text, also known as a neural network language model (NNLM) [28, 29, 30]. Continuous skip-gram [31] is a type of NNLM that performs unsupervised feature learning, with the implementation known as Word2Vec[1]. Word2Vec is a feedforward neural network that uses the words from a vocabulary as the input into the network and *embeds* them as vectors that are projected into a lower dimensional space. Skip-gram has been used to find the semantic similarity between medical concepts directly from electronic health records, as an alternative to Systematized Nomenclature of Medicine-Clinical Terms (SNOMED-CT) [32]. SNOMED-CT is a vocabulary of clinical terms that organizes medical concepts into hierarchies and semantic networks. The skip-gram architecture has been extended to find similarities at the sentence, paragraph and document level with paragraph vectors, which learns a fixed-length feature representations from the variable-length of documents [33]. Paragraph vectors have two methods; paragraph vectors-distributed memory (PV-DM) and paragraph vectors-distributed bag of words (PV-DBOW). PV-DM creates paragraph embeddings simultaneously with word embeddings and PV-DBOW renders just paragraph embeddings [33]. Paragraph vectors that are used for creating feature-length representations of entire documents have been

[1] `https://code.google.com/archive/p/word2vec`

referred to as *document embeddings* [34, 35] in the research literature. Since this research uses paragraph vectors at the document level, the term document embeddings will be adopted as well for consistency.

Dimensionality reduction enables machine learning algorithms to be more effective and efficient by the removal of irrelevant features with subsequent noise reduction [36]. Feature reduction is particularly important when the number of features p are greater than the number of observations n. Commonly referred to as the high-dimensional problem of $p_{>>}n$, inevitably leads to overfitting of a model [37]. The general rule of thumb is to have at least five or ten times as many samples as variables [38]. An approach to reduce the dimensionality is the combination of features by projecting the data into a low-dimensional subspace that captures the essential data [39]. (PCA) [36] and linear discriminant analysis [40] are two popular approaches that are used to reduce the dimensionality of the feature space [40]. Multiple discriminant analysis (MDA) is the generalization of linear discriminant analysis[2]. The main distinction between these two approaches, PCA maximizes variance in the data and MDA maximizes the separation between multiple classes.

Machine learning methods such as neural networks can be applied to document classification tasks. A type of neural network called an extreme learning machine (ELM) [41] is well-known for its efficiency and accuracy in classification, as well as in regression, feature learning and clustering tasks [42]. The efficiency of ELM can be attributed to the random non-updated weights connecting the input to the hidden neurons and a singular learning of the weights from the hidden neurons to the output, with the latter resulting in a linear model. The simplicity of ELM is in stark contrast to the common training method of neural networks that uses back-propagation along with an optimization method such as gradient descent to achieve the optimum weights. In this research ELM has been used for the classification of clinical narratives. In Section 5, our future

[2]LDA already denotes latent dirichlet allocation so the term MDA will be used instead. Also, MDA is more applicable due to the multi-class dataset that was used in this research.

work will discuss some ideas for using ELM for feature learning in deriving word embeddings.

The organization of this paper is as follows: Section 2 describes the related work and background for this research. Section 3 describes the proposed approach along with the experiments that were conducted. Section 4 reports on the results along with evaluation, Section 5 provides a discussion and describes future work, and Section 6 is the conclusion.

2. Related work and background

This section describes the related work and background for understanding the primary methods used in this paper which include the skip-gram model, PV-DBOW model, MDA, and kernel-based ELM.

2.1. Related work

Pre-trained word embeddings incorporating domain expertise were generated from skip-gram using millions of PubMed article titles as well as abstracts and the word similarities were averaged for a disease predictive model [43]. The authors predict the onset of five adverse outcomes related to cardiovascular disease, which are stroke, heart failure, heart attack, diabetes and high blood cholesterol from an insurance research database. The prediction task used a regularized logistic regression model.

Feature generation using the PV-DM model and ELM for classification was done in a text classification task [44]. As mentioned in Section 1, PV-DM is a feedforward neural network that generates a feature vector for the document and for each word from the vocabulary. The dataset used in their study entailed 25,000 bibliography records with five equally balanced classes. The best results were achieved using a sigmoid activation function in the ELM hidden layer. A sigmoid function is one of several mapping functions used in ELM [42]

Random projection is a dimensionality reduction method that projects a set of points from a high-dimensional space to a randomly chosen low-dimensional subspace while preserving the pairwise distances [45]. A random projection-extreme learning machine (RP-ELM) combines the feature mapping of ELM with random projection [46]. RP-ELM compared to ELM without random projection had been done using two binary classification gene datasets for colon cancer and leukemia [46].

Addressing the potential of overfitting, a text categorization method based on regularization with ELM referred to as RELM was proposed [47]. The datasets used in the study consisted of single-label and multi-label data. Multi-label data is when more than two class labels are assigned to a document. The RELM approach performed well on single-label and multi-label data. LSA was used for representing the features from the text. The experiments were performed using a radial basis function for the activation function in the ELM hidden layer.

This paper proposes a semi-supervised approach that applies MDA separately to skip-gram and PV-DBOW for document embeddings. The feature sets generated from these two approaches are then combined to arrive at what will be referred to as *discriminant document embeddings*. This research combines methods that are typically used separately as mentioned previously in this section with [44] using skip-gram and [43] utilizing PV-DM. A kernel-based ELM is used for the classification of the combined reduced feature set. A comparative study is also done with the other methods; BoW, N-gram, LSA and LDA. This is a semi-supervised document classification task that uses a highly imbalanced clinical corpus pertaining to hip replacement surgery data. No medical domain expertise had been used in this study. The results achieved show that the proposed combined method of discriminant document embeddings provides an improvement, especially with regard to the minority classes from the dataset.

2.2. Skip-gram model

As mentioned in Section 1, the skip-gram model is an unsupervised feature learning algorithm. Skip-gram predicts the neighboring words also known as the word's context, from each word in a sentence. Given the training words $w_1, w_2, ..., w_N$, where N refers to the total word count, the following objective function is maximized:

$$P = \frac{1}{N} \sum_{n=1}^{N} \left(\sum_{-c \leq j \leq c, j \neq 0} \log p(w_{n+j}|w_n) \right) \tag{1}$$

The outer summation represents the words from the training corpus. The inner summation spans the the *left* context $-c$ and the *right* context c, computing the log probability of predicting the word context w_{n+j}, given the input word w_n. The basic skip-gram equation defines $p(w_{n+j}|w_n)$ with the softmax function resulting in outputs that sum to one. This probability distribution is defined as:

$$p(w_{n+j}|w_n) = \frac{\exp\,(u_{w_{n+j}}^T v_{w_n})}{\sum_{v=1}^{V} \exp\,(u_v^\top v_{w_n})} \tag{2}$$

The vocabulary is denoted as V, the input u_w and output v_w are vector representations of word w. The input vector consists of several words that are added together to predict the context word. A normalized hierarchical softmax objective function makes the skip-gram model more efficient [48, 49]. The efficiency is made possible by approximating the probability distribution in Equation (2) using a huffman binary tree [50] that is used for the output layer. The leaves of the huffman tree represent the words and each child node contains the relative probabilities. An improvement in training time is due to a reduction in computational complexity for log $p(w_{n+j}|w_n)$ [51]. Stochastic gradient descent [52] is the optimization method used in the skip-gram model. Initially the weights of the network are randomized, after each target word prediction task the error is back-propagated through the network. Training completes with a word vector for every word in the vocabulary capturing the distributional representation of the words, the word vectors are the word embeddings.

2.3. PV-DBOW model

The PV-DBOW model for generating document vectors is similar to the skip-gram model described in Section 2.2 for generating word vectors. The distinction from skip-gram is that PV-DBOW uses a unique token to identify the document, which is the input for generating the document vectors [33]. Equations (1) and (2) for the skip-gram model also apply to the PV-DBOW model with a slight caveat. Specifically, w_n is now replaced with a document vector d_n in $p(w_{n+j}|d_n)$. The PV-DBOW model predicts words that have been randomly sampled from the paragraph in the output, making $p(w_{n+j})$ still valid. The PV-DBOW model is also trained using stochastic gradient descent [33].

2.4. Multiple discriminant analysis (MDA)

Maximizing *between-class* distances while simultaneously minimizing *within-class* distances is how MDA achieves class discrimination. The two matrices of interest are the *between-class* scatter matrix $\mathbf{S}_b$ and the *within-class* scatter matrix $\mathbf{S}_w$. Suppose there are c classes, let M_j be the total number of samples in class j, where $j = 1, 2, ..., c$. The total number of samples is $M = M_1 + M_2 + ...$ $+M_c$. For each class j, let the sample mean be noted $\bar{x}_j$ and the sample mean for the entire dataset $\bar{x}$. Let x_{jk} be the k^{th} pattern from class c_j, so:

$$\bar{x}_j = \frac{1}{M_j}\sum_{k=1}^{M_j} x_{jk} \tag{3}$$

$$\bar{x} = \frac{1}{M}\sum_{j=1}^{c} M_j\bar{x}_j = \frac{1}{M}\sum_{j=1}^{c}\sum_{k=1}^{M_j} x_{jk} \tag{4}$$

The *between-class* $\mathbf{S}_b$ and *within-class* $\mathbf{S}_w$ matrices are given by:

$$\mathbf{S}_b = \sum_{j=1}^{c} M_j(\bar{x}_j - \bar{x})(\bar{x}_j - \bar{x})^\top \tag{5}$$

$$\mathbf{S}_w = \sum_{j=1}^{c}\sum_{k=1}^{M_j}(\bar{x}_{jk} - \bar{x_j})(\bar{x}_{jk} - \bar{x})^\top \tag{6}$$

The objective of MDA is to find the projection matrix that maximizes $|\mathbf{S}_b|/$ $|\mathbf{S}_w|$. This ratio is known as Fisher's criterion [40] given by:

$$\mathbf{W} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W}^T\mathbf{S}_b\mathbf{W}|}{|\mathbf{W}^T\mathbf{S}_w\mathbf{W}|} \tag{7}$$

Equation (7) is maximized when the projection matrix $\mathbf{W}$ is composed of the eigenvectors $\mathbf{S}_w^{-1}\mathbf{S}_b$:

$$\mathbf{W} = eig(\mathbf{S}_w^{-1}\mathbf{S}_b) \tag{8}$$

There will be at most $(c-1)$ nonzero eigenvectors and eigenvalues [53].

2.5. Extreme learning machines (ELM)

ELM is a two layer feedforward neural network where the hidden layer weights are set randomly and the output layer weights are computed from the training data [41, 54]. Consider a dataset containing N training examples $[(\mathbf{x}_i, y_i)]_{i=1}^{N}$ where $\mathbf{x}_i \in \mathbb{R}^n$ is the input and $y_i \in \mathbb{R}$ is the desired output. Let ℓ define the number of hidden neurons and $g(.)$ represents the activation function:

$$y_j = \sum_{i=1}^{\ell} \beta_i g(\mathbf{w}_i \,.\, \mathbf{x}_j + b_i), \qquad j = 1, 2, ..., N \tag{9}$$

Here, the weight vector $\mathbf{w}_i = [w_{i1}, w_{i2}, ..., w_{in}]^T$ connects the i^{th} hidden neuron and the input neurons, b_i is the bias of the i^{th} hidden neuron, and β_i is the weight that connects the i^{th} hidden neuron with the output neuron. In matrix form:

$$\mathbf{y} = \mathbf{H}\beta, \tag{10}$$

where,

$$\mathbf{y} = [y_1, y_2, ..., y_N]^T \tag{11}$$

$$\mathbf{H} = \begin{bmatrix} g(w_1x_1 + b_1) & ... & g(w_\ell x_1 + b_\ell) \\ \vdots & ... & \vdots \\ g(w_1x_N + b_1) & ... & g(w_\ell x_N + b_\ell) \end{bmatrix}_{N\times\ell} \tag{12}$$

$$\beta = [\beta_1, \beta_2, ..., \beta_\ell]^T \tag{13}$$

Typically, $\mathbf{H}$ will be a nonsquare matrix so there may not exist $\mathbf{w}_i, b_i, \beta_i$, where $i = 1, 2, ..., N$ such that $\mathbf{y} = \mathbf{H}\beta$. The least-square solution of this linear system is:

$$\hat{\beta} = \mathbf{H}^{\dagger}\mathbf{y} \tag{14}$$

where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix $\mathbf{H}$ [42].

2.5.1. Kernel ELM

To improve ELM's generalization performance, a kernel-based ELM was proposed [55, 42]. The ELM kernel matrix has two forms: $\mathbf{H}^T\mathbf{H}$ and $\mathbf{H}\mathbf{H}^T$. The reduced feature space provided by MDA in Section 2.4 results in training patterns being significantly larger than the hidden neurons so Equation (15) is applicable as [42]:

$$\beta = (\mathbf{H}^T\mathbf{H} + \frac{I}{\lambda})^{-1}\mathbf{H}^T\mathbf{y} \tag{15}$$

where I is the identity matrix and λ is a regularization coefficient. The output function for the ELM classifier is expressed as [55]:

$$f(x) = h(x)\beta = h(x)(\mathbf{H}^T\mathbf{H} + \frac{I}{\lambda})^{-1}\mathbf{H}^T\mathbf{y} \tag{16}$$

If the feature mapping function $h(x)$ is unknown then a kernel function $K(x_i, x_j)$ can be used as shown in [55]. The kernel matrix is defined as [55]:

$$\Omega_{ELM} = \mathbf{H}^T\mathbf{H} : \Omega_{ELM_{i,j}} = h(x_i).h(x_j) = K(x_i, x_j) \tag{17}$$

The output function of the ELM classifier can be compactly expressed as [55]:

$$f(x) = h(x)(\mathbf{H}^T\mathbf{H} + \frac{I}{\lambda})^{-1}\mathbf{H}^T\mathbf{y} \tag{18}$$

$$f(x) = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T (\frac{I}{\lambda} + \Omega_{ELM})^{-1}\mathbf{y}$$

Various activation functions are used with kernel-based ELM. For this research, the radial basis function (RBF) kernel was utilized, also known as a Gaussian kernel [42, 56].

3. Methodology and experiments

This section describes the data, preprocessing of the data, the proposed approach, and the experiments that were conducted for this research.

3.1. Describe data

The dataset utilized in this research study is highly imbalanced as illustrated in Figure 1. The highest class $C1$ has a total count of 2,252 clinical narratives with the lowest class $C5$ having 62 clinical narratives.

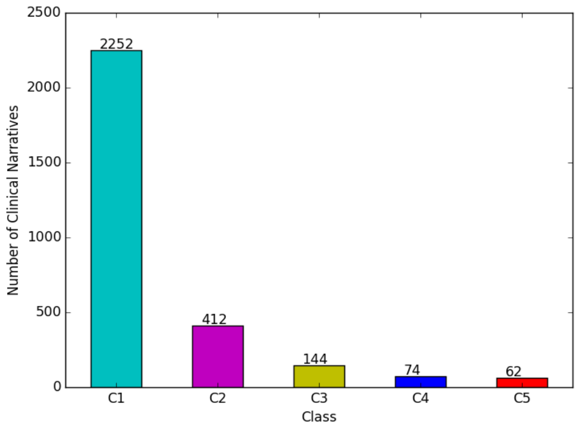


Figure 1: The total counts of narratives per class.

The class skewness illustrated in Figure 1 also extends to the narrative character length with the imbalance illustrated in Table 1.

Table 1: The character length for the clinical narratives.

Mininum	Average	Maximum
169	4,730	22,234

The data used in this study consists of clinical narratives that pertain to hip replacement surgery, also known as arthroplasty. The total number of clinical narratives used for this research study were 2,944. Each clinical narrative had one associated label out of five possible labels pertaining to the current procedural terminology (CPT) codes. The CPT codes are used to document the procedure that had been done. The CPT codes have similarity to the International Classification of Diseases, Tenth Revision, Clinical Modification (ICD-10-CM)[3] which documents the care process. Table 2 contains a description of the five CPT codes from the hip replacement surgery data used in this study.

Table 2: The description of classes for hip replacement surgery.

.

Class	CPT code	Description
C1	27130	acetabular proximal & femoral prosthetic replacement
C2	27132	conversion of previous hip surgery
C3	27134	revision of cetabular & femoral components
C4	27137	revision of cetabular component
C5	27138	revision of femoral component

Figure 2 provides an illustration of the process flow for the following Sections 3.2-3.9. Each process in Figure 2 corresponds to its respective sub-section in this section. Also, illustrated in Figure 2 are the output feature vectors of each process that are the input to the next process.

3.2. Preprocess data

In NLP, preprocessing text usually involves stemming as well as stopword removal. With the contextual aspect of the skip-gram and PV-DBOW models, these prepocessing steps are unnecessary. Generating the distributed word representation or document representation in the form of word or document embeddings, relies on the actual words and their placement in the training corpus.

Multi-words are usually regarded as a single term in linguistic processing, especially with medical corpora where multi-word terms are plenteous. For example, the medical term *glucose metabolism disorders* denotes the multi-word

[3]http://www.cdc.gov/nchs/icd/icd10cm.htm

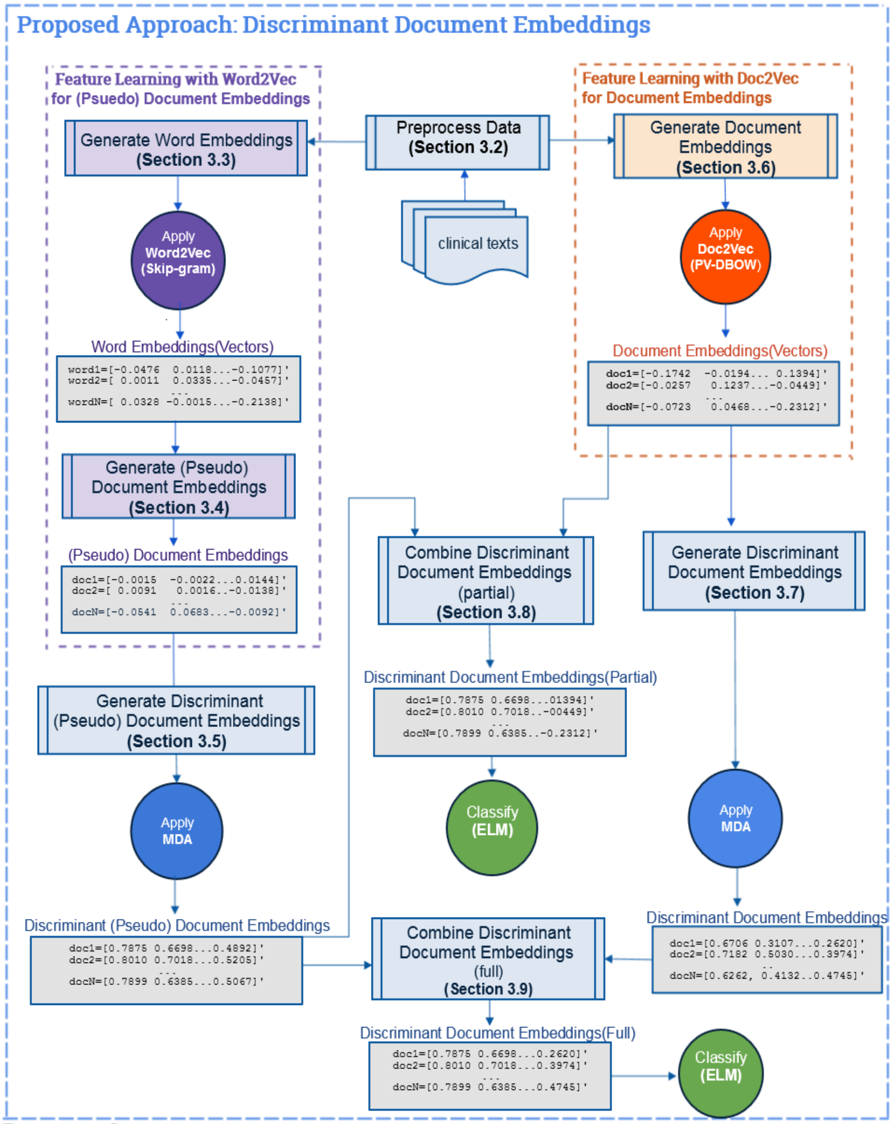


Figure 2: Illustration of Sections 3.2-3.9 for the proposed approach.

term: *glucose_metabolism_disorders* in the vocabulary [57]. In this research, no preprocessing was done to holistically represent multi-words as one term. The formation of multi-word terms are likely to be redundant because of the context-focus aspect of a neural network language model.

Preprocessing entailed the removal of non-letters from the corpus and the conversion of upper case words to lower case. Each narrative was also split into lists of sentences consisting of tokenized words. The construction of the sentence list was identified by each period in the narrative. The training corpus consisted of 126,585 sentences and significant variation was also noted in each sentence as illustrated in Table 3.

Table 3: The sentence word length for the clinical narratives.

Mininum	Average	Maximum
1	13	137

Training and testing datasets were rendered from the corpus with a respective 75% and 25% split, utilizing the stratified holdout [58] method. Stratification with the holdout approach on the training and testing datasets is done to ensure that training and testing have a balanced distribution of classes.

3.3. Generate word embeddings

This research used the Gensim[4] Python library for training the skip-gram model in rendering the word embeddings. The primary parameters are feature dimension size, word count minimum, and window size.

The size of the word vector is determined by the feature dimension parameter. The contextual information in word embeddings captures semantic and syntactic word properties with the feature dimension size being a critical parameter. In the research literature, the feature dimension size is typically in the 300 to 1000 range and appears to be on corpora that have relatively equal classes. In general, finding the ideal feature dimension size for a particular dataset is done experimentally. For the medical corpus utilized in this research study, a low feature dimension size only achieved good classification accuracy on the majority class. It appears that a lower dimension for the word vectors did not identify

[4]https://radimrehurek.com/gensim/models/word2vec.html

the subtle distinctions that differentiated the minority class records from the majority class records. During experimentation, the feature dimensions were 500, 1000, 1500, 2000, and 2500.

The word count minimum parameter is the threshold for adding words to the vocabulary. For this study, the minimum word counts were 15, 30, and 40. The window size is the number of words to the left and right of the vocabulary word in each narrative. This parameter is instrumental for determining the number of surrounding words to take into consideration, this is the context for the word. During experimentation, the window sizes were 5, 10, and 15.

The final parameters were a feature dimension size of 2000, context window size of 10 and word count minimum of 30 were used in rendering the word embeddings. The best results were achieved using these aforementioned values from preliminary experiments. The hierarchical softmax approach was also used with the skip-gram model. This section corresponds to Section 3.3 in Figure 2.

3.4. Generate (psuedo) document embeddings

The word embeddings generated from Section 3.3 can be easily used for determining semantic similarity as stated in Section 1, but can not be directly applied to document classification due to the inherent variability in the documents. Using a simple pooling method based on averaging the word vectors to represent the document has done well in document classification [59]. We refer to the approach for averaging word vectors to represent the document as *(pseudo) document embeddings*, since it's descriptive as well as succinct. To arrive at the pseudo document embeddings for the clinical narratives, the word vectors (generated from the skip-gram model) associated with the vocabulary words matching the words in the clinical narrative were averaged. That is, the vocabulary words matching words from each clinical narrative were added together then divided by the total words from the matching clinical narrative words in the vocabulary. A total of 2,208 (pseudo) document embeddings were used for the training set and 736 utilized for the testing set. This section corresponds to Section 3.4 in Figure 2.

3.5. Generate discriminant (psuedo) document embeddings

The output matrix from Section 3.4 consists of 2,000 features with 2,208 training patterns which is far from the rule of thumb of having at least five times as many training records in comparison to features [38]. PCA and MDA are two popular methods for dimensionality reduction. From our previous work it was discovered that using discriminants over principal components performed better using the same dataset in this study [60]. As mentioned in Section 1 and expanded on in Section 2, MDA is a supervised method for dimensionality reduction. The nonsingularity of the matrix rendered from this clinical corpus in Section 3.4 required a pseudoinverse and the Moore-Penrose pseudoinverse [61] was used before Equation (8) could be computed from Section 2.4. A reduced feature space consisting of four features was achieved after the application of MDA to the embeddings. As stated in Section 2.4, MDA results in $c-1$ feature projections. The embeddings resulted in a matrix of p by n or 2000 by 2208 with p being the number of features and n being the training instances. Similarly for the test set but with a p by n matrix of 2000 by 736. Applying MDA results in a 4 by 2208 matrix for training and a 4 by 736 matrix for testing. This section corresponds to Section 3.5 in Figure 2.

3.6. Generate document embeddings

This study used the Gensim[5] Python library for training the PV-DBOW model in generating the document embeddings. In Section 3.4, averaging the word vectors to arrive at a fixed representation size for each document was due to the variability of the documents. PV-DBOW resolves this problem by generating document vectors directly, which can be used easily for document classification.

The primary adjustable parameters are feature dimension size, word count minimum, and window size. The description of these parameters are described in Section 3.3. The feature dimensions used during experimentation were 300,

[5] https://radimrehurek.com/gensim/models/doc2vec.html

500, and 1000. The window sizes were 5, 10, and 15. The word count minimum parameter is the threshold for words added to the vocabulary. The minimum word counts for experimentation were 10 and 20.

The final parameters selected for generating the document embeddings: feature dimension size of 500, context window size and word count minimum of 10. This section corresponds to Section 3.6 in Figure 2. Experimentation was also done using paragraph vectors-distributed memory (PV-DM), the other model in Doc2Vec. The PV-DM model generates word vectors along with document vectors [62]. The PV-DBOW model performed much better than the PV-DM model on the dataset that was used in this paper.

3.7. Generate discriminant document embeddings

The document embeddings from Section 3.6 have a feature dimension size of 500. The application of MDA to the document embeddings resulted in a reduced feature space of four features, just as with Section 3.5. As stated in Section 2.4, MDA results in $c - 1$ feature projections. The embeddings from 3.6 resulted in a matrix of of p by n or 500 by 2208 with p being the number of features and n being the training instances. Similarly for the test set but with a p by n matrix of 500 by 736. Applying MDA results in a 4 by 2208 matrix for training and a 4 by 736 matrix for testing. This section corresponds to Section 3.7 in Figure 2.

3.8. Combine discriminant document embeddings (partial)

The document embeddings from Section 3.5 and 3.6 were concatenated to form a combined feature set with 504 feature dimensions. MDA had been applied to the (pseudo) document embeddings in Section 3.5, but not the document embeddings in Section 3.6. ELM classification was implemented on the combined feature set, the results are presented in Table 5. This section corresponds to Section 3.8 in Figure 2.

3.9. Combine discriminant document embeddings (full)

The document embeddings from Section 3.5 and 3.7 were concatenated to form a combined feature set with eight feature dimensions. MDA had been applied to the (pseudo) document embeddings in Section 3.5 and to the document embeddings in Section 3.7. ELM classification was implemented on the combined feature set, the results are presented in Table 5. This section corresponds to Section 3.9 in Figure 2.

4. Results and evaluation

Results on the classification task are evaluated using a standard machine learning evaluation measure, the F_1 score applied to each class:

$$F_1 = 2 \, . \, \frac{Precision \, . \, Recall}{Precision + Recall} \tag{19}$$

$$Precision = \frac{True\,Positive}{True\,Positive + False\,Positive} \tag{20}$$

$$Recall = \frac{True\,Positive}{True\,Positive + False\,Negative} \tag{21}$$

4.1. Baseline comparison

The methods that were mentioned in Section 1 provide a baseline for comparison. These include the bag of words (BoW) model, n-gram-based text categorization [16], latent semantic analysis (LSA) [20], and latent dirichlet allocation (LDA) [25]. The results in Table 4 show the BoW model and LSA perform better overall compared to the other methods.

4.2. Classification with ELM

ELM results in an overall improvement in classification compared to support vector machines (SVM) and a multilayer perceptron (MLP) for the classification of discriminant (pseudo) document embeddings from Section 3.5 [63]. The ELM

Table 4: The baseline using BoW, N-gram, LSA and LDA.

	BoW	N-gram	LSA	LDA	
Class	F1-Score	F1-Score	F1-Score	F1-Score	Test Ct.
C1	0.96	0.95	0.95	0.93	535
C2	0.51	0.18	0.44	0.04	45
C3	0.91	0.72	0.94	0.92	121
C4	0.63	0.48	0.59	0.48	19
C5	0.42	0.23	0.25	0.00	16

kernel[6] Matlab source code was utilized for this study. The key parameters for using the ELM kernel are the regularization coefficient and kernel parameter. For this research, a fixed regularization coefficient of 0.01 and kernel parameter values were 0.1, 1, 10, and 100. ELM classification was done on (pseudo) document embeddings from Section 3.5, document embeddings from Section 3.6, discriminant document embeddings (partial) from Section 3.8, and discriminant document embeddings (full) from Section 3.9. Classification was executed for 20 trials with the mean and standard deviation reported in Table 5.

In Table 5 for Class 2 (C2), the document embeddings (DE) from Section 3.6 used 500 features with no MDA applied has a mean F_1 Score of 0.59 and the discriminant document embeddings (DDE partial) from Section 3.8 using 504 features has a mean F_1 Score of 0.80. For C2, there is a 21% improvement in accuracy with DDE partial in comparison to DE. To reiterate, the feature set for discriminant document embeddings (partial) are just the four features provided by discriminant (psuedo) document embeddings in Section 3.5 combined with the document embeddings from Section 3.6.

Table 5 shows an overall improvement for both the discriminant document embeddings (partial and full) compared to the document embeddings and discriminant (pseudo) document embeddings, especially for the minority classes.

[6] `http://www.ntu.edu.sg/home/egbhuang/elm_kernel.html`

Table 5: The comparison of ELM classification on document embeddings (DE) with no MDA applied, discriminant (pseudo) document embeddings (DPDE), discriminant document embeddings (DDE) partial and (DDE) full.

	DE (No MDA)	DPDE	DDE (partial)	DDE (full)	
	F1-Score	F1-Score	F1-Score	F1-Score	Test
Class	$\mu\pm\sigma$	$\mu\pm\sigma$	$\mu\pm\sigma$	$\mu\pm\sigma$	Ct.
C1	0.97±0.0024	0.97±0.003	0.98±0.0018	0.97±0.0008	535
C2	0.59±0.0395	0.77±0.041	0.80±0.0196	0.77±0.0116	45
C3	0.95±0.0054	0.92±0.015	0.95±0.0028	0.95±0.0029	121
C4	0.71±0.0286	0.70±0.031	0.73±0.0331	0.77±0.0284	19
C5	0.61±0.0405	0.61±0.059	0.63±0.0364	0.68±0.0106	16

There is also overall less variability in the mean F_1 Score for the discriminant document embeddings (partial and full) as noted from the standard deviation for the 20 trials.

The results presented in Table 5 provide an overall summary but Figure 3 provides more detail into the F_1 Score for each one of the 20 trials for all four methods reported in Table 5. Figure 3a-e represents each of the five classes. Figure 3a for Class 1 shows the discriminant document embeddings (full) as being more stable in terms of less variability per trial. This stability throughout the trials for the discriminant document embeddings (full) is also illustrated in Figure 3b, Figure 3c and Figure 3e. However, in Figure 3, the discriminant document embeddings varies as much as the other methods specifically for trials 4-7 and trials 16-20.

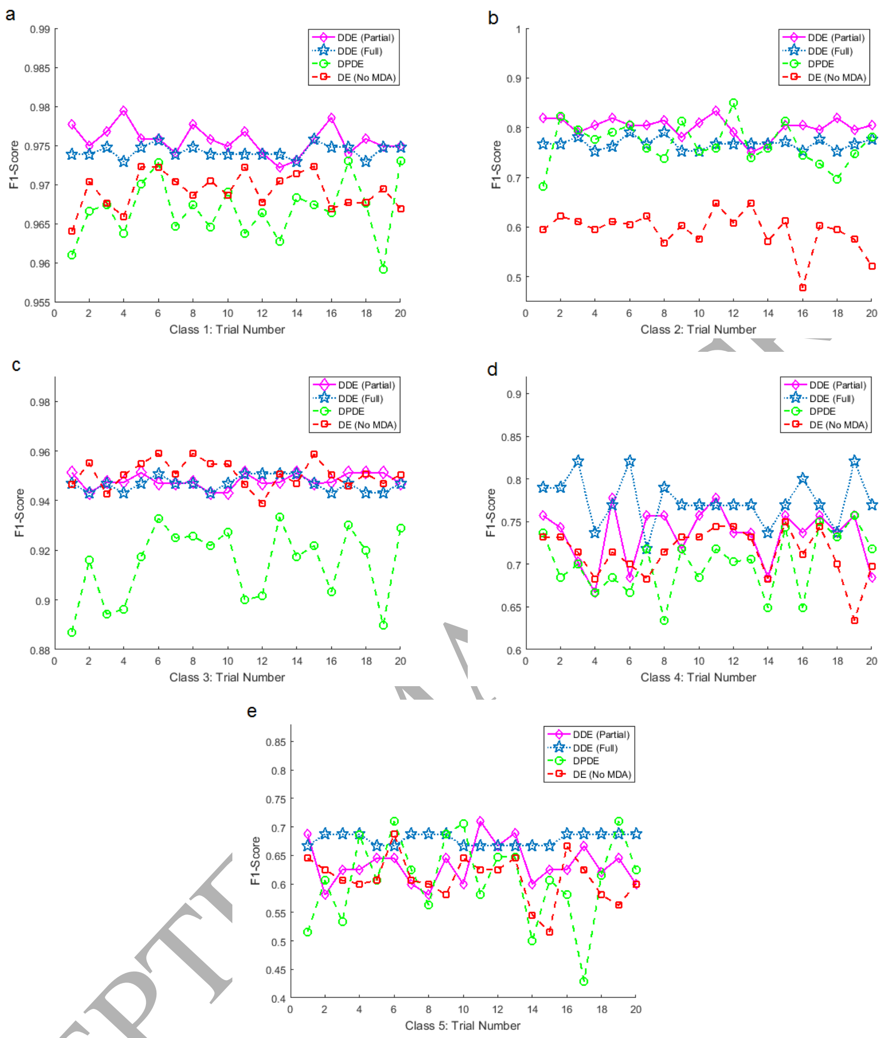


Figure 3: The F_1 Score for each trial for the methods reported in Table 5 on the five classes (a) Class 1 (b) Class 2 (c) Class 3 (d) Class 4 (e) Class 5.

5. Discussion and future work

The proposed method uses both unsupervised learning with skip-gram and PV-DBOW and supervised learning with MDA. However, there is a limitation with MDA, which restricts that the reduced number of dimensions be less than the number of classes in the dataset. A recent dimensionality reduction method referred to as semi-random projection (SRP) in combination with ELM has the discriminitive power of MDA without the reduced dimension space restriction [64]. In addition SRP works well with high-dimensional data which could be beneficial in the case of a larger corpus.

In this research the issue of class imbalance was not directly addressed. The slight boost with regard to the minority classes resulted indirectly from the combinational approaches of Section 3.8 and Section 3.9. Undersampling and oversampling methods are traditional approaches for addressing class imbalance. Essentially, the former method decreases the majority classes and the latter approach increases the minority classes. A recent weighted tanimoto ELM (T-WELM) addresses imbalance using a weighted approach based on the tanimoto coefficient, which may work well with imbalanced text data [65]. Since the tanimoto coefficient compares binary vectors, T-WELM could work directly with the one-hot representation. A one-hot representation is where each word in a vocabulary is represented as a binary vector with the length of the vector corresponding to the size of the vocabulary. The SRP and T-WELM methods are just a few of the recent trends in computational intelligence using ELM [66, 67].

The dataset for this study also contained imbalance at the sentence, word, and character level as described in Section 3.1 and Section 3.2. It's unclear if this type of imbalance necessitated the different feature dimensions for skip-gram and PV-DBOW in the first place. The results in Section 4 for the discriminant document embeddings looks promising but further study using different datasets is warranted to provide a conclusive answer. Further study is also needed to determine if the proposed method across various datasets improves accuracy with classification on a reduced feature space. The use of both medical and non-medical datasets would be ideal. For this study, the proposed approach permitted different feature dimensions for each method and the application of MDA appeared to provide an equalizing effect. That is, the reduced feature space for both methods are equal for the discriminant document embeddings in

Section 3.9, because of the $c-1$ property of MDA.

This research study used ELM for its classification capability but there is also a feature learning capability using ELM that could be used for directly generating the embeddings. An ELM auto-encoder (ELM-AE) is considered a unique implementation of ELM where the input matches the output and the objective of ELM-AE is to map the input features into a compressed, sparse or equal dimensional space [68]. The compressed representation of ELM-AE would be ideal for rendering distributed representations or embeddings for text.

6. Conclusion

In conclusion, this research has demonstrated that combining both skip-gram and PV-DBOW with MDA for rendering discriminant document embeddings with ELM classification provides an improvement especially for the minority classes on the dataset. Further research is needed to address the issues discussed in Section 5. These include experimentation using a variety of datasets, the inclusion of recent ELM methods that address imbalance and resolve the MDA reduced dimension space restriction. Also, considering the speed and generalization capability of ELM, exploring the feature learning aspect of ELM could expedite the generation of embeddings immensely.

7. Acknowledgements

This research was partially supported by the National Science Foundation (NSF) East Asia and Pacific Summer Institute (EAPSI) Fellowship Program under grant no. NSF-1614024.

References

[1] A. L. Benin, G. Vitkauskas, E. Thornquist, E. D. Shapiro, J. Concato, M. Aslan, H. M. Krumholz, Validity of using an electronic medical record for assessing quality of care in an outpatient setting, Medical Care 43 (7) (2005) 691–698.

[2] M. Z. Hydari, R. Telang, W. M. Marella, Electronic health records and patient safety, Communications of the ACM 58 (11) (2015) 30–32.

[3] T. Borlawsky, C. Friedman, Y. A. Lussier, Generating executable knowledge for evidence-based medicine using natural language and semantic processing, AMIA Annual Symposium proceedings (2006) 56–60.

[4] J. Mayer, T. Greene, J. Howell, J. Ying, M. A. Rubin, W. E. Trick, M. H. Samore, Agreement in classifying bloodstream infections among multiple reviewers conducting surveillance, Clinical Infectious Diseases 55 (3) (2012) 364–70.

[5] B. de Bruijn, S. Carini, S. Kiritchenko, J. Martin, I. Sim, Automated information extraction of key trial design elements from clinical trial publications, AMIA Annual Symposium proceedings (2008) 141–5.

[6] S. Meystre, P. J. Haug, Natural language processing to extract medical problems from electronic clinical documents: performance evaluation, Journal of Biomedical Informatics 39 (6) (2006) 589–99.

[7] S. T. Rosenbloom, J. C. Denny, H. Xu, N. Lorenzi, W. W. Stead, K. B. Johnson, Data from clinical notes: a perspective on the tension between structure and flexible documentation, Journal of the American Medical Informatics Association 18 (2) (2011) 181–6.

[8] D. Mladeni, J. Brank, M. Grobelnik, G. I. Webb, Document Classification, Springer US, Boston, MA, 2010, pp. 289–293.

[9] B. Pang, L. Lee, Opinion mining and sentiment analysis, Trends Informational Retrieval 2 (1-2) (2008) 1–135.

[10] T. Baldwin, M. Lui, Language identification: The long and the short of the matter, in: Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Assoc. for Comp. Linguistics, 2010, pp. 229–237.

[11] P. Petrenz, B. Webber, Stable classification of text genres, Computational Linguistics 37 (2) (2011) 385–393.

[12] C. D. Manning, H. Schütze, Foundations of statistical natural language processing, MIT press, 1999.

[13] R. Bouslimi, A. Messaoudi, J. Akaichi, Using a bag of words for automatic medical image annotation with a latent semantic, Int'l Journal Artificial Intelligent Applications, 2013 4(3).

[14] D. Dinh, L. Tamine, Towards a context sensitive approach to searching information based on domain specific knowledge sources, Web Semantics: Science, Services and Agents on the World Wide Web 12 (2012) 41–52.

[15] K. R. Bayyapu, P. Dolog, Tag and neighbor based recommender systems for medical events, Proceedings of the 1st int'l Workshop on Web Science and Information Exchange in the Medical Web.

[16] W. B. Cavnar, J. M. Trenkle, et al., N-gram-based text categorization, Ann Arbor MI 48113 (2) (1994) 161–175.

[17] Y. Zhou, P. K. Amundson, F. Yu, M. M. Kessler, T. L. Benzinger, F. J. Wippold, Automated classification of radiology reports to facilitate retrospective study in radiology, Journal of digital imaging 27 (6) (2014) 730–736.

[18] M. Chary, N. Genes, A. McKenzie, A. F. Manini, Leveraging social networks for toxicovigilance, Journal of Medical Toxicology 9 (2) (2013) 184–191.

[19] Y. Luo, A. R. Sohani, E. P. Hochberg, P. Szolovits, Automatic lymphoma classification with sentence subgraph mining from pathology reports, Journal of the American Medical Informatics Association 21 (5) (2014) 824–832.

[20] T. K. Landauer, P. W. Foltz, D. Laham, An Introduction to Latent Semantic Analysis, Discourse Processes 25 (1998) 259–284.

[21] P. D. Turney, P. Pantel, From frequency to meaning: vector space models of semantics, Journal of Artificial Intelligence Research 37 (2010) 141–188.

[22] A. S. Cohen, B. Elvevåg, Automated computerized analysis of speech in psychiatric disorders, Current opinion in psychiatry 27 (3) (2014) 203.

[23] S. C. Mihai, M. Corneliu, Thematic correlations of the patients with severe depressive episode. a case study, Procedia-Social and Behavioral Sciences 187 (2015) 163–167.

[24] W. Kintsch, The potential of latent semantic analysis for machine grading of clinical case summaries, Journal of biomedical informatics 35 (1) (2002) 3–7.

[25] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent dirichlet allocation, The Journal of Machine Learning Research 3 (2003) 993–1022.

[26] K. R. Chan, X. Lou, T. Karaletsos, C. Crosbie, S. Gardos, D. Artz, G. Ratsch, An empirical analysis of topic modeling for mining cancer clinical notes, 2013 IEEE 13th Int'l Conference on Data Mining Workshops.

[27] I. Korkontzelos, T. Mu, A. Restificar, S. Ananiadou, Text mining for efficient search and assisted creation of clinical trials, in: Proc. of the 5th int'l workshop on data and text mining in biomedical informatics, ACM, 2011, pp. 43–50.

[28] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, Journal of Machine Learning Research 3 (2003) 1137–1155.

[29] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th Int'l Conference on Machine learning, ACM, 2008, pp. 160–167.

[30] A. Mnih, G. Hinton, Three new graphical models for statistical language modelling, in: Proceedings of the 24th int'l conference on Machine learning, ACM, 2007, pp. 641–648.

[31] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, ICLR Workshop, 2013.

[32] E. Choi, A. Schuetz, W. F. Stewart, J. Sun, Medical concept representation learning from electronic health records and its application on heart failure prediction, arXiv preprint arXiv:1602.03686, 2016.

[33] Q. V. Le, T. Mikolov, Distributed representations of sentences and documents, In Proceedings of ICML, 2014.

[34] P. Xie, Y. Deng, E. Xing, Diversifying restricted boltzmann machine for document modeling, in: Proceedings of the 21th ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1315–1324.

[35] A. M. Dai, C. Olah, Q. V. Le, Document embedding with paragraph vectors, arXiv preprint arXiv:1507.07998, 2015.

[36] T. Pang-Ning, M. Steinbach, V. Kumar, Introduction to data mining, Library of Congress, 2006 74.

[37] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer New York, New York, NY, 2009, Ch. 18, pp. 649–698.

[38] T. Hastie, R. Tibshirani, Expression arrays and the p >>n problem, Technical Report, 2003.

[39] K. P. Murphy, Machine learning: a probabilistic perspective, MIT Press, 2012.

[40] R. O. Duda, P. E. Hart, D. G. Stork, Pattern classification, John Wiley & Sons, 2012.

[41] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Neural Networks, 2004.

Proceedings. 2004 IEEE International Joint Conference on, Vol. 2, IEEE, 2004, pp. 985–990.

[42] G. Huang, G.-B. Huang, S. Song, K. You, Trends in extreme learning machines: a review, Neural Networks 61 (2015) 32–48.

[43] Y. Liu, K.-T. Chuang, F.-W. Liang, H.-J. Su, C. M. Stultz, J. V. Guttag, Transferring knowledge from text to predict disease onset, arXiv preprint arXiv:1608.02071, 2016.

[44] L. Zeng, Z. Li, Text classification based on paragraph distributed representation and extreme learning machine, in: Int'l Conference in Swarm Intelligence, Springer, 2015, pp. 81–88.

[45] S. S. Vempala, The random projection method, Vol. 65, American Mathematical Society, 2005.

[46] P. Gastaldo, R. Zunino, E. Cambria, S. Decherchi, Combining elm with random projections, IEEE intelligent systems 28 (6) (2013) 46–48.

[47] W. Zheng, Y. Qian, H. Lu, Text categorization based on regularization extreme learning machine, Neural Computing and Applications 22 (3) (2013) 447–456.

[48] A. Mnih, G. E. Hinton, A scalable hierarchical distributed language model, in: Advances in neural information processing systems, 2009, pp. 1081–1088.

[49] T. Mikolov, A. Deoras, D. Povey, L. Burget, J. Černockỳ, Strategies for training large scale neural network language models, in: Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on, IEEE, 2011, pp. 196–201.

[50] T. H. Cormen, Introduction to algorithms, MIT press, 2009.

[51] F. Morin, Y. Bengio, Hierarchical probabilistic neural network language model, in: Aistats, Vol. 5, Citeseer, 2005, pp. 246–252.

[52] L. Bottou, Online learning and stochastic approximations, On-line learning in neural networks 17 (9) (1998) 142.

[53] C. M. Bishop, Pattern recognition and machine learning, Springer, 2006.

[54] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.

[55] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42 (2) (2012) 513–529.

[56] G.-B. Huang, L. Chen, C. K. Siew, et al., Universal approximation using incremental constructive feedforward networks with random hidden nodes, IEEE Transactions on Neural Networks 17 (4) (2006) 879–892.

[57] J. A. Miñarro-Giménez, O. Marín-Alonso, M. Samwald, Applying deep learning techniques on medical corpora from the world wide web: a prototypical system and evaluation, CoRR abs/1502.03682, 2015.

[58] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, IJCAI Proceedings of the 14th Int'l Joint Conference on Artificial Intelligence (1995) 1137–1143.

[59] R. Liu, D. Wang, C. Xing, Document classification based on word vectors, ISCSLP, 2014.

[60] P. Lauren, G. Qu, F. Zhang, Discriminant word embeddings on clinical narratives, in: 5th Workshop on Data Mining for Medicine and Healthcare, Florida, May 5-7, SIAM, 2016, pp. 74–84.

[61] A. Ben-Israel, T. N. Greville, Generalized inverses: theory and applications, Vol. 15, Springer Science & Business Media, 2003.

[62] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[63] P. Lauren, G. Qu, F. Zhang, A. Lendasse, Clinical narrative classification using discriminant word embeddings with elm, in: Int'l Joint Conference on Neural Networks, Vancouver, Canada, July 24-29, IEEE, 2016.

[64] R. Zhao, K. Mao, Semi-random projection for dimensionality reduction and extreme learning machine in high-dimensional space, IEEE Computational Intelligence Magazine 10 (3) (2015) 30–41.

[65] W. M. Czarnecki, Weighted tanimoto extreme learning machine with case study in drug discovery, IEEE Computational Intelligence Magazine 10 (3) (2015) 19–29.

[66] G.-B. Huang, E. Cambria, K.-A. Toh, B. Widrow, Z. Xu, New trends of learning in computational intelligence [guest editorial], IEEE Computational Intelligence Magazine 10 (2) (2015) 16–17.

[67] G.-B. Huang, E. Cambria, K.-A. Toh, B. Widrow, Z. Xu, New trends of learning in computational intelligence (part ii)[guest editorial], IEEE Computational Intelligence Magazine 10 (3) (2015) 8–8.

[68] L. L. C. Kasun, H. Zhou, G.-B. Huang, C. M. Vong, Representational learning with elms for big data, IEEE Intelligent Systems 28 (6) (2013) 31–34.

Paula Lauren is currently a Ph.D. Student in the Department of Computer Science and Engineering at Oakland University located in Rochester, Michigan. She received a bachelor of science in business administration with a major in information systems at Wayne State University. Paula received a master of science in computer science and information science from The University of Michigan-Dearborn. Her research interests include data mining and machine learning, natural language processing and healthcare computing. At Oakland University she is pursuing a Ph.D. in computer science and informatics..

Guangzhi Qu received his B.S. and M.S. degrees in both computer science and engineering from Beijing University of Aeronautics and Astronautics, Beijing, China in 1996 and 1999, respectively. He received a Ph.D. degree in Computer Engineering from The University of Arizona in 2005. Since then he was a research assistant professor in ECE department at The University of Arizona before he joined the Department of Computer Science and Engineering at Oakland University in 2007. He is now an associate professor at Oakland University. He served as the conference chair of the 13th International Conference on Machine Learning and Applications in 2014. His current research interests include data mining and machine learning, operating systems, and healthcare computing.

Feng Zhang received the B.S. degree from Beihang University, Beijing, China, in 1996, the M.S. degree and the Ph.D. degree from Sun Yat-set University, Guangzhou, China, in 2003 and 2008 respectively. All are in computer science. He worked at Kent State University as a visiting scholar from Feb. 2012 to Feb. 2013. Currently, he is an associate professor at China University of Geosciences, Wuhan, China. He is the author or coauthor of more than 30 scientific papers. His research interests include machine learning and privacy-preserving data processing.

Amaury Lendasse was born in 1972, in Belgium. He received a M.S. degree in Mechanical Engineering from the Universite Catholique de Louvain (Belgium) in 1996, a M.S. in Control in 1997 and a Ph.D. in Applied Mathematics in 2003 from the same university. In 2003, he was a postdoctoral researcher in the Computational Neurodynamics Lab at the University of Memphis. From 2004 to 2014, he was a senior researcher and an Adjunct Professor in the Adaptive Informatics Research Centre in the Aalto University School of Science (better known as the Helsinki University of Technology) in Finland. He has created and lead the Environmental and Industrial Machine Learning at Aalto. He is now an Associate Professor at The University of Iowa (USA) and a visiting Professor at Arcada University of Applied Sciences in Finland. He was the Chairman of the annual ESTSP conference (European Symposium on Time Series Prediction) and member of the editorial board and program committee of several journals and conferences on machine learning. He is the author or coauthor of more than 200 scientific papers in international journals, books or communications to conferences with reviewing committee. His research includes Big Data, time series prediction, chemometrics, variable selection, noise variance estimation, determination of missing values in temporal databases, nonlinear approximation in financial problems, functional neural networks and classification.