Int. J. Nonlinear Anal. Appl. 12 (2021) No. 2, 2161-2178
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/IJNAA.2021.5363

Reducing explicit word vectors dimensions using BPSO-based labeling algorithm and voting method

Atefe Pakzad[a], Morteza Analoui[a,*]

[a]School of Computer Engineering, Iran University of Science and Technology,Tehran, Iran

(Communicated by Madjid Eshaghi Gordji)

Abstract

Interpretability of word vector components is very important for obtaining conceptual relations. Word vectors derived from counting models are interpretable but suffer from the high-dimensionality problem. Our goal in this study is to obtain interpretable low-dimensional word vectors in such a way that the least accuracy loss occurs. To achieve this goal, we propose an approach to reduce the dimensions of word vectors using a labeling method based on the BPSO algorithm and a voting method for selecting final context words. In this approach, we define several different base models to solve the labeling problem using different data and different objective functions. Then, we train each base model and select 3 of the best solutions for each model. We create the target word vectors of the dictionary based on the context words labeled "1". Next, we use the three best solutions of each base model to build the ensemble. After creating the ensemble, we use the voting method to assign the final label to the primary context words and select N final context words. In this study, we use the corpus ukWaC to construct word vectors. We evaluate the resulting word vectors on the MEN, RG-65, and SimLex-999 test sets. The evaluation results show that by reducing the word vectors dimensions from 5000 to 1507, the Spearman correlation coefficient of the proposed approach has been reduced to a lesser extent compared to each base model. Therefore, the accuracy drop of the proposed approach is justified after reducing the dimensions from 5000 to 1507. It is not a large penalty because the resulting word vectors are low-dimensional and interpretable.

Keywords: Explicit word vectors, Voting method, Objective function, Binary particle swarm optimization, Labeling method
2010 MSC: 47N10,91B12

*Corresponding author
Email addresses: a_pakzad@comp.iust.ac.ir (Atefe Pakzad), analoui@iust.ac.ir (Morteza Analoui)

Received: February 2021 *Accepted:* August 2021

1. Introduction

Distributional semantic models (DSMs) are a set of methods that extract the meaning of words from large corpora. In DSMs, the meaning of a word in a sentence is represented by a vector automatically [4]. There are two models for constructing word vectors: count-based models and prediction-based models [5]. Count-based models usually consider the most frequent words in a corpus as context words. Then, they count the co-occurrence of the dictionary target words with the context words using a fixed window around the dictionary words [23]. Semantic word vectors obtained by counting-based models are usually high-dimensional and each dimension corresponds to a natural word, so word vectors are interpretable and meaningful. Dimensional reduction methods such as Non-negative Matrix Factorization (NMF) [14] and Singular Value Decomposition (SVD) [24] are usually used to reduce the vector's dimensions and implicit word vectors are generated. The resulting implicit word vector dimensions have no semantic equivalent.

Predictive models often use neural methods to obtain word vectors and generate dense implicit word vectors that the resulting dimensions are not meaningful [28]. Predictive models such as Word2Vec [25], Glove [29] and FastTexts [7] usually have high performance in NLP tasks. Many tasks such as machine translation [1], sentiment analysis [27, 18], question answering [12], text classification [33], and topic modeling [11] use distributional semantic vectors. The word vectors obtained by prediction models have good performance in these applications but cannot reflect the text information because the resulting word vectors are not interpretable. Explicit word vectors obtained by counting methods are meaningful, but the main drawback is the high dimensions of word vectors. Common dimension reduction methods produce implicit vectors and eliminate interpretability. In this study, for the first time, we intend to reduce the word vector's dimensions by using an optimization method, to keep the interpretability of word vectors. We identify effective context words to obtain word vectors using the BPSO-based labeling algorithm by two different objective functions that are applied on two different co-occurrence matrices on a corpus. Then, we report the best low-dimensional set of context words for producing semantic word vectors by the voting method.

We will briefly explain how to construct word vectors in counting-based methods.

- We usually select some of the most frequent words in the corpus as primary context words. In a co-occurrence matrix, each column corresponds to a context word, and each row corresponds to a dictionary word namely the target word.
- Consider a fixed window size (W) and count the co-occurrence of the target word with the context words that are placed in the neighborhood of the target word ($\pm W$) in the sentence.
- The component C_{ij} in the co-occurrence matrix corresponds to the co-occurrence of the i_{th} target word and the j_{th} context word in the interval $\pm W$ in sentences.
- The i_{th} row of the co-occurrence matrix represents the target word vector which is corresponding to the i_{th} row.

Raw co-occurrence numbers in the co-occurrence matrix are biased. Usually, the Pointwise Mutual Information method is used to eliminate this bias. Pointwise mutual information reflects the correlation between two words in a corpus. A larger PMI association means more correlation between two words or frequent occurrences of two words in the corpus. Also, a smaller PMI association means that if the first word occurs in one sentence, we expect that the second word is not in the sentence. So a larger PMI means more semantic relevance of the two words and vice versa. We obtain the PMI measure for the target word t and the context word c based on the following Equation [21]:

$$PMI(t,c) = log\frac{P(t,c)}{P(t)P(c)} \tag{1.1}$$

If the target word t and the context word c are independent, the PMI measure will be zero. Usually, a positive PMI or PPMI measure is used to eliminate co-occurrence matrix bias.

$$PPMI = \max(PMI, 0) \tag{1.2}$$

In this study, we propose a method for classifying primary context words into two classes, effective and not_effective, which assigns a label "1" or "0" to each context word using a BPSO-based labeling algorithm. Labels "1" and "0" correspond to effective context words and not_effective context words, respectively. To solve the classification problem using BPSO-based labeling algorithm, we define two different objective functions namely OF_1 and OF_2. The function OF_1 tries to maximize the Spearman correlation coefficient between the similarities of word pairs. The function OF_2 minimizes the sum of squares of the words vector's differences. Next, we get the optimal solutions obtained by objective functions OF_1 and OF_2 using different co-occurrence matrices, which have 1000 and 500 columns. Then, we use a voting method in ensemble to classify primary context words as effective and not_effective. The proposed method classifies N context words from 5K primary context words by the label "1". The evaluation results show that the vectors dimensions are reduced from 5000 to N=1500 by minimal loss of accuracy.

The following of this paper is structured as follows: Section 2 discusses related works in literature, while section 3 explains how the proposed approach reduce explicit word vectors dimension by finding final context words. We describe the evaluation results and discussions in detail in Section 4. Finally, the paper is concluded in Section 5.

2. Related Work

Distributional semantic models are divided into two category namely count-based and prediction-based models. The word vectors obtained by the count-based model are explicit. Each dimension of a word vector corresponds to a lexical word. Prediction-based models generate implicit word vectors. That is, each dimension of the word vector is just a real number and does not refer to a specific concept. Predictive models are usually very accurate. Accuracy in count-based models is usually less but, the word vectors are meaningful. Explicit word vectors are valuable because a lot of information can be extracted from each word vector. We can obtain information such as word senses and words relatedness in different fields using explicit word vectors.

Despite the remarkable success and widespread acceptance of word embedding models, there is still a drawback: the inability to provide a conceptual equivalent of word embedding dimensions. In implicit word vectors, the meaning of individual dimensions is unattainable. This issue is a problem in general and in sensitive fields like medicine in particular. Because using uninterpretable word vectors, the conceptual relationship between different diseases and drugs cannot be accurately identified. For example, in the medical concepts of Insulin and Diabetes mellitus, word embedding models can achieve semantic similarity between two words. But it is not possible to determine how much the concept of pharmacological substance or hormone is related to insulin. Conceptual relations can be easily extracted by having interpretable word vectors. Interpretable representations can provide explanatory answers to questions about conceptual relatedness [19].

Attempts have been made in the literature to transform word embeddings into interpretable representations. Reference [34] uses sparseness and non-negativity by changing the loss function in Glove to create interpretable word vectors. Reference [26] uses non-negative matrix factorization

(NMF) on the co-occurrence matrix to extract interpretable representations. Article [35] uses a k-sparse denoising autoencoder to construct a sparse high-dimensional non-negative mapping for word embeddings called SParse Interpretable Neural Embeddings (SPINE). Reference [15] suggests Sparse Overcomplete Word Vectors for solving the optimization problem and produces a high-dimensional sparse non-negative mapping of implicit word vectors. The main idea of these researches for creating an interpretable representation is to create sparseness in the word embedding dimensions. They turn low-dimensional word embeddings into the sparse non-negative high-dimensional mapping of word vectors. Then, a semantic equivalent is assigned to each word vector dimension. These methods are not able to extract low-dimensional interpretable word vectors directly from the corpus.

The main disadvantage of semantic word vectors resulting from count-based methods is high dimensionality. Dimensional reduction methods such as NMF, SVD, and PCA are usually used to reduce the dimensions of explicit word vectors, which cause the vectors to lose their interpretability. So, the word vectors are transferred to a new implicit vector space with low dimensionality.

Attempts have been made to reduce the dimensions of explicit word vectors. Reference [8] believes that it is not crucial to consider all non-existent relationships which are zero in the co-occurrence matrix. The word pair relations are only valuable if they are non-zero in vector representation. References [9] and [30] use filtering methods to reduce the dimension of word vectors. They choose the most relevant context words for each target word. References [16] and [17] use filtering methods to extract r relevant context words based on the highest likelihood score. In the literature, not much attention has been paid to reducing the dimensions of explicit word vectors that maintain the interpretability of word vectors. The above research identifies important context words for each target word in the dictionary. The filtering methods do not explicitly suggest the final context words for each corpus. As a result, it is not possible to construct the target word vectors in the dictionary based on final context words.

Particle swarm optimization is an efficient evolutionary optimization algorithm introduced by Kennedy and Eberhart in 1995 [22]. The PSO algorithm has recently been used to select important data features in many tasks. Reference [32] suggests a PSO-based multi-objective method to select features. First, the feature selection method displays the features using a graph representation model. Then, the feature centralities are calculated for all graph nodes. Finally, a PSO-based search process is run to select important features. The authors of [31] introduce an online feature selection method based on the multi-objective PSO algorithm for multi-label classification. Reference [10] provides an efficient feature selection algorithm based on BPSO and CS-BPSO methods. The algorithm improves the efficiency of Arabic email authorship analysis. Article [2] introduces a multi-objective PSO-based method that rates based on the frequency of features. Then, it uses these ratings to select remarkable features. In this study, we propose a BPSO-based labeling method to find the final context words. First, we assign labels "0" and "1" to the most frequent context words in the corpus using the proposed labeling optimization algorithm. The label "1" means the context word is effective, and the label "0" means the context word is not_effective. We perform context word labeling operation using two different objective functions OF_1 and OF_2, on two different co-occurrence matrices, namely SD and DD. We solve the optimization problem with the constraints N= 500 and N= 1000, namely that the number of context words by label "1" should be 500 and 1000, respectively. Then, we use a voting method in ensemble to select the final context words from all the different base models in which the labeling problem is solved. Then, we generate and evaluate the co-occurrence matrix of the target words in the dictionary using the final context words obtained by the voting method in ensemble. The evaluation results report that using ensemble and voting method in comparison to each base model improves the Spearman correlation coefficient of word vectors on test sets. The results are presented in detail in Section 4.2.

3. Proposed Approach

Our goal in this study is to select the final context words to reduce the dimensions of explicit word vectors. So, the interpretability of word vector components is maintained. To achieve this goal, we propose a method based on the BPSO algorithm and the voting method in an ensemble to select the final context words. The proposed method consists of two phases. In the first phase, by the BPSO-based labeling method, we label the primary context words in different modes using different objective functions and different co-occurrence data. In the second phase, we get the best label for each primary context word using the voting method in an ensemble. In Section 3.1, we briefly describe the particle swarm optimization method. In Section 3.2, we introduce the context word labeling method, which is based on the BPSO algorithm. The labeling method uses two proposed objective functions to solve the problem. We utilize the voting method in the ensemble for selecting the final context words in section 3.3.

3.1. Particle Swarm Optimization

The PSO algorithm is a population-based stochastic search algorithm inspired by the swarm behavior of some species, such as flocks of birds and schools of fish. Kennedy and Eberhart introduced the PSO algorithm in 1995 [22]. A swarm consists of a collection of particles. Each particle is a member of the swarm population. A particle is a candidate solution for the desired problem. The PSO algorithm uses a multi-dimensional search to find the best solution. Each particle flies in a multi-dimensional search space. A particle uses the best position discovered by itself and discovered by its neighbors (other particles) to move towards the optimal solution [10, 2, 3]. Each particle has position and velocity parameters. The position and velocity of the i_{th} particle are denoted by x_i and v_i, respectively. The movement of each particle is based on its best position (pbest) and the best solution in the swarm (gbest), namely a particle with the best pbest [10]. To solve the feature selection problem using the PSO algorithm, we consider a multi-dimensional vector for the position of each particle. Each dimension of the particle position corresponds to a feature [31]. Suppose the search space has D dimensions and there are m particles in the swarm. The i_{th} particle has a position $x_i = [x_{i1}, x_{i2}, \ldots, x_{iD}]$ with velocity $v_i = [v_{i1}, v_{i2}, \ldots, v_{iD}]$ where $i = 1, 2, \ldots, m$. In the PSO algorithm, each particle moves to its best position (pbest) and best position of swarm (gbest), which is $pbest_i = [pbest_{i1}, pbest_{i2}, \ldots, pbest_{iD}]$ and $gbest_i = [gbest_{i1}, gbest_{i2}, \ldots, gbest_{iD}]$. The i_{th} particle updates its position based on its velocity (v_i), which is randomly generated based on the pbest and gbest positions. For each dimension h of the i_{th} particle, the velocity v_{ih} and the position x_{ih} are calculated by Equations 3.1 and 3.2.

$$v_{ih}^t = wv_{ih}^{t-1} + c_1 b_1 (pbest_{ih}^{t-1} - x_{ih}^{t-1}) + c_2 b_2 (gbest_h^{t-1} - x_{ih}^{t-1}) \tag{3.1}$$

$$x_{ih}^t = x_{ih}^{t-1} + v_{ih}^t \tag{3.2}$$

In Equations 3.1 and 3.2, t is the number of iterations (number of generations). Inertia weight is used to control the speed and balance of the algorithm's exploration and exploitation capabilities. Large w maintains the high velocity of particles and prevents particles from being caught in the local optimum. A small w causes the particles to take advantage of their current search location. The constants c_1 and c_2 in Equations 3.1 and 3.2 are the acceleration coefficients. These constants determine the degree to which particles tend to be closer to the pbest and gbest positions. Fixed numbers b_1 and b_2 are random numbers with uniform distribution in the range 0 and 1. The condition for termination of the PSO algorithm is the achievement of maximum iterations (number of generations), a specific value of pbest, or no improvement in pbest [2]. The particle efficiency is measured

by the degree of proximity of the particle to the local optimal. The proximity is calculated based on the predefined objective function for the problem [13]. The first step to solving the optimization problem is to select an appropriate objective function.

First, the particle swarm optimization algorithm was used for continuous problems. Later, it was developed for discrete problems, known as binary particle swarm optimization [36]. In the BPSO algorithm, discrete values 0 and 1 are assigned to the position of the i_{th} particle based on the velocity v_i. The sigmoid function is applied based on the following equations:

$$x_{ih}^t = \begin{cases} 1, & \text{if } rand(0,1) < S(v_{ih}^t) \\ 0, & \text{else} \end{cases} \tag{3.3}$$

$$S(v_{ih}^t) = \frac{1}{1 + e^{-v_{ih}^t}} \tag{3.4}$$

The function rand() specifies a random number between zero and one. Parameter v_{ih}^t represents the new velocity of the particle at the moment t. When $S(v_{ih}^t)$ is more significant than random numbers, the position of the particle will be 1. Figure 1 shows a D-dimensional binary solution obtained by a particle [10].

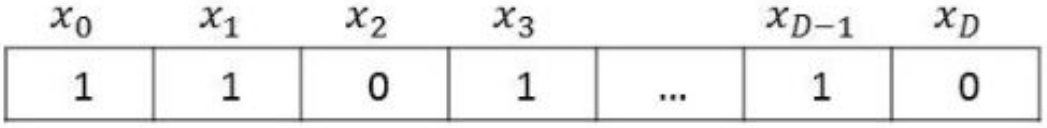

Figure 1: D-dimensional solution obtained by a particle.

3.2. Labeling context words using BPSO-based labeling algorithm

In this study, using the BPSO algorithm, we label the primary context words by "0" and "1". Label "1" means the context word is effective, and label "0" means the context word is not_effective. First, we consider the D most frequent words in the corpus (of the type of noun, verb, adverb, and adjective) as the primary context words. The target word vector in the dictionary is $\overrightarrow{W} = [w_1, w_2, \ldots, w_D]$. The set of labels assigned to the primary context words is denoted by $L = l_1, l_2, \ldots, l_D$. This study selects a limited number of N words from the primary context words as the final context words. Therefore, we propose a constraint to solve the BPSO-based labeling problem. That is, the number of words that have a binary label "1" is N.

$$\Sigma_{j=1}^{D} l_j = N \tag{3.5}$$

To solve the labeling problem, we create a training set by selecting m words from the dictionary. First, we create a co-occurrence matrix for the target words in the dictionary using the primary context words. Then, we extract the word vectors of the training set from the co-occurrence matrix and place them in the matrix T. The proposed method defines the label l_j for the j_{th} column of the matrix (columns correspond to the primary context words). The matrix T has m rows and D columns. Then, we multiply the binary label of the j_{th} column by all components of the j_{th} column of matrix T. As a result, each element of the matrix T is denoted by t, which corresponds to the cell in i_{th} row and the j_{th} column. Implicit word vectors obtained by prediction-based methods have high

efficiency. To get the objective function, we try to use the implicit vectors obtained by prediction-based methods. So the low-dimensional explicit word vectors obtained by the labeling method have the most similarity and the most negligible difference with the implicit vectors. For this reason, we propose to reduce the difference between low-dimensional explicit word vectors obtained by the labeling method and implicit word vectors in the objective function. In this research, we use implicit word vectors obtained by word2vec software. We get the implicit vectors of the training set words by 1000 dimensions and put them in the matrix W2V. The dimensions of the matrix W2V are $m \times 1000$. Each component of the matrix W2V is denoted by $wv_{i,j}$, corresponding to the i_{th} row and the j_{th} column. The smaller distance between the target word vectors obtained by the labeling method (matrix T) and the implicit vectors obtained by the Word2Vec method (matrix W2V) means that low-dimensional explicit word vectors attained by the labeling method are more efficient and accurate. Comparing the components of two matrices is not possible because the dimensions of the matrices T and W2V are not equal. For this reason, we obtain the following product matrices:

$$\mathcal{T} = T \times T' \tag{3.6}$$

$$\mathcal{W} = W2V \times W2V' \tag{3.7}$$

The matrices $\mathcal{T}$ and $\mathcal{W}$ have m rows and m columns. Each component of the matrix $\mathcal{T}$ is denoted by $\tau_{i,j}$, which corresponds to i_{th} row and j_{th} column. Also, each element of the matrix $\mathcal{W}$ is denoted by $\omega_{i,j}$, which corresponds to i_{th} row and j_{th} column. Each component $\tau_{i,j}$ of the matrix $\mathcal{T}$ is the inner product of the target word vector corresponding to the i_{th} row ($\overrightarrow{t}_i$) and the target word vector corresponding to the j_{th} row ($\overrightarrow{t}_j$) of the matrix T. Also, each component $\omega_{i,j}$ of the matrix $\mathcal{W}$ is the inner product of the target word vector corresponding to i_{th} row ($\overrightarrow{wv}_i$) and the target word vector corresponding to j_{th} row ($\overrightarrow{wv}_i$) of the matrix W2V.

$$\tau_{i,j} = \overrightarrow{t}_i . \overrightarrow{t}_j \tag{3.8}$$

$$\omega_{i,j} = \overrightarrow{wv}_i . \overrightarrow{wv}_j \tag{3.9}$$

Then, for solving the labeling problem, we try to make the word vectors of matrix T similar to implicit word vectors of matrix W2V. In this research, we propose two different objective functions to solve the labeling optimization problem. In Section 3.2.1, we describe an objective function based on the Spearman correlation coefficient. In Section 3.2.2, we explain an objective function based on the sum of squares of the word vector's differences.

3.2.1. Objective function OF_1

One way to evaluate the similarity of word pairs is to use the Spearman correlation coefficient. If the Spearman correlation coefficient between the similarity of the word pairs of the matrices W2V and T is more remarkable, we get better and more accurate labels. We define an objective function that maximizes the Spearman correlation coefficient between the similarity of word pairs of matrices W2V and T. The steps for obtaining cosine similarity of word pairs in matrices W2V and T are as follows:

1. The inner product of the word pairs in the matrices W2V and T is found in matrices $\mathcal{W}$ and $\mathcal{T}$, respectively. The matrices $\mathcal{W}$ and $\mathcal{T}$ are $m \times m$.

2. Calculate $o = \frac{m^2-m}{2} + m$.

3. $VT = [vt_1, vt_2, \ldots, vt_O]$ and $WT = [wt_1, wt_2, \ldots, wt_O]$.
4. for $i = 1 \ldots m\{$
   a. For $j = i \ldots m\{$
      (i) $vt_z = \frac{\tau_{i,j}}{\sqrt{\tau_{i,i} \times \tau_{j,j}}}$
      (ii) $wt_z = \frac{\omega_{i,j}}{\sqrt{\omega_{i,i} \times \omega_{j,j}}}$
      (iii) $z = z + 1$
   b. $\}$
5. $\}$

To calculate the Spearman correlation coefficient, the word pair's similarities in matrices W2V and T should be written in vectors WT and VT, respectively. The word pairs similarity matrix is symmetric, so the number of components needed to write the word pairs similarities on a vector is $o = \frac{m^2-m}{2} + m$. In step 3, we construct the vectors VT and WT, which have O columns. In step 4-a-i, we calculate the cosine similarity of the word pair in the i_{th} row and j_{th} column of the matrix t and put it in vt_z. Also, in step 4-a-ii, we get the cosine similarity of the word pair in the i_{th} row and j_{th} column in the matrix W2V and put it in wt_z. Then, we define the objective function based on the Spearman correlation coefficient of vectors WT and VT as follows:

$$OF_1 = SpearmanCorrelation(WT, VT) \tag{3.10}$$

The labeling method labels the primary context words (l_j) in such a way that maximizes the objective function OF_1.

3.2.2. Objective function OF_2

In this study, we propose to reduce the distance between the target word vectors obtained by final context words ($\overrightarrow{t}_i$) and the implicit target word vectors obtained by Word2Vec software ($\overrightarrow{wv}_i$) to construct low-dimensional explicit word vectors accurately. Therefore, we define the objective function of the problem based on the sum of squares of the word vector's differences as follows:

$$OF_2 = \Sigma_{i=1}^{f} \Sigma_{j=1}^{f} (\omega_{i,j} - \tau_{i,j})^2 \tag{3.11}$$

To solve the BPSO-based labeling problem, context word labels (l_j) must be selected in a way that minimizes the objective function of Equation 3.11.

3.2.3. Labeling algorithm

In the labeling method based on the BPSO algorithm, NP is the number of population. Each member of the population is a particle. The position of each particle is a D-dimensional vector. The components of the particle position vector correspond to the labels of the primary context words. The best particle is determined after running the following optimization algorithm:

1. The population size is NP.
2. Initialize the population (position and velocity of each particle) randomly.
   a. The particle position should be randomly initialized using binary labels "0" and "1".

3. For each particle:
    a. Apply the objective function (OF_1 or OF_2) and get the new pbest.
4. Choose the best pbest from all the particles and place it in gbest.
5. Do:
    a. For each particle:
        (i) Calculate objective function.
        (ii) Update pbest.
    b. Update gbest.
    c. Update the new velocity of the particle based on Equation 3.1.
    d. Update the new position of the particle based on Equation 3.3.
6. Until the termination condition (maximum iterations or a specific value of pbest, or no improvement in pbest) is met.

In the next section, we will explain how to use the voting method in the ensemble for finding the final context words.

3.3. Apply voting method in the ensemble to find final context words

A simple machine learning model usually has limited capability. In recent years the ensemble has been used instead of a simple model because it is difficult to find the best model. The ensemble combines several models to provide a better and more accurate output. The output of the ensemble is more representative. The advantage of using an ensemble is that it uses the ability of different models to estimate different patterns. Also, the errors of one model are compensated by other models. As a result, using an ensemble of models is more efficient than using a simple machine learning model [20, 37]. Research has shown that to have a high-performance ensemble; we must use strong models in the construction of the ensemble and leave out the weak models [20].

Ensembles usually consist of three sub-functions: 1- Choose a suitable base model depending on the problem. Choosing a base model with high accuracy has a significant impact on the accuracy and efficiency of the ensemble. 2- We have to sample the training data several times and use the sampled data for training by the base algorithm and produce several base models. 3- Combine the results of base models to obtain an accurate output. Majority voting methods are usually used to aggregate the results of simple base models and generate ensemble output.

In this research, we use the BPSO-based labeling method as a base model. Then we create the dictionary using 45K most frequent words, including 20K nouns, 10K verbs, 10K adjectives, and 5K adverbs. Then we consider the 5K most frequent words of the corpus (including nouns, verbs, adjectives, and adverbs) as the primary context words. Then, we obtain the co-occurrences of the target words in the dictionary with the primary context words using the exponential function $e^{-0.1\alpha}$ and put the resulting co-occurrence numbers in the matrix X_{SD}. The exponential function assigns a higher coefficient to context words that are closer to the target word. Component x_{ij} shows the co-occurrence of the target word corresponding to the i_{th} row and the primary context word corresponding to the j_{th} column. Parameter α is the absolute value of the simple distance between the target word and the primary context word. Also, we obtain the co-occurrence matrix of the target words in the dictionary (45K) and the primary context words (5K) using the dependency

distance between the target word and the primary context word and put it in the matrix X_{DD}. We use co-occurrence matrices X_{SD} and X_{DD} as training examples. To solve the BPSO-based labeling problem, we use two constraints $N = 500$ and $N = 1000$. We also use two different objective functions OF_1 and OF_2, to solve the problem. We solve the problem of selecting N final context words in the following cases and assign a binary label to each primary context word. We consider eight different base optimization models for ensemble construction:

1. Matrix X_{SD}, N = 1000, and objective function OF_1.
2. Matrix X_{DD}, N = 1000, and objective function OF_1.
3. Matrix X_{SD}, N = 1000, and objective function OF_2.
4. Matrix X_{DD}, N = 1000, and objective function OF_2.
5. Matrix X_{SD}, N = 500, and objective function OF_1.
6. Matrix X_{DD}, N = 500, and objective function OF_1.
7. Matrix X_{SD}, N = 500, and objective function OF_2.
8. Matrix X_{DD}, N = 500, and objective function OF_2.

Spearman correlation coefficient of word similarity task on MEN, RG-65, and SimLex-999 test sets in matrix X_{DD} is better than the matrix X_{SD}. Spearman's correlation coefficient of test sets in cases $N = 1000$ than in cases $N = 500$ is higher. We assign labels to primary context words using the eight base optimization models mentioned above. Then, we create an ensemble and combine the results of these base optimization models. We use the voting method to assign the final label to the primary context words and select the N final context words. Then, we use the N final context words to create the co-occurrence matrices X'_{SD} and X'_{DD} using the simple distance and dependency distance, respectively. Next, we evaluate the resulting word vectors on test sets using the word similarity task. The evaluation results are reported in Section 4.2. The results show that using the ensemble, labeling the primary context words, and selecting the final context words are improved compared to the base models.

4. Experimental Evaluations

4.1. Corpus

The ukWaC corpus [6] is a huge corpus for the English language that contains over one billion words. The corpus was created by web crawling. The corpus is used as a general resource for the English language. The corpus includes the Part Of Speech Tag (POS) and the dependency parsing index. In this research, we use the first part of the ukWaC (namely ukwac_dep_parsed_01) to overcome computational constraints. Then, we evaluate the resulting word vectors on the word similarity task using the MEN, RG-65, and SimLex-999 test sets.

4.2. Evaluation results and discussions

As mentioned in Section 3.3, we create a dictionary using 20K most frequent nouns, 10K most frequent verbs, 10K most frequent adjectives, and 5K most frequent adverbs in the corpus. We also consider 5K of the most frequent words (including nouns, verbs, adjectives, and adverbs) as primary context words. Next, we construct the co-occurrence matrices X_{SD} and X_{DD} using the dictionary target words and the primary context words. Then, we obtain the Spearman correlation coefficient of the co-occurrence matrices X_{SD} and X_{DD} on the MEN, RG-65, and SimLex-999 test sets. The results are reported in Table 1. As you can see in Table 1, the Spearman correlation coefficient of the matrix X_{DD} is higher than the matrix X_{SD} in the MEN, RG-65, and SimLex-999 test sets by 0.65%, 6.5%, and 5.08%, respectively. As a result, we find that the word vectors obtained using the dependency distance are more accurate.

Table 1: Spearman correlation coefficient of matrices X_{SD} and X_{DD}.

test sets	Matrix X_{SD}	Matrix X_{DD}
MEN dataset	67.79	68.44
RG-65 dataset	56.33	62.83
SimLex-999 dataset	26.73	31.81

In the first step, to evaluate the first base model, we solve the BPSO-based labeling problem using constraint N = 1000, data X_{SD}, and the objective function OF_1. Then, we get three of the best solutions obtained by the labeling optimization algorithm (which maximizes the objective function OF_1). We construct the vectors of the dictionary target words using the final context words and evaluate them on the test sets. The results of evaluating word vectors using three of the best solutions obtained are reported in Table 2. As you can see in Table 2, the accuracy of the 1000-dimensional word vectors has decreased compared to the matrix X_{SD}, which has 5000 dimensions. The accuracy drop is expected because we have lost considerable amounts of data to maintain interpretability in dimensional reduction operations. Solution 3 obtained better Spearman correlation coefficients than solutions 1 and 2. Spearman correlation coefficient of solution 3 compared to matrix X_{SD} decreased for the MEN test set by 2.36%. Also, the Spearman correlation coefficient increased by 0.8% and 0.31% in RG-65 and SimLex-999 test sets, respectively. We use the labels obtained by the first base model in solutions 1, 2, and 3 to build the ensemble.

Table 2: Spearman correlation coefficient of first base model solutions in comparison to matrix X_{SD}.

test sets	Solution 1	Solution 2	Solution 3	Matrix X_{SD}
MEN dataset	65.69	64.94	65.43	67.79
RG-65 dataset	51.08	56.68	57.13	56.33
SimLex-999 dataset	24.43	23.69	27.04	26.73

To evaluate the second base model, we solve the BPSO-based labeling problem with constraint N = 1000 using data X_{DD} and the objective function OF_1. Then, we get three of the best solutions obtained by the labeling optimization algorithm (which maximizes the objective function OF_1). Then, we construct the dictionary target words vectors obtained by using the N=1000 final context words and evaluate them on the test sets. The evaluation results of the best solutions 1, 2, and 3 are presented in Table 3. The Spearman correlation coefficient of the second solution is much larger than

solutions 1 and 3. Therefore, we consider the second solution as the best answer obtained from the labeling problem. Spearman correlation coefficient of the second solution compared to matrix X_{DD} decreased on MEN and SimLex-999 test sets by 5.21% and 4.72%, respectively. Also, the Spearman correlation coefficient in the RG-65 test set increased by 2.91%. Examining the results in Table 3, we find that the accuracy drop in the word vectors obtained by the dependency distance is more severe. We use solutions 1, 2, and 3 obtained in the second base model to construct the ensemble.

Table 3: Spearman correlation coefficient of second base model solutions in comparison to Matrix X_{DD}.

test sets	Solution 1	Solution 2	Solution 3	Matrix X_{DD}
MEN dataset	63.04	63.23	63.50	68.44
RG-65 dataset	55.91	65.74	55.53	62.83
SimLex-999 dataset	27.02	27.09	28.34	31.81

To evaluate the third base model, we solve the PSO-based labeling problem using constraint N = 1000, data X_{SD}, and the objective function OF_2. Then, we get three of the best solutions obtained by the labeling optimization algorithm (which minimizes the objective function OF_2). Next, using N = 1000 final context words, we construct the dictionary target word vectors and evaluate them on the test sets. The evaluation results of solutions 1, 2, and 3 are presented in Table 4. We consider the third solution the best solution by comparing the results obtained by solutions 1, 2, and 3. The Spearman correlation coefficient of the second solution compared to the matrix X_{SD} is decreased in the MEN and SimLex-999 test sets by 1.18% and 2.71%, respectively. In the RG-65 test set, the Spearman correlation coefficient is increased by 2.04%. In the third base model, we use solutions 1, 2, and 3 to construct the ensemble.

Table 4: Spearman correlation coefficient of third base model solutions in comparison to matrix X_{SD}.

test sets	Solution 1	Solution 2	Solution 3	Matrix X_{SD}
MEN dataset	65.83	66.61	65.18	67.79
RG-65 dataset	55.63	58.37	56.99	56.33
SimLex-999 dataset	24.53	24.02	23.58	26.73

Next, to evaluate the fourth base model, we solve the PSO-based labeling problem with constraint N = 1000 using the data X_{DD} and the objective function OF_2. Then, we select three of the best solutions obtained by the labeling optimization algorithm (which minimizes the objective function OF_2). Then, we construct the vectors of the dictionary target words based on the final context words obtained by each solution and evaluate them on the test sets. Spearman correlation coefficients of solutions 1, 2, and 3 are reported in Table 5. We use the three solutions obtained by the fourth base model to construct the ensemble. Solution 2 got the best Spearman correlation coefficients in the RG-65 and SimLex-999 test sets. Solution 3 also got the best Spearman correlation coefficient on the MEN test set. The second solution compared to matrix X_{DD} decreases the Spearman correlation coefficient of MEN, RG-65, and SimLex-999 datasets by 5.34%, 1%, and 5.08%, respectively.

To evaluate the fifth base model, we solve the PSO-based labeling problem using constraint N = 500, the data X_{SD}, and the objective function OF_1. We select the three best solutions that maximize the objective function OF_1 and construct the dictionary target word vectors using the final context words of each solution. Then, we evaluate the explicit word vectors that have 500 dimensions on the

Table 5: Spearman correlation coefficient of fourth base model solutions in comparison to matrix X_{DD}.

test sets	Solution 1	Solution 2	Solution 3	Matrix X_{DD}
MEN dataset	62.85	63.10	65.95	68.44
RG-65 dataset	54.18	61.83	57.84	62.83
SimLex-999 dataset	26.28	26.73	26.00	31.81

test sets. The evaluation results are shown in Table 6. The first solution in comparison to matrix X_{SD} decreases the spearman correlation coefficients on MEN, RG-65, and SimLex-999 test sets by 3.33%, 3.37%, and 3.27%, respectively. We use the three solutions 1, 2, and 3 obtained by the fifth base model to construct the ensemble.

Table 6: Spearman correlation coefficient of fifth base model solutions in comparison to matrix X_{SD}.

test sets	Solution 1	Solution 2	Solution 3	Matrix X_{SD}
MEN dataset	64.46	64.21	63.89	67.79
RG-65 dataset	52.96	50.43	51.58	56.33
SimLex-999 dataset	23.46	24.60	23.96	26.73

To evaluate the sixth base model, we solve the PSO-based labeling problem using constraint N = 500, data X_{DD}, and the objective function OF_1. Then, we select three of the best solutions obtained by the labeling optimization algorithm (which maximizes the objective function OF_1). Then, we construct the dictionary target words vectors using N=500 final context words obtained from each solution and evaluate them on the test sets using the word similarity task. The evaluation results of solutions 1, 2, and 3 are presented in Table 7. Solution 1 provides a better Spearman correlation coefficient than solutions 2 and 3. Of course, in this base model, the accuracy drop is very severe. The first solution in comparison to the matrix X_{DD} decreases the Spearman correlation coefficient by 7.72%, 7.78%, and 3.67% on the MEN, RG-65, and SimLex-999 test sets, respectively. This sharp accuracy drop is expected because we have retained only 500 of the 5,000 dimensions and leave out a large amount of information. Results show that the accuracy loss due to dimension reduction in the word vectors obtained by the dependency distance is greater than the simple distance.

Table 7: Spearman correlation coefficient of sixth base model solutions in comparison to matrix X_{DD}.

test sets	Solution 1	Solution 2	Solution 3	Matrix X_{DD}
MEN dataset	60.72	63.23	60.30	68.44
RG-65 dataset	55.05	43.81	54.61	62.83
SimLex-999 dataset	28.14	29.93	26.66	31.81

To evaluate the seventh baseline model, we solve the PSO-based labeling problem using constraint N = 500, data X_{SD}, and the objective function OF_2. Next, we select three of the best solutions to the labeling problem that minimizes the objective function OF_2. Then, we construct the dictionary target words vectors using N=500 final context words obtained by each solution and evaluate them on the test sets using the word similarity task. Evaluation results of the labeling problem solutions are reported in Table 8. The Spearman correlation coefficient of the second solution compared to

matrix X_{SD} is decreased by 4.12%, 2.83%, and 2.05% on MEN, RG-65, and SimLex-999 test sets, respectively. We derive solutions 1, 2, and 3 from the seventh base model to construct the ensemble.

Table 8: Spearman correlation coefficient of seventh base model solutions in comparison to matrix X_{SD}.

test sets	Solution 1	Solution 2	Solution 3	Matrix X_{SD}
MEN dataset	62.10	63.67	61.19	67.79
RG-65 dataset	50.97	53.50	50.24	56.33
SimLex-999 dataset	25.30	24.68	22.68	26.73

Next, to evaluate the eighth base model, we solve the PSO-based labeling problem by constraint N = 500, data X_{DD}, and the objective function OF_2. Next, we select three of the best optimization problem solutions that minimize the objective function OF_2. Then, we obtain the target word vectors of the dictionary using N=500 final context words obtained by each solution, and evaluate the resulting word vectors on the test sets. The results of the evaluations are reported in Table 9. Evaluation results of solution 2 are better than solutions 1 and 3 on the test sets. Spearman correlation coefficient of word vectors obtained by the second solution compared to matrix X_{DD} is decreased on MEN, RG-65, and SimLex999 test sets by 7.08%, 2.78%, and 3.78%, respectively. Solutions 1, 2, and 3 are used to construct the ensemble.

Table 9: Spearman correlation coefficient of eighth base model solutions in comparison to matrix X_{DD}.

test sets	Solution 1	Solution 2	Solution 3	Matrix X_{DD}
MEN dataset	59.54	61.36	61.73	68.44
RG-65 dataset	56.32	60.05	56.77	62.83
SimLex-999 dataset	25.23	28.03	26.79	31.81

In the second step, we create an ensemble using three of the best BPSO-based labeling problem solutions for each base model. And we count the number of labels "1" assigned to each primary context word. For each primary context word, a number is obtained to indicate the number of labels "1" assigned. Then, we try to determine the final label of each primary context word. For this reason, we do an ablation study to find the threshold of voting. Our studies show that a threshold smaller than four does not achieve the goal of obtaining low-dimensional interpretable word vectors. So, we consider the thresholds:

1. $vote \geq 4$
2. $vote \geq 5$
3. $vote \geq 6$
4. $vote \geq 7$

Then, we construct the co-occurrence matrices X'_{SD} and X'_{DD} using the final context words obtained by each threshold. Then, we evaluate the dictionary target word vectors on the test sets. The evaluation results are reported in Table 10. In the matrix X'_{SD} , compared to matrix X_{SD} in

threshold $vote \geq 4$, the Spearman correlation coefficient on the MEN and RG-65 test sets is decreased by 3.87% and 1.13%, respectively. In the SimLex-999 test set, the accuracy has increased by 1.27%. By reducing the dimensions of word vectors from 5000 to 2557, the accuracy on the SimLex-999 test set is improved. Also, the Spearman correlation coefficient of matrix X'_{SD} compared to matrix X_{SD} in threshold $vote \geq 5$ is decreased on MEN and RG-65 test sets by 0.93% and 1.13%, respectively. Therefore, by reducing the dimensions of word vectors from 5000 to 1507, we see a 1% accuracy drop on the MEN and RG-65 test sets. Also, the Spearman correlation coefficient is increased by 1.65% on the SimLex-999 test set. The matrix X'_{SD} in threshold $vote \geq 6$ has 713 dimensions. The matrix X'_{SD} compared to matrix X_{SD}, leaves out 4287 primary context words. In this case, the Spearman correlation coefficient in MEN and RG-65 test sets has decreased by 2.34% and 1.76%, respectively. The accuracy of the SimLex-999 test set increases by 0.12%. In threshold $vote \geq 7$, word vector dimension reduced sharply from 5000 to 298. So, the Spearman correlation coefficient is decreased dramatically on MEN, RG-65, and SimLex-999 by 3.87%, 7.53%, and 2.21, respectively. In thresholds $vote \geq 4$, $vote \geq 5$, and $vote \geq 6$, the accuracy improves in the SimLex-999 test set. The accuracy improvement is justified because in the co-occurrence matrix X_{DD} compared to the matrix X_{SD}, the Spearman correlation coefficient on the SimLex-999 test set is 5.08% higher. As a result, by using the ensemble, we benefit from the strengths of base models that use the data X_{DD}.

Table 10: Spearman correlation coefficient Matrix X'_{SD} using the voting method.

	Matrix X_{SD}	Matrix X'_{SD}			
voting threshold	no voting	$vote \geq 4$	$vote \geq 5$	$vote \geq 6$	$vote \geq 7$
Num of context words	5000	2557	1507	713	298
MEN dataset	67.79	63.92	66.86	65.45	63.92
RG-65 dataset	56.33	55.78	55.20	54.57	48.80
SimLex-999 dataset	26.73	28.00	28.38	26.85	24.52

In the next step, we construct the matrix X'_{DD} using the final context words obtained from thresholds $vote \geq 4$, $vote \geq 5$, $vote \geq 6$, and $vote \geq 7$. Then, we evaluate the resulting word vectors on the test sets. The evaluation results are reported in Table 11. By reducing the word vectors dimensions to 2557 by threshold $vote \geq 4$, we see a decrease in Spearman correlation coefficient on the MEN and RG-65 test sets by 2.5% and 2.45%, respectively. Also, the Spearman correlation coefficient of the SimLex-999 test set is increased by 0.3%. By reducing the word vectors dimensions to 1507 by threshold $vote \geq 5$, the Spearman correlation coefficient on MEN, RG-65, and SimLex-999 test sets is decreased by 3.4%, 2.26%, and 0.56%, respectively. This decrease in accuracy is justifiable, as we have omitted 3493 dimensions from the word vector and left out the amount of information to reduce vectors dimensions. In the threshold $vote \geq 6$, the number of word vector dimensions is 713. In this threshold the Spearman correlation coefficient for MEN, RG-65, and SimLex-999 test sets is decreased by 4.98%, 6.11%, and 3.31%, respectively. In the threshold $vote \geq 7$, the word vector dimensions are reduced from 5000 to 298. In evaluating the word vectors using threshold $vote \geq 7$, the Spearman correlation coefficient on the MEN, RG-65, and SimLex-999 sets is decreased significantly by 5.71%, 10.46%, and 6.6%, respectively.

By examining Tables 10 and 11 and comparing their results with the results of each base model, we find that we have been able to reduce the dimensions of the vectors by using the ensemble, and we also experience less reduction in the Spearman correlation coefficient compared to each base model. Depending on the processing power of researchers, they can use the appropriate number of dimensions for their research. As shown in Tables 10 and 11, the Spearman correlation coefficients

Table 11: Spearman correlation coefficient Matrix X'_{DD} using the voting method.

	Matrix X_{DD}		Matrix X'_{DD}		
voting threshold	no voting	$vote \geq 4$	$vote \geq 5$	$vote \geq 6$	$vote \geq 7$
Num of context words	5000	2557	1507	713	298
MEN dataset	68.44	65.94	65.04	63.46	62.73
RG-65 dataset	62.83	60.38	60.57	56.72	52.37
SimLex-999 dataset	31.81	32.11	31.25	28.50	25.21

of word vectors by thresholds $vote \geq 4$ and $vote \geq 5$ are not significantly different. Therefore, we propose to use the threshold $vote \geq 5$, to produce interpretable low-dimensional word vectors in which word vectors have 1507 dimensions. In this case, in the matrix X'_{SD} , we have removed 3493 primary context words and selected only 1507 final context words. Therefore, the Spearman correlation coefficient change occurs on the MEN, RG-65, and SimLex-999 test sets by -0.93, -1.13, and +1.65, respectively. Also, in the matrix X'_{DD}, in threshold $vote \geq 5$ using 1507 dimensions, the Spearman correlation coefficient on the MEN, RG-65, and SimLex-999 test sets is decreased by 3.4%, 2.26%, and 0.56%, respectively. As mentioned before, by reducing the word vector dimensions, a more severe decrease in the resulting word vectors is observed using the dependency distance than the simple distance. However, because the words vectors obtained by the dependency distance have a higher Spearman correlation coefficient than the simple distance, they are preferred to construct more accurate word vectors.

5. Conclusion

Our goal in this study is to obtain interpretable low-dimensional word vectors. To achieve this goal, we propose a labeling method based on the BPSO algorithm. To solve the labeling problem, we use two different co-occurrence matrices that use the simple distance and dependency distance as the data of the problem. To construct co-occurrence matrices, we use 5K of the most frequent words in the corpus as primary context words. To solve the labeling optimization problem, we define two different objective functions: 1- Maximizing Spearman correlation coefficient and 2- Minimizing the sum of squares of the word vector's differences using implicit word vectors obtained by Word2Vec software. Then, to solve the problem, we consider eight different base models and solve the labeling problem for each base model. We use three of the best answers obtained by the labeling problem for each base model to build the ensemble. After creating the ensemble, we count the number of labels "1" for each of the 5K primary context words. Then, using different voting thresholds, $vote \geq 4$, $vote \geq 5$, $vote \geq 6$ and $vote \geq 7$ extracted from an ablation study, we assign the final label to each primary context word. After applying the voting method, primary context words that have the label "1" are final context words. Using the final context words obtained by each voting threshold, we construct the dictionary target word vectors by simple distance and dependency distance and place them in the matrices X'_{SD} and X'_{DD}, respectively. Next, we evaluate the target word vectors of matrices X'_{SD} and X'_{DD} using each voting threshold. The evaluation results show that using the ensemble and the voting method, the word vector dimensions in the cases $vote \geq 4$, $vote \geq 5$, $vote \geq 6$, and $vote \geq 7$ reduce to 2557, 1507, 713, and 298, respectively. In the co-occurrence matrix X'_{SD}, the Spearman correlation coefficient of word vectors in thresholds $vote \geq 4$, $vote \geq 5$ is increased by 1.13% and 1.65%, respectively on the SimLex-999 test set. These results indicate the utilization of base models expertise based on data X_{DD} in the ensemble. In the threshold $vote \geq 5$,

we see an accuracy drop of about 1% in the MEN and RG-65 test sets. An accuracy drop of about 1% on the test sets is justified due to word vector dimensions reduction from 5000 to 1507. Also, in the matrix X'_{DD}, we see a 2 -3% accuracy drop on the MEN and RG-65 test sets. The accuracy drop on the SimLex-999 test set is a small number of 0.5%. In threshold $vote \geq 7$ using 298 dimensions, a sharp accuracy drop occurs on the word vectors of matrices X'_{SD} and X'_{DD}. We can use final context words in this threshold as golden words for NLP tasks such as finding the title and keywords extraction. In this study, we tried to create low-dimensional explicit word vectors (1507 dimensions) by the least accuracy drop. If the researchers wish, we will provide them the final context words. Researchers can apply this approach to their favorite corpus, extract the final context words, and create interpretable low-dimensional word vectors.

References

[1] Abdelsalam, A., Bojar, O., & El-Beltagy, S. R, *Bilingual embeddings and word alignments for translation quality estimation*, In Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, (2016, August) 764-771.

[2] Amoozegar, M., & Minaei-Bidgoli, B, *Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism*, Expert Systems with Applications, 113, (2018) 499-514.

[3] Alguliyev, R. M., Aliguliyev, R. M., & Abdullayeva, F. J, *PSO+ K-means algorithm for anomaly detection in Big Data*, Statistics, Optimization & Information Computing, 7(2), (2019) 348-359.

[4] Baroni, M, *Composition in distributional semantics*, Language and Linguistics Compass, 7(10),(2013) 511-522.

[5] Baroni, M., Dinu, G., & Kruszewski, G, *Don't count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors*, In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (2014, June) 238-247.

[6] Baroni, M., Bernardini, S., Ferraresi, A., & Zanchetta, E, *The WaCky wide web: a collection of very large linguistically processed web-crawled corpora*, Language resources and evaluation, 43(3), (2009) 209-226.

[7] Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T, *Enriching word vectors with subword information*, Transactions of the Association for Computational Linguistics, 5, (2017) 135-146.

[8] Biemann, C., & Riedl, M, *Text: Now in 2D! a framework for lexical expansion with contextual similarity*, Journal of Language Modelling, 1(1), (2013) 55-95.

[9] Bordag, S, *A comparison of co-occurrence and similarity measures as simulations of context*, In International Conference on Intelligent Text Processing and Computational Linguistics, (2008, February) 52-63, Springer, Berlin, Heidelberg.

[10] BinSaeedan, W., & Alramlawi, S, *CS-BPSO: Hybrid feature selection based on chi-square and binary PSO algorithm for Arabic email authorship analysis*, Knowledge-Based Systems, (2021) 107224.

[11] Chen, J., Gong, Z., & Liu, W, *A nonparametric model for online topic discovery with word embeddings*, Information Sciences, 504, (2019) 32-47.

[12] Esposito, M., Damiano, E., Minutolo, A., De Pietro, G., & Fujita, H, *Hybrid query expansion using lexical resources and word embeddings for sentence retrieval in question answering*, Information Sciences, 514, (2020) 88-105.

[13] Eberhart, R. C., & Shi, Y, *Computational intelligence: concepts to implementations*, Elsevier, (2011).

[14] Févotte, C., & Idier, J, *Algorithms for nonnegative matrix factorization with the β-divergence*, Neural computation, 23(9),(2011) 2421-2456.

[15] Faruqui, M., Tsvetkov, Y., Yogatama, D., Dyer, C., & Smith, N, *Sparse overcomplete word vector representations*, arXiv preprint arXiv:1506.02004,(2015).

[16] Gamallo, P, *Comparing explicit and predictive distributional semantic models endowed with syntactic contexts*, Language Resources and Evaluation, 51(3), (2017) 727-743.

[17] Gamallo, P., & Bordag, S, *Is singular value decomposition useful for word similarity extraction?*, Language resources and evaluation, 45(2), (2011) 95-119.

[18] Giatsoglou, M., Vozalis, M. G., Diamantaras, K., Vakali, A., Sarigiannidis, G., & Chatzisavvas, K. C, *Sentiment analysis leveraging emotions and word embeddings*. Expert Systems with Applications, 69, (2017) 214-224.

[19] Jha, K., Wang, Y., Xun, G., & Zhang, A, *Interpretable word embeddings for medical domain*, In 2018 IEEE international conference on data mining (ICDM) (2018, November) 1061-1066.

[20] Kim, S. Y., & Upneja, A, *Majority voting ensemble with a decision trees for business failure prediction during economic downturns*, Journal of Innovation & Knowledge, 6(2), (2021) 112-123.

[21] Khan, F.H., Qamar, U. and Bashir, S *SentiMI: Introducing point-wise mutual information with SentiWordNet to improve sentiment polarity detection*, Applied Soft Computing, 39, (2016) 140-153.
[22] Kennedy, J.,& Eberhart, R, *Particle swarm optimization*, In Proceedings of ICNN'95-international conference on neural networks, Vol. 4, (1995, November) 1942-1948, IEEE.
[23] Lenci, A, *Distributional models of word meaning*, Annual review of Linguistics, 4, (2018) 151-171.
[24] Landauer, T. K., & Dumais, S. T, *A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge*, Psychological review, 104(2),(1997) 211.
[25] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J, *Distributed representations of words and phrases and their compositionality*, In Advances in neural information processing systems, (2013) 3111-3119.
[26] Murphy, B., Talukdar, P., & Mitchell, T, *Learning effective and interpretable semantic models using non-negative sparse embedding*, In Proceedings of COLING (2012, December) 1933-1950.
[27] Naderalvojoud, B., & Sezer, E. A, *Sentiment aware word embeddings using refinement and senti-contextualized learning approach*, Neurocomputing, 405, (2020) 149-160.
[28] Orhan, U., & Tulu, C. N, *A novel embedding approach to learn word vectors by weighting semantic relations: SemSpace*, Expert Systems with Applications, 180, (2021) 115146.
[29] Pennington, J., Socher, R., & Manning, C. D, *Glove: Global vectors for word representation*, In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), (2014, October) 1532-1543.
[30] Padró, M., Idiart, M., Villavicencio, A., & Ramisch, C, *Nothing like good old frequency: Studying context filters for distributional thesauri*, In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), (2014, October) 419-424.
[31] Paul, D., Jain, A., Saha, S., & Mathew, J, *Multi-objective PSO based online feature selection for multi-label classification*, Knowledge-Based Systems, 222, (2021) 106966.
[32] Rostami, M., Forouzandeh, S., Berahmand, K., & Soltani, M, *Integration of multi-objective PSO based feature selection and node centrality for medical datasets*, Genomics, 112(6), (2020) 4370-4384.
[33] Stein, R. A., Jaques, P. A., & Valiati, J. F, *An analysis of hierarchical text classification using word embeddings*, Information Sciences, 471, (2019) 216-232.
[34] Sun, F., Guo, J., Lan, Y., Xu, J., & Cheng, X, *Sparse word embeddings using l1 regularized online learning*, In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (2016, July) 2915-2921.
[35] Subramanian, A., Pruthi, D., Jhamtani, H., Berg-Kirkpatrick, T., & Hovy, E, *Spine: Sparse interpretable neural embeddings*, In Thirty-Second AAAI Conference on Artificial Intelligence, (2018, April).
[36] Wahde, M, *Biologically inspired optimization methods: an introduction*, WIT press, (2008).
[37] Werbin-Ofir, H., Dery, L., & Shmueli, E, *Beyond majority: Label ranking ensembles based on voting rules*, Expert Systems with Applications, 136, (2019) 50-61.