Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2022) 30: 2586 – 2604

doi:10.55730/1300-0632.3957

# A rule-based/BPSO approach to produce low-dimensional semantic basis vectors set

**Atefe PAKZAD, Morteza ANALOUI***
School of Computer Engineering, Iran University of Science and Technology, Tehran, Iran

**Received:** 23.09.2021 • **Accepted/Published Online:** 27.09.2022 • **Final Version:** 28.11.2022

**Abstract:** The present study aims to generate low-dimensional explicit distributional semantic vectors. In explicit semantic vectors, each dimension corresponds to a word, which makes word vectors interpretable. In this study, a new approach is proposed to obtain low-dimensional explicit semantic vectors. Firstly, the suggested approach considers three criteria, namely, word similarity, number of zeros, and word frequency as features for words in a corpus. Next, some rules are extracted to obtain the initial basis words using a decision tree which is drawn based on the three features. Secondly, a binary weighting method is proposed based on the binary particle swarm optimization algorithm which obtains $N_B$ = 1000 context words. In addition, a word selection method is used to provide $N_S$ = 1000 context words. Thirdly, the golden words of the corpus are extracted based on the binary weighting method. Subsequently, the extracted golden words are added to the context words which are selected by the word selection method as the golden context words. The ukWaC corpus is utilized for constructing the word vectors. MEN, RG-65, and SimLex-999 test sets are used to evaluate the word vectors. Next, the results are compared to a baseline which uses 5K most frequent words in the corpus as the context words. The baseline method uses a fixed window to count the cooccurrences. The word vectors are obtained using the 1000 selected context words along with the golden context words. Compared to the baseline method, the suggested approach can increase Spearman's correlation coefficient for the MEN, RG-65, and SimLex-999 test sets by 4.66%, 14.73%, and 1.08%, respectively.

**Key words:** Explicit word vectors, rule-based selection method, golden context words, final basis words

## 1. Introduction

It is important to identify semantic similarities or relatedness between words for many natural language processing (NLP) applications. Distributional semantic models (DSMs) obtain word vectors. DSMs have two categories. The first category is the count-based models while the second one is prediction-based models which mainly use neural methods. The count-based models produce explicit word vectors in which each vector component refers to a lexical word. Prediction-based models produce implicit word vectors [1]. The components of implicit word vectors which are produced by prediction-based models have no lexical equivalents. Implicit word vectors are used in many natural language processing tasks such as sentiment classification [2–4], part-of-speech (POS) tagging [5], named entity recognition [6], question answering [7], information retrieval [8], and recommendation systems [9, 10]. Low-dimensional explicit word vectors can help NLP tasks such as sentiment analysis, recommender systems, question answering, and information retrieval since these vectors are fully interpretable and the text information is reflected in them. For constructing explicit word vectors in the

*Correspondence: analoui@iust.ac.ir

count-based models, important context words must be identified. Next, the cooccurrence of the target words with the context words is counted and the word-context matrix is obtained. In the word-context matrix, each row corresponds to a target word and each column corresponds to a context word [11]. In count-based models, a window surrounding the target word is generally considered to use smaller contexts. After counting cooccurrences, a high dimensional sparse matrix X is constructed. Each matrix cell $X_{(i,j)}$ contains the association between the target word $w_i$ and the context word $c_j$. After counting cooccurrences, a high dimensional sparse matrix X is constructed. Each matrix cell $X_{(i,j)}$ contains the association between the target word $w_i$ and the context word $c_j$. A well-known measure for the association is pointwise mutual information (PMI). The association criterion is very effective in computing the similarity of words. PMI is the log ratio between P(w,c) and P(w)P(c). The PMI is defined as follows ([12]):

$$PMI(w,c) = log\frac{(P(w,c))}{(P(w)P(c))}. \tag{1}$$

Joint probability P(w,c) determines how often two words cooccur. The product of marginal probability P(w)P(c) indicates how often the two words are expected to cooccur when they occur independently. Positive PMI (PPMI) is a more common approach which replaces all negative values by 0 ([12]):

$$PPMI(w,c) = \max\left(PMI(w,c), 0\right). \tag{2}$$

Therefore, the PPMI measure should be applied to $X_{i,j}$ which corresponds to the $i_{th}$ target word and the $j_{th}$ context word. The PPMI measure removes the bias of the raw cooccurrence numbers of the matrix X. Each row of the resulting PPMI matrix corresponds to a target word vector. The similarity or relatedness between word pairs is obtained by comparing the semantic vectors of the words. In this paper, $N_B$ important context words are selected using the binary particle swarm optimization algorithm (PSO). The authors in [13] developed a heuristic method called PSO. The PSO algorithm uses a set of particles flying in the problem space. Flying particles follow optimal particles and move to a promising area to achieve a global optimum. The position of each particle is considered a potential solution. Particles are the candidate solutions and start their fly from random positions in the search space [14]. Due to the discrete nature of many problems, the authors in [15] proposed the discrete version of the PSO, which uses discrete binary variables. The sigmoid function is used to map the continuous velocity presented in Eq. (3) to [0, 1]. The sigmoid function is shown in Eq. (4) [15]. The superscript i refers to $i_{th}$ particle number. The superscript j refers to $j_{th}$ bit of the velocity of the particle. The position of a particle should be updated by Eq. (5) [16]. The position and velocity of the $j_{th}$ bit of the $i_{th}$ particle in the $k_{th}$ iteration are denoted by $X_k^{ij}$ and $V_k^{ij}$, respectively.

$$V_{k+1}^{ij} = w.V_k^{ij} + c_1 r_1 (p_k^{ij} - X_k^{ij}) + c_2 r_2 (p_k^{gj} - X_k^{ij}), \tag{3}$$

$$V_k'^{ij} = sig(V_{k+1}^{ij}) = \frac{1}{1 + e^{-V_k^{ij}}}, \tag{4}$$

$$X_{k+1}^{ij} = \begin{cases} 1, & \text{if } r^{ij} < sig(V_{k+1}^{ij}). \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

Parameters $c_1$ and $c_2$ are cognitive and social learning acceleration coefficients, respectively. These parameters are arbitrary constants in $[0, +\infty]$. Parameters $r_1$ and $r_2$ are random numbers in the range of 0–1. The best position of the $j_{th}$ bit of the $i_{th}$ particle is determined by $p_k^{ij}$. The best global position in the swarm up to $k_{th}$ iteration for the $j_{th}$ bit is shown by $p_k^{gj}$ [14]. $r^{ij}$ represents a random number with a uniform distribution in the range [0,1]. In this paper, a new approach is introduced for the first time to select N informative context words as final basis vectors. Firstly, the most frequent words are considered the candidate context words. Next, WS (word similarity) and NZ (number of zeros) criteria are computed for each candidate context word based on implicit word vectors produced by word2vec software. In addition, the common word frequency (WF) criterion (word frequency in the corpus) is used to select the context words. The proposed method extracts some rules to select informative context words based on WF, WS, and NZ criteria using the decision tree. In the next step, a binary weight is assigned to each candidate context word using the binary particle swarm optimization (BPSO) algorithm to select $N_B$ context words. $N_S$ context words are selected using the word selection method via distance matrices. Afterward, the golden context words are extracted using the binary weighting method. Context words whose BPSO weights are one are selected for different executions. The $N_G$ golden context words are added to the $N_S$ context words obtained by the word selection method. The target word vectors are obtained after selecting the $N = N_S + N_G$ final basis words. Next, the resulting word vectors in the word similarity task are evaluated. The results show a significant improvement in the accuracy of the test sets such as MEN [17], RG-65[18], and SimLex-999 [19] datasets.

This paper is structured as follows. Section 2 provides a review of the related literature. In Section 3, the methodology and the proposed methods for producing semantic basis vectors set are described. The experimental setups are explained in detail in Section 4. The discussion and conclusions are presented in Sections 5 and 6.

## 2. Literature review

In count-based models, word distributions are characterized by high-dimensional sparse explicit vectors. Hence, explicit word vectors with high dimensions are mapped to a space of fewer latent dimensions, which causes implicit vectors with smaller dimensions to appear. Given that explicit word vectors have high dimensions, dimensional reduction methods can be used to produce implicit word vectors. The most common way to create implicit representation is to map a word-context matrix to a reduced latent semantic space with matrix reduction algorithms such as singular value decomposition (SVD), principal component analysis (PCA), and nonnegative matrix factorization (NMF) [20, 21]. Predictive models have been recently used to infer implicit dense vectors using neural network methods. Word similarity computations are easier for dense vectors since low-dimensional matrix operations are needed. Count-based models produce explicit word vectors with high dimensions, but they can perform effective computations using hashing functions. In the hashing functions, the keys are word-context pairs whose values are nonzero scores. In explicit semantic models, there are many nonexistent relations with zero value in the word vector. Previous studies showed that these nonexistent relations could be ignored [22]. Thus, for each word, only context words which have a nonzero value in the word vector can be stored. In general, little attention has been paid to creating low-dimensional explicit word vectors in count-based models.

Past studies ([22] and [23]) used filtering strategies to reduce the dimensions of explicit vectors by choosing the most relevant context words for each word. Other studies ([20] and [24]) described a count-based model which used the idea of filtering contexts to reduce nonzero values. The filtering method keeps only the R-relevant context words based on the highest likelihood score in the hash table. A recent study [24] indicated

that low dimensional vectors produced by SVD were not more computationally efficient than sparse matrices which used hash tables for nonzero values. Additionally, vectors with low implicit dimensions did not have more generalizations in comparison to explicit word vectors. The authors in [12] showed that implicit word vectors generated based on skip-gram and negative sampling implicitly created a word-context matrix. Each component of the word-context matrix was equal to the PMI weight of the word and context pairs. This method is very similar to count-based models using dimension reduction methods such as SVD to obtain dense implicit word vectors, which indicates the similarities between count-based models and prediction-based models. Therefore, good context filters produce low-dimensional explicit word vectors. Low dimensional explicit word vectors can be used in several NLP tasks. The authors in [25] believe that semantic models do not need meaningless latent variables to represent word vectors. Thus, word cooccurrences should be used instead of reduced latent variables. Implicit word vector dimensions do not have linguistic equivalence whereas explicit word vectors are interpretable since each word vector dimension equals a word that is extracted from the corpus.

## 3. Method

In this section, the proposed approach is described to obtain low-dimensional explicit word vectors. N words are selected from the corpus as the final basis words. Each final basis word is equivalent to a basis vector. Therefore, the resulting word vectors are meaningful and can reflect the contextual information of the corpus. The main steps of the proposed approach are illustrated in Figure 1.

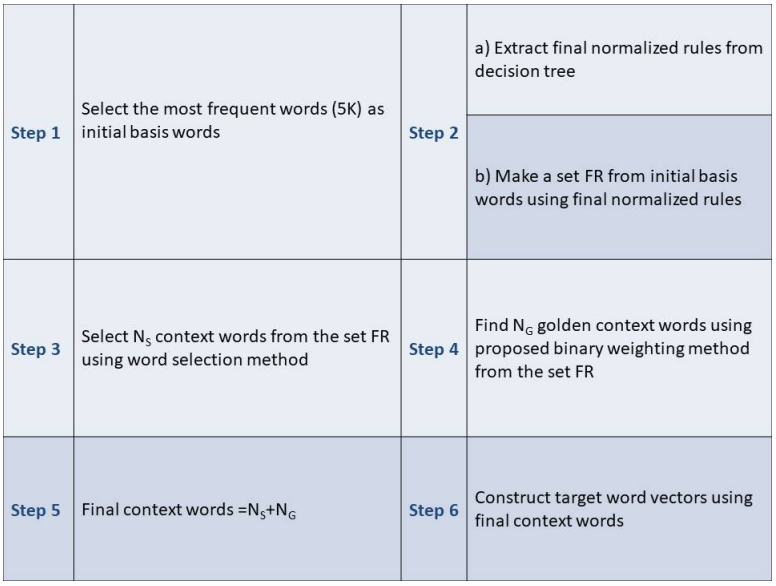



**Figure 1**. The main steps of the proposed approach.

The main steps of the proposed approach are described in detail below.

### 3.1. Extracting rules to select initial basis words

The decision tree is a hierarchical data structure for classification and regression applications. An important advantage of the decision tree is its interpretability as well as its potential to be converted to a set of rules. The decision tree, which has internal decision nodes and final leaves, is created by a sequence of successive splits

[26]. The output of each leaf is a label which specifies the sample class in the classification problems. Each local area in the input space specifies a class. The boundaries are determined by separators, which are defined by internal decision nodes from root to final leaves. In the decision tree classifier, a path from the root to the leaf determines the conditions for reaching the leaf class label. The conditions for reaching from the root to each leaf can be written using a set of IF-THEN rules. Each rule explains why a class label is selected by the decision tree classifier. The set of extracted rules is called the rule base and can be used to extract knowledge [26, 27]. In this section, three features are defined for words so as to draw their decision tree. Additionally, we explain how to obtain initial basis words based on the rules extracted from the decision tree classifiers. Note that C is the candidate set which includes the 10K most frequent words in the corpus. Two criteria, namely, word similarity (WS) and the number of zero components (NZ) in the word vector are computed. To calculate the word similarity criterion for a particular word, the similarity of a particular word with other words in the set C is assessed by the implicit word vectors obtained by the word2vec software. Furthermore, to obtain the criterion of the number of zero components in a particular word vector, the number of zero components of the particular word vector, which is obtained by the word2vec software, is counted. Word vectors obtained by this software have no zero components. Thus, if the absolute value of the vector component is less than 0.01, the component is assumed as zero. Furthermore, word frequency in the corpus is calculated as a common criterion for obtaining context words. To this end, for each word in the candidate set C, the word similarity, the number of zero components, and word frequency criteria are calculated. Next, $\mathcal{M}$ members of set C with the highest word similarity criterion are placed in set S. In addition, $\mathcal{M}$ members of set C with the highest frequency are put in set F. Additionally, $\mathcal{M}$ members with the highest number of zero components are placed in the set Z. The union of three sets S, F, and Z is defined as triple context words set (Triple set). Finally, the 5K most frequent words of the corpus are put in set A. We extract the initial rules for obtaining initial basis words based on the features WS, NZ, WF are given in the following steps:

1. Labeling the common words of set A and set Triple (set Common) by '1'.
2. Labeling the words in set Triple which are not in set A (set $IN = Triple - A$) by '2'.
3. Labeling the words in set A which are not in set Triple (set $OUT = A - Triple$) by '3'.
4. Let $U_{AT} = A \cup Triple$.
5. Computing WS, NZ, and WF criteria for all words in $U_{AT}$.
6. Drawing a decision tree for all words in the set $U_{AT}$.
7. Drawing a decision tree for nouns in the set $U_{AT}$.
8. Drawing a decision tree for verbs in the set $U_{AT}$.
9. Drawing a decision tree for adjectives in the set $U_{AT}$.
10. Drawing a decision tree for adverbs in the set $U_{AT}$.
11. Extracting the common rules between the 5 decision trees depicted in steps 6–10.
12. Normalizing the rules obtained in step 11 based on the infinity norm.
13. Selecting words from candidate set C which satisfy the initial rules in step 12 as IR set.

Some words with different labels but similar features could be found by carefully examining the decision trees. Therefore, they are considered noisy samples. The decision tree for all words in the set $U_{AT}$ is shown in Figure 2. Variables x1, x2, and x3 correspond to WS, WF, and NZ criteria, respectively.

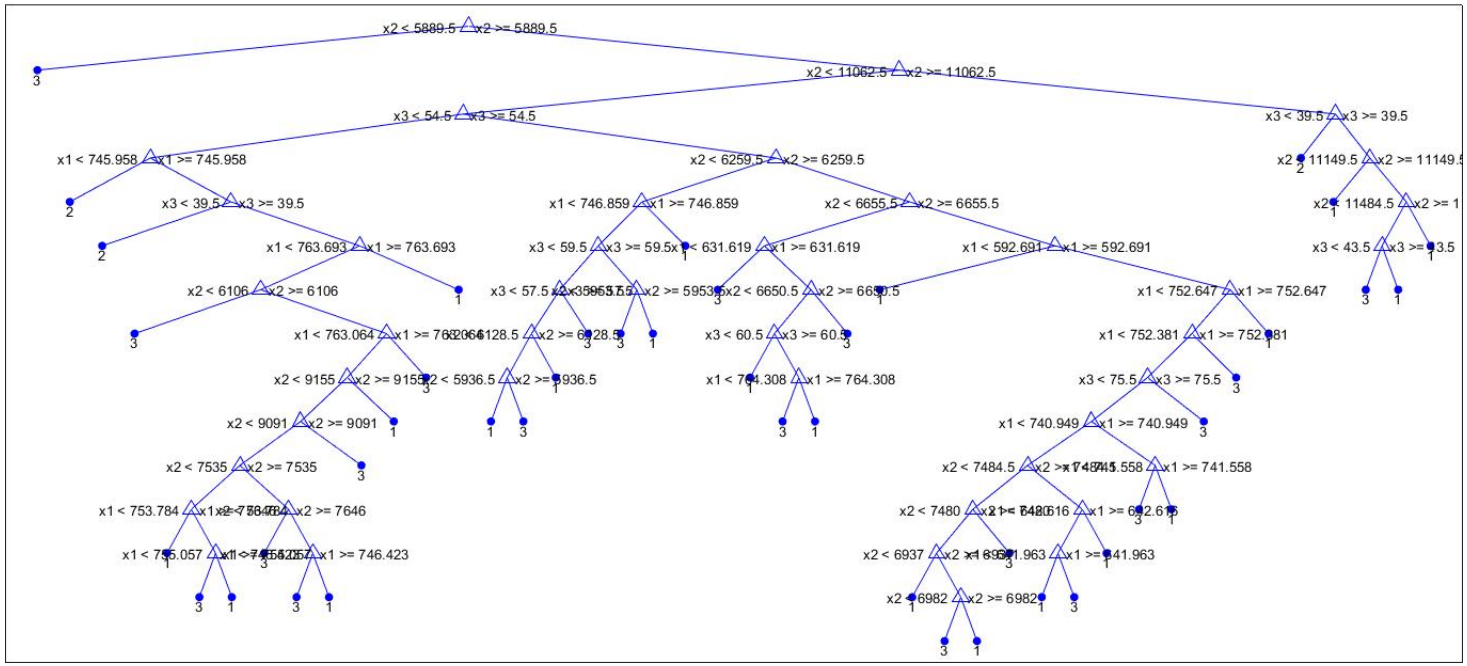



**Figure 2**. The classification tree for all words in the set $U_{AT}$.

The classification tree of nouns, verbs, adjectives, and adverbs is used so that the extracted rules are not based on a particular POS tag. To find the initial rules:

1. Discard the branches containing the noisy nodes.
2. Consider the numbers of nodes in the five decision trees and find the corresponding groups.
3. All five decision trees start by the WF criterion. Thus, we start the initial rule by the WF criterion.
4. Pay attention to the inequalities to find the number corresponding to each group. If the inequality is larger, the smallest number is selected; otherwise, the largest number is selected.

For each word in set C, the initial rules extracted from the decision trees are shown in Algorithm 1. A closer look at the trees indicates that a small WF means that the word is not selected while a very large WF criterion is suitable for selecting a word. Words with intermediate WF (5900–11,063) are needed to satisfy the large WS criterion. The abovementioned initial rules are normalized. Each word in set C which satisfies the normalized rules is placed in the IR set. $\mathcal{M} = 3000$ is used to evaluate the triple context words set. This set contains 4867 words including noun, verb, adjective, and adverb lemmas. Additionally, the IR set obtained according to the initial normalized rules contains 3770 words including nouns, verbs, adjectives, and adverbs. Two word-context matrices are constructed for the words in vocabulary to evaluate the performance of the sets Triple and IR. Word vectors are evaluated based on MEN, RG-65, and SimLex-999 test sets. The word-context matrix using set Triple increases Spearman's correlation coefficient (Spearman's $\rho$) in MEN and SimLex-999 datasets which are relatively large test sets. The results show that the initial rules extracted based on the three features WS, NZ, and WF are effective. However, Spearman's $\rho$ in the RG-65 test set is decreased. Due to the reduced accuracy of the RG-65 set, it is concluded that some words in the sets IN and OUT are mislabeled. The results of examining the selected words of the set IR are as follows:

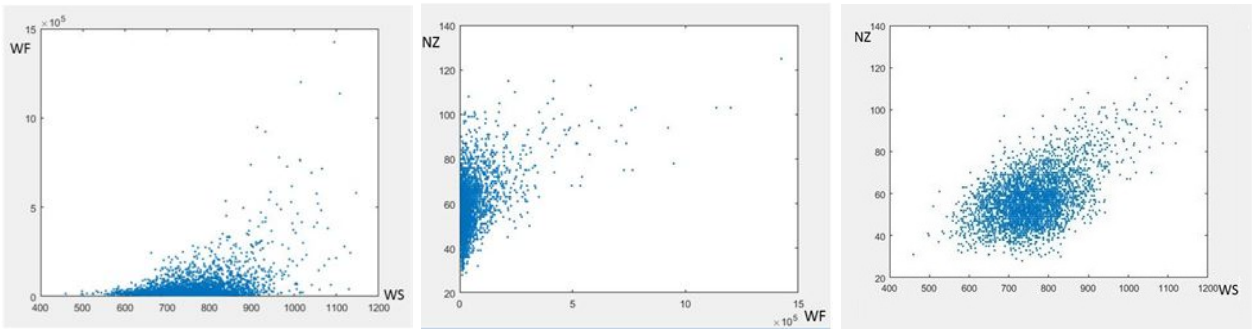
1. Some words in the sets are common and IN cannot satisfy the initial normalized rules. Thus, they are not selected.
2. Some words in set OUT satisfy the initial normalized rules.

**Algorithm 1** The initial rules extracted from the decision trees

```
if (WF < 5900) then
    Unselect
else if (WF ≥ 5900) then
    if (WF ≥ 11063 & NZ > 30) then
        Select
    else if (WF ≥ 8000 & WS > 700) then
        Select
    else if (6600 ≤ WF ≤ 11063 & WS > 763 & NZ > 39.5) then
        Select
    else if (6600 ≤ WF ≤ 11063 & WS > 643 & NZ > 54.5) then
        Select
    else if (6600 ≤ WF ≤ 11063 & WS > 746 & 39.5 ≤ NZ ≤ 54.5) then
        Select
    else if (5900 ≤ WF ≤ 6600 & WS > 763 & NZ ≥ 54.5) then
        Select
    else if (WF > 7000 & NZ ≥ 54.5) then
        Select
    else
        Unselect
    end if
end if
```

These words are likely to be classified incorrectly. Results show that many verb lemmas are selected from the set OUT. About other POS tags (nouns, adjectives, and adverbs), fewer words (about 20 or fewer)

are selected from the set OUT. A large number of words in the set IN can satisfy the initial normalized rules, which indicates that the selected context words based on the extracted rules are informative. Additionally, many words in the set OUT cannot satisfy the initial normalized rules. Failure to select a large number of ineffective words in the set OUT is an important factor which improves the word vectors obtained using the set IR. Furthermore, a large number of words in the set common are selected. The features WS, NZ, and WF are firstly studied in pairs WF-WS, NZ-WS, and NZ-WF. For words in the sets IR, Common, IN, and OUT, the word scattering is plotted in 2D based on the WF-WS, NZ-WS, and NZ-WF features pairs. Figure 3 shows the data scattering for the set Common. Figure 3(a) shows the words scattering based on the WF-WS features pair. Figure 3(b) demonstrates the words scattering based on the NZ-WF features pair. Furthermore, Figure 3(c) shows the words scattering based on the NZ-WS pair of features. The plots of the data scattering are examined using the extracted initial rules and the boundaries derived from the decision tree. For each WF-WS, NZ-WS, and NZ-WF pair, several candidate rules are proposed based on the data scattering and the initial rules. The obtained context words are evaluated using the candidate rules in each pair of WS, NZ, and WF features. Subsequently, the best candidate rule is obtained for each pair of the features. In the next step, the rules obtained for each pair of features are combined to obtain candidate rules based on three features (i.e. WS, NZ, and WF). The candidate rules are obtained based on the aforementioned three features and have almost the same Spearman's $\rho$ in the test sets. Subsequently, a rule is chosen as the final rule which can select fewer context words. It should be noted that the final rule is more effective than the initial rule since it can further improve Spearman's $\rho$ for test sets. The final normalized rule is as follows:




(a) (b) (c)

**Figure 3**. Words scattering in 2D based on (a) the WF-WS (b) the NZ-WF (c) the NZ-WS.

Next, the context words are obtained based on the abovementioned final normalized rule and placed in the set FR. The results show that Spearman's $\rho$ of word vectors using set FR as context words are larger than those which use set IR as context words. To test the generalizability, the initial and final normalized rules are applied to the ukWaC4 corpus. The experiment provided Spearman's $\rho$ which were similar to the ukWaC1 corpus. Therefore, the normalized driven rule can work very well on ukWaC4 although the rule is not based on the ukWaC4 data. Thus, the normalized rule can be generalized. The final rule is applied to the frequent words of the corpus to obtain a relatively smaller set of initial basis words. The set of initial basis words is not sufficiently small. Therefore, an optimal way of context word selection which uses a binary weighting method is introduced in the next section. Subsequently, the binary weighting method is used to select golden context words in Section 3.2.

### 3.2. BPSO-based weighting method for selecting $N_B$ context words

A weighting method is proposed based on the BPSO algorithm to select the $N_B$ context words. Firstly, set $\aleph$ is considered the context words set which has $l$ words $(w_1 ... w_l)$. A word vector is shown in Eq. (6). The binary weight vector (BW) is shown in Eq. (7). A binary weight $(bw_i)$ is computed for each word $w_i$ using the proposed weighting method. Each component of BW is the weight of the word $w_i$ . To reduce the number of context words using the optimization algorithm, binary weight '1' is only assigned to $N_B$ context words that is shown in Eq. (8).

$$\overrightarrow{W} = [v_1, v_2, , v_l], \tag{6}$$

$$BW = (bw_1, bw_2, , bw_l), \tag{7}$$

$$\Sigma_{i=1}^{l} bw_i = N_B. \tag{8}$$

The weighting method based on the optimization algorithm helps obtain the weights of the context words in such a way that the objective function of the problem is minimized. We proposed three different objective functions to solve the binary weighting problem. We performed the experiments with all three proposed objective functions. In the first objective function, we used Spearman's correlation coefficient. In the second objective function, we used the square of the differences between the word vectors. In the third objective function, we used the cosine function. The results obtained using the cosine function provided the best performance. Subsequently, in the cosine objective function, we calculated the difference in cosine similarity between the word vectors obtained by the word2vec method and the word vectors obtained by the binary weighting method. The results obtained using the difference in cosine similarity of unweight word vectors and weighted word vectors were very similar. Therefore, we considered the final objective function based on the difference in cosine similarity of the word vectors before and after applying binary weighting method. Thus, the final objective function of the optimization problem is defined as follows:

$$\begin{aligned} F(BW) &= \Sigma_{(i=1,j=1)}^{L} (Cosine(\overrightarrow{W_i}, \overrightarrow{W_j}) - Cosine(\overrightarrow{W_i'}, \overrightarrow{W_j'}))^2 \\ &= \Sigma_{(i=1,j=1)}^{L} (\frac{\overrightarrow{W_i}.\overrightarrow{W_j}}{\|\overrightarrow{W_i}, \overrightarrow{W_j}\|} - \frac{\overrightarrow{W_i'}.\overrightarrow{W_j'}}{\|\overrightarrow{W_i'}, \overrightarrow{W_j'}\|})^2 \\ &= \Sigma_{(i=1,j=1)}^{L} (\frac{\Sigma_{f=1}^{l} v_{if}.v_{jf}}{\sqrt{\Sigma_{f=1}^{l}(v_{if})^2}.\sqrt{\Sigma_{f=1}^{l}(v_{jf})^2}} - \frac{\Sigma_{f=1}^{l} v_{if}.v_{jf}.(bw_f)^2}{\sqrt{\Sigma_{f=1}^{l}(v_{if}.bw_f)^2}.\sqrt{\Sigma_{f=1}^{l}(v_{jf}.bw_f)^2}})^2. \end{aligned} \tag{9}$$

The training set contains L words. Word vectors ($\overrightarrow{W_i}$ and $\overrightarrow{W_j}$) use set $\aleph$ as context words with $l$ components. The semantic word vectors $\overrightarrow{W_i'}$ and $\overrightarrow{W_j'}$ use $N_B$ context words which have binary weight '1'. To solve the problem based on the proposed optimization algorithm, a population with NP members. Each member of this population is a particle. Each particle position is equivalent to a weight vector for the context words, i.e. it is a vector with $l$ components. Each component of the particle position specifies the binary weight of the corresponding context word. The best particle is selected after executing the optimization algorithm and has the least amount of objective function in comparison to other particles. The best particle position is considered the final binary weight vector. $v_{if}$ is the $f_{th}$ component of the $i_{th}$ word vector. $bw_f$ is the the weight of $f_{th}$

component of word vector. The weighting method based on the BPSO algorithm is described in several steps in Algorithm 2.

**Algorithm 2** Binary weighting method based on the BPSO algorithm

1. Consider NP particles.
2. For each particle, initialize particle position ($x_k^i$) and velocity ($v_k^i$). Particle position should be randomly initialized by binary weights '0' and '1', in such a way that the sum of binary weights '1' should be $N_B$. Set K=1 represents the number of repetitions.
3. Apply the objective function in Eq. 9 to each particle of the population.
4. Do the following procedure
    5. The best-known position ($p_k^i$) of the particle should be initialized by $x_k^i$ (means $p_k^i = x_k^i$).
    6. The objective function is computed for the particles again.
    7. The best-known position ($p_k^i$) of the particle should be updated by $x_k^i$.
    8. The best-known position ($p_k^i$) of each particle and best-known position ($p_k^g$) of the swarm are updated.
    9. The particle velocity is calculated according to the velocity Eq. 3.
    10. The particle position is updated according to the position Eq. 5.
11. While maximum iterations are not attained.
12. The best-known position ($p_k^g$) of the swarm is the final weight vector.

Parameters $c_1$ and $c_2$ in Eq. (3) are constant and can be used to change the weighting between personal and population experience, respectively. In the experiments, cognitive and social components are set to 0.15. The inertia weight (w) is 0.7. The number of iterations is considered 20, which is the stopping criteria, and the population size is 30. The termination criterion of the PSO algorithm includes the maximum number of generations or no further improvement in pbest.

### 3.3. Finding the golden context words

There are some important words in every textual data which reflect the main information of the text. These golden words can indicate the scope of the text and provide information related to sentiment and topic. In this section, the golden words of the corpus are obtained using the PSO-based binary weighting algorithm described in Section 3.2. One of the advantages of identifying the golden words is that they can be used as the golden basis words to obtain the target word vectors. The following steps are taken to find the golden words of the corpus:

1. Obtain the semantic word vectors of the words in the training set using the context words which are selected by the final normalized rule.
2. Apply the binary weighting algorithm to word vectors. The binary weighting algorithm must produce a vector with 1000 binary weights '1'.
3. Apply the binary weighting algorithm several times (more than 100 times). Subsequently, select the three best solutions which minimize the objective function. Thus, three binary weight vectors are selected.
4. Find the context words corresponding to the binary weight '1' for each of the three selected weight vectors.
5. Find common context words between the three selected solutions. These common words are the golden words of the corpus.

The following selection introduces another context word selection method called 'word selection method'.

### 3.4. Word selection method for selecting $N_S$ context words

The 5K most frequent words are put in the corpus, which includes nouns, verbs, adjectives, and adverbs in set A. We find $N_S$ informative context words of set A using the word selection method. Firstly, a training set which includes h words is created by the vocabulary words. To find informative context words for each context word in set A follow Algorithm 3. The $N_S$ informative words are chosen as the context words from the 5K most frequent words in set A. In Step 7, the Frobenius norm is used to calculate the distance between two matrices ($M = M_A - M_{(A-j)}$). The Frobenius norm of matrix M is obtained as follows [28]:

$$\|M\|_F = (\Sigma_{i=1}^{h}\Sigma_{j=1}^{h}(m_{ij})^2)^{\frac{1}{2}} = (trace(M^T M))^{\frac{1}{2}}. \tag{10}$$

**Algorithm 3** Finding informative context words

1. Create a word-context matrix for the words in the training set using the context words in set A. Each row of the word-context matrix represents a semantic word vector for the corresponding target word. A word vector has 5K components.
2. Create a distance matrix $M_A$ which is $h \times h$ and obtain the Euclidean distance of the word pairs which are in the training set.
3. For all $j = 1, \ldots, 5K$
    4. Remove the $j_{th}$ column from the word-context matrix.
    5. Create a distance matrix $M_{(A-j)}$ which is $h \times h$ and obtain the Euclidean distance of the word pairs which are in the training set.
    6. Calculate $M = M_A - M_{(A-j)}$.
    7. Calculate the Frobenius norm of Matrix M which determines the awareness level of the $j_{th}$ context word.
8. Select $N_S$ context words from the set A with the highest Frobenius norms.

### 3.5. Adding golden context words

The golden words obtained in Section 3.3 are considered the golden context words. The number of golden context words is denoted by $N_G$. In the first step, the golden context words are added to the $N_S$ context words which are selected by the word selection method. Subsequently, the word vectors are obtained using $N_S + N_G$ selected context words. The obtained word vectors are evaluated on the test sets. The results show that adding $N_G$ golden context words to the $N_S$ context words, which are selected by the word selection method, can significantly improve Spearman's $\rho$ on test sets. In the next step, the golden context words are added to the $N_B$ context words which are obtained by the binary weighting method. Subsequently, the word vectors are obtained using $N_B + N_G$ selected context words. The obtained word vectors are evaluated based on the test sets.

## 4. Experimental setup

The ukWaC corpus is a very large corpus for the English language which includes over a billion words. The corpus was created by web crawling. The ukWaC is used as a general-purpose source [29] and contains the part-of-speech tag and the dependency parsing index. In this study, the experiments and parameter adjustments are performed on the first part of the ukWaC, i.e. ukWaC1. Subsequently, the ability to generalize experiments is examined based on the fourth part of the ukWaC (i.e. ukWaC4). We define a vocabulary which includes 20K, 10K, 10K, and 5K of the most frequent nouns, verbs, adjectives, and adverbs. Subsequently, 5K most

frequent words are considered the context words set. An exponential coefficient $e^{(-0.1\alpha)}$ is used to calculate the cooccurrence number of a target word with a context word. Parameter $\alpha$ specifies the distance of the target word from the context word in a sentence. The component $x_{ij}$, which corresponds to the $i_{th}$ target word and the $j_{th}$ context word in the word-context matrix X, is calculated using $x_{ij} = \Sigma_{AllSentences}\, e^{(-0.1\alpha)}$.

After calculating all the components of the matrix X, the PPMI criterion is applied to each component of the matrix X to obtain the PPMI matrix. Each row of the PPMI matrix is a semantic vector of a target word. Set A contains the 5K most frequent words (including nouns, verbs, adjectives, and adverbs) in the corpus. The PPMI matrix is called $X_A$ when the words in set A are used as the context words. In Section 3.1, some rules were extracted by using the decision tree to obtain the initial basis words. Firstly, the initial normalized rules were extracted and analyzed to derive the final normalized rule. The initial and final normalized rules were obtained using the ukWaC1 corpus. To test the generalizability of the extracted rules, the initial basis words are derived for the ukWaC4 corpus which satisfies the normalized rules. Three criteria including WS, NZ, and WF are obtained for the words in candidate set C. Subsequently, infinity norm is used to normalize the criteria. Next, words in set C which can satisfy the initial normalized rule are selected as the initial basis word and added to the set IR. Any word in set C which can satisfy the final normalized rule is placed in the set FR. Subsequently, word-context matrices $X_{IR}$ and $X_{FR}$ are constructed using the initial basis words in sets IR and FR, respectively. The word vectors obtained by matrices $X_{IR}$ and $X_{FR}$ are evaluated based on the test sets. For selecting context words using the binary weighting method, 2K words in the vocabulary are selected as a training set. Subsequently, the matrices $B_A$, $B_{IR}$, and $B_{FR}$ are constructed for the words of the training set using the context words in sets A, IR, and FR, respectively. Next, the binary weighting algorithm described in Section 3.2 is applied to the training set word vectors and $N_B = 1000$ context words with a binary weight '1' are obtained. The vocabulary word vectors are obtained again using 1000 selected context words and evaluated on test sets. For selecting context word using the word selection method, 5K of the most frequent words (nouns, verbs, adjectives, and adverbs) in the corpus are used as the context words in set A. In addition, a training set, including the 8K, 4K, 4K, and 2K most frequent nouns, verbs, adjectives, and adverbs, is built. The word-context matrix is constructed for the words in the training set. Subsequently, $N_S = 1000$ context words are selected using the word selection method described in Section 3.4 . The vocabulary word vectors are obtained using the $N_S$ selected context words. The golden words of ukWaC1 and ukWaC4 are found by the method described in Section 3.3. Subsequently, the common words between the golden words of ukWaC1 and ukWaC4 are chosen. A total of 20 golden words which are common between the ukWaC1 and ukWaC4 are chosen as the golden context words. The set SM includes 1000 context words selected by the word selection method. Next, common golden context words are added to the set SM. Vocabulary word vectors are reconstructed using the context words and evaluated on test sets.

## 5. Results and discussion

In this section, we report the results of the experiments in detail. In addition, we discuss why the results are obtained and we compare them. Firstly, the 5K most frequent words of the corpus are put in set A. The matrix $X_{baseline}$ is obtained as a baseline for vocabulary items using words in set A as context words and a window=10. Subsequently, the word-context matrix $X_A$, which uses the exponential coefficient $e^{(-0.1\alpha)}$, is obtained for computing cooccurrence numbers to examine the effect of using the words distances in the sentence. Additionally, set A is used as the context words. The results of evaluating the matrices $X_{baseline}$ and $X_A$ are reported in Table 1. As can be seen, Spearman's $\rho$ of the matrix $X_A$, in comparison to the matrix $X_{baseline}$

has increased for MEN, RG-65, and SimLex-999 datasets by 1.73%, 5.74%, and 1.44%, respectively. The results show that using the distance between the target word and the context word in the sentence for constructing the word-context matrix has a significant effect on improving word vectors. The initial context words which are obtained using the initial normalized rules are placed in the set IR. Subsequently, the matrix $X_{IR}$ is obtained and evaluated based on the context words in set IR. As shown in Table 1, the matrix $X_{IR}$ has 1230 dimensions fewer than matrix $X_A$. Spearman's $\rho$ of matrix $X_{IR}$, rather than matrix $X_A$ decreases by 0.33% and 1.35% for MEN and SimLex-999 test sets, respectively. In the RG-65 test set, Spearman's $\rho$ increases by 0.71%. The reduced accuracy observed in the SimLex-999 test set could be due to some incorrect labeling in the Common, IN, and OUT sets.

Next, the word-context matrices $X_{IR}$ and $X_{FR}$ are constructed using the selected context words in sets IR and FR for ukWaC4 corpus. The results of the evaluations are reported in Table 2. The matrix $X_{FR}$ has 1621 dimensions fewer than matrix $X_A$. Spearman's $\rho$ of the matrix $X_{FR}$, in comparison to the matrix $X_A$, decreased by 0.14, 0.56, and 0.14% for the MEN, RG-65, and SimLex999 test sets, respectively. A slight decrease in accuracy is justifiable since 1621 dimensions of the word-context matrix are reduced. Table 2 shows that Spearman's $\rho$ in the matrix $X_{IR}$ is much lower than the matrix $X_{FR}$ for the RG-65 dataset. A similar result was already obtained in the ukWaC1 corpus. The results indicate the good performance of the final normalized rule for selecting the initial basis words.

**Table 1**. Spearman's $\rho$ of matrices $X_{baseline}$, $X_A$, and $X_{IR}$ for the ukWaC1 corpus.

| | matrix $X_{baseline}$ | matrix $X_A$ | matrix $X_{IR}$ |
|---|---|---|---|
| Number of context words | 5K | 5K | 3770 |
| MEN dataset | 66.89 | 68.62 | 68.29 |
| RG-65 dataset | 56.96 | 62.70 | 63.41 |
| SimLex-999 dataset | 26.22 | 27.66 | 26.31 |

**Table 2**. Spearman's $\rho$ of matrices $X_A$, $X_{FR}$, and $X_{IR}$ in the ukWaC4 corpus.

| | matrix $X_A$ | matrix $X_{FR}$ | matrix $X_{IR}$ |
|---|---|---|---|
| Number of context words | 5K | 3379 | 3697 |
| MEN dataset | 68.42 | 68.56 | 68.38 |
| RG-65 dataset | 61.77 | 61.21 | 58.98 |
| SimLex-999 dataset | 27.76 | 27.62 | 27.37 |

Next, the binary weighting method is applied on $X_A$ and $X_{FR}$ matrices to obtain $N_B = 1000$ context words. The best weight vector for each of the matrices $X_A$ and $X_{FR}$ are found, and the resulting context words are put in sets BA and BFR, respectively. Subsequently, the word-context matrices $X_{BA}$ and $X_{BFR}$ are constructed using the context words in sets BA and BFR, respectively. Subsequently, the resulting word-context matrices are evaluated on the test sets. Table 3 presents the results of evaluating the word-context matrices obtained using the ukWaC1 and ukWaC4 corpora. The matrices $X_{BA}$ and $X_{BFR}$ use only 1K context words for constructing the word vectors. According to the results shown in Table 3, Spearman's $\rho$ of the matrix $X_{BA}$ in comparison to matrix $X_A$, decreases by 1.26%, 2.55%, and 1.66% for the MEN, RG-65, and SimLex-999 test sets, respectively. In addition, Spearman's $\rho$ in the matrix $X_{BFR}$, in comparison to the matrix $X_{FR}$ for MEN and SimLex-999 sets, decreases by 2.08% and 0.61%, respectively. The accuracy is increased by 1.23%

for the RG-65 test set. The matrix $X_{BFR}$ has 2359 dimensions fewer than the $X_{FR}$. Table 3 also reports the evaluation results of the word-context matrices for the ukWaC4 corpus. In the matrix $X_{BA}$ compared to the matrix $X_A$, Spearman's $\rho$ is decreased by 2.04%, 0.24%, and 2.16% for the MEN, RG-65, and SimLex-999 test sets. In addition, Spearman's $\rho$ in the matrix $X_{BFR}$ compared to the matrix $X_{FR}$ is decreased by 1.49% and 0.58% for the MEN and SimLex-999 datasets, respectively. The accuracy is increased by 3.97% in the RG-65 set. According to the results presented in Table 3, a 1%–2.5% accuracy decrease occurs in the test sets by reducing the number of basis words to 1K using the binary weighting method. The accuracy drop could be justified by considering the fact that the word vectors have 1K interpretable dimensions.

**Table 3**. Spearman's $\rho$ of matrices $X_{BA}$, $X_A$, $X_{BFR}$, and $X_{FR}$ in the ukWaC1 and ukWaC4.

| ukWaC1 corpus | Matrix $X_{BA}$ | Matrix $X_A$ | Matrix $X_{BFR}$ | Matrix $X_{FR}$ |
|---|---|---|---|---|
| Number of context words | 1K | 5K | 1K | 3359 |
| MEN dataset | 67.36 | 68.62 | 66.75 | 68.83 |
| RG-65 dataset | 60.15 | 62.70 | 64.92 | 63.69 |
| SimLex-999 dataset | 27.00 | 27.66 | 27.23 | 27.84 |
| ukWaC4 corpus | Matrix $X_{BA}$ | Matrix $X_A$ | Matrix $X_{BFR}$ | Matrix $X_{FR}$ |
| Number of context words | 1K | 5K | 1K | 3379 |
| MEN dataset | 66.38 | 68.42 | 67.07 | 68.56 |
| RG-65 dataset | 61.53 | 61.77 | 65.18 | 61.21 |
| SimLex-999 dataset | 25.60 | 27.76 | 27.04 | 27.62 |

In this step, $N_S = 1000$ context words are selected from set FR by the word selection method and placed in the SFR set. Furthermore, $N_S = 1000$ words are selected from set A and put in set SA using the word selection method. Subsequently, the word-context matrices $X_{SA}$ and $X_{SFR}$ are constructed by the context words in sets SA and SFR, respectively. Next, the resulting word-context matrices are evaluated. Table 4 demonstrates the results of evaluating the word-context matrices using the ukWaC1 corpus. Table 4 shows the results of evaluating the word-context matrices using the ukWaC4 corpus. As shown in Table 4, when 1K context words are selected from set A using the word selection method, Spearman's $\rho$ in vocabulary word vectors for MEN, RG-65, and SimLex-999 sets decrease by 2.36%, 2.98%, and 0.94%, respectively. Furthermore, by selecting 1K words from the set FR using the word selection method, there is a 1.7% and 2.76% accuracy drop for the MEN and SimLex-999 test sets, respectively. The accuracy of the RG-65 test set is increased by 4.77%. Table 4 presents the evaluation results of the word-context matrices using the ukWaC4 corpus. In the matrix $X_{SA}$, in comparison to the matrix $X_A$, Spearman's $\rho$ is decreased by 1.77%, 4.65%, and 1.84% for the MEN, RG-65, and SimLex-999 test sets, respectively. By decreasing 2379 context words from the set FR, the accuracy of vocabulary word vectors in the MEN, RG-65, and SimLex-999 test sets is increased by 0.39%, 1.62%, and 0.43%, respectively. The comparison of the results presented in Tables 3 and 4 indicates that the accuracy drop in the binary weighting method is less than the word selection method. Next, the binary weighting method is used to get the golden words of the corpora ukWaC1 and ukWaC4 in the set G. Subsequently, the golden context words in set G are added to the context words in set SA for ukWaC1 and ukWaC4. Next, $X_{(SA+G)}$ is computed for ukWaC1 and ukWaC4 (Table 5). Spearman's $\rho$ of the matrix $X_{(SA+G)}$, in comparison to $X_{SA}$, by using ukWaC1 corpus for MEN, RG-65, and SimLex-999 test sets is increased by 5.32%, 9.36%, and 0.2%, respectively. Furthermore, by adding golden context words to set G using ukWaC4 corpus, the accuracy is

increased for MEN, RG-65, and SimLex-999 test sets by 5.09%, 10.39%, and 1.44%, respectively. The results indicate that adding golden context words can dramatically improve Spearman's $\rho$ for MEN and RG-65 test sets in comparison to $X_{SA}$ and $X_A$.

**Table 4**. Spearman's $\rho$ of matrices $X_{SA}$, $X_A$, $X_{SFR}$, and $X_{FR}$ in ukWaC1 and ukWaC4.

| ukWaC1 corpus | Matrix $X_{SA}$ | Matrix $X_A$ | Matrix $X_{SFR}$ | Matrix $X_{FR}$ |
|---|---|---|---|---|
| Number of context words | 1K | 5K | 1K | 3359 |
| MEN dataset | 66.26 | 68.62 | 67.13 | 68.83 |
| RG-65 dataset | 59.72 | 62.70 | 68.46 | 63.69 |
| SimLex-999 dataset | 26.72 | 27.66 | 25.08 | 27.84 |
| ukWaC4 corpus | Matrix $X_{SA}$ | Matrix $X_A$ | Matrix $X_{SFR}$ | Matrix $X_{FR}$ |
| Number of context words | 1K | 5K | 1K | 3379 |
| MEN dataset | 66.65 | 68.42 | 68.95 | 68.56 |
| RG-65 dataset | 57.12 | 61.77 | 62.83 | 61.21 |
| SimLex-999 dataset | 25.92 | 27.76 | 28.05 | 27.62 |

**Table 5**. Spearman's $\rho$ of matrices $X_{(SA+G)}$ ,$X_{SA}$, and $X_A$ in the ukWaC1 and ukWaC4.

| | Matrix $X_{(SA+G)}$ | Matrix $X_{SA}$ | Matrix $X_A$ | Matrix $X_{(SA+G)}$ | Matrix $X_{SA}$ | Matrix $X_A$ |
|---|---|---|---|---|---|---|
| Corpus | ukWaC1 | ukWaC1 | ukWaC1 | ukWaC4 | ukWaC4 | ukWaC4 |
| Number of context words | 1085 | 1K | 5K | 1087 | 1K | 5K |
| MEN dataset | 71.58 | 66.26 | 68.62 | 71.74 | 66.65 | 68.42 |
| RG-65 dataset | 69.08 | 59.72 | 62.70 | 67.51 | 57.12 | 61.77 |
| SimLex-999 dataset | 26.92 | 26.72 | 27.66 | 26.73 | 25.29 | 27.76 |

Next, the common words among the golden words of the two corpora (ukWaC1 and ukWaC4) are selected and put in the set $G_c$. The words in the set $G_c$ are added as golden context words to the set SA. The results obtained on the ukWaC1 corpus are reported in Table 6. By adding the golden context words to the set SA, Spearman's $\rho$ of the vocabulary word vectors for the MEN, RG-65, and SimLex-9999 test sets are increased by 6.01%, 11.97%, and 0.58%, respectively. In addition, Spearman's $\rho$ in the matrix $X_{(SA+G_c)}$ is significantly improved in comparison to the matrix $X_A$ for the MEN and RG-65 test sets by 3.74% and 8.99%, respectively. Table 6 shows the results of adding set $G_c$ to the set SA using the ukWaC4 corpus. As can be seen, there is a significant increase in the accuracy of the matrices $X_{(SA+G_c)}$ in comparison to the matrices $X_{SA}$. Adding the golden context words to the set BA decreases Spearman's $\rho$ by 1%–3% on the test sets. This finding is expected since the context words in sets BA and BFR are obtained using the optimization algorithm, and adding new words to the sets destroys the optimality. Lack of optimality reduces Spearman's $\rho$. Given the fact that the words in set SA are selected by comparing the distance matrices, a significant increase in Spearman's $\rho$ occurs by adding the golden context words. Spearman's $\rho$ for $X_{(SA+G_c)}$, in comparison to the matrices $X_{SA}$, is increased for MEN, RG-65, and Simlex-999 test sets by 5.6%, 12.34%, and 1.44%, respectively. The results indicate that it would be effective to add golden context words in set $G_c$ to the context word sets which are selected by the word selection method.

**Table 6**. Spearman's $\rho$ of matrices $X_{(SA+G_c)}$ ,$X_{SA}$, and $X_A$ in the ukWaC1 and ukWaC4.

| | Matrix $X_{(SA+G_c)}$ | Matrix $X_{SA}$ | Matrix $X_A$ | Matrix $X_{(SA+G_c)}$ | Matrix $X_{SA}$ | Matrix $X_A$ |
|---|---|---|---|---|---|---|
| Corpus | ukWaC1 | ukWaC1 | ukWaC1 | ukWaC4 | ukWaC4 | ukWaC4 |
| Number of context words | 1020 | 1K | 5K | 1020 | 1K | 5K |
| MEN dataset | 72.36 | 66.26 | 68.62 | 72.25 | 66.65 | 68.42 |
| RG-65 dataset | 71.69 | 59.72 | 62.70 | 69.46 | 57.12 | 61.77 |
| SimLex-999 dataset | 27.30 | 26.72 | 27.66 | 27.36 | 25.29 | 27.76 |

As shown in Table 7, Spearman's $\rho$ of the matrix $X_{(SA+G_c)}$, in comparison to the matrix $X_{baseline}$ , using ukWaC1 corpus is increased by 5.47%, 14.64%, and 1.03% for the MEN, RG-65, and SimLex-999 test sets, respectively. Additionally, the matrix $X_{(SFR+G)}$ , in comparison to the matrix $X_{baseline}$, increases the accuracy by 4.66%, 14.73%, and 1.08% for the MEN, RG-65, and SimLex-999 datasets, respectively. The best Spearman's $\rho$ for the low-dimensional explicit word vectors using ukWaC1 corpus for the MEN, RG-65, and SimLex-999 datasets are 72.36%, 71.69%, and 27.30%, respectively. Furthermore, the best accuracy in the ukWaC4 corpus for the MEN, RG-65, and SimLex-999 test sets is 72.25%, 69.46%, and 27.36%, respectively.

In this research, our goal is to produce interpretable low-dimensional word vectors in such a way that it does not decrease Spearman's correlation coefficient after reducing the dimensions. As described above, by reducing the word vectors dimensions from 5000 to 1020, Spearman's correlation coefficient of the test sets using the proposed approach is increased significantly. To quantify the interpretability of the resulting word vectors and compare word vectors with other studies, we quantified the interpretability using the interpretability score criterion proposed in the article [30]. Interpretability scores obtained on word vectors derived from word2vec [31] and FastText [32] methods, matrices $X_A$ and $X_{(SA+G_C)}$, and the method $\tau^*$ presented in the article [30] are reported in Table 8. As you can see, the interpretability of the word2vec and FastText methods is much lower than the proposed approach. Moreover, the interpretability score of the matrix $X_{(SA+G_C)}$ is 0.23 higher than the matrix $X_A$. In addition, the interpretability score of the matrix $X_{(SA+G_C)}$ is 0.9 higher than the method $\tau^*$ presented in the paper [30]. Therefore, we conclude that the proposed method is efficient and successful.

**Table 7**. Spearman's $\rho$ of word-context matrices $X_{(SA+G_c)}$, and $X_{baseline}$.

| | Matrix $X_{baseline}$ | Matrix $X_{(SA+G_c)}$ | Matrix $X_{(SA+G_c)}$ |
|---|---|---|---|
| Corpus | ukWaC1 | ukWaC1 | ukWaC4 |
| Number of context words | 5K | 1020 | 1020 |
| MEN dataset | 66.89 | 72.36 | 72.25 |
| RG-65 dataset | 56.96 | 71.69 | 69.46 |
| SimLex-999 dataset | 26.22 | 27.30 | 27.36 |

**Table 8**. Interpretability scores of different methods in the ukWaC1.

| Method | method $\tau^*$ [30] | $X_{(SA+G_C)}$ | $X_A$ | FastText | word2vec |
|---|---|---|---|---|---|
| Interpretability score | 51.5 | 52.60 | 52.37 | 28.97 | 27.36 |

## 6. Conclusion

This study proposed an approach to provide low-dimensional explicit semantic word vectors. The 5K most frequent words of the corpus were put in set A. Subsequently, the word-context matrix was calculated using the words in set A as context words considering window = 10 and the resulting semantic word vectors were evaluated. The resulting word-context matrix was called $X_{baseline}$. Subsequently, the matrix $X_A$ was obtained using the words in set A as context words. The matrix components were computed by the exponential coefficient $e^{(-0.1\alpha)}$ versus a fixed window. Next, a decision tree was drawn based on three criteria including WS, NZ, and WF, and the initial rules were extracted and normalized using the infinity norm. By reexamining the boundaries, the final normalized rule were inferred. The initial basis words were extracted according to the final normalized rule and placed in the set FR. The binary weighting method was suggested based on the BPSO algorithm to extract context words. Subsequently, $N_B = 1000$ context words were extracted from set A and set FR and put in sets BA and BFR, respectively. The word selection method was applied to obtain context words from sets A and FR, and the selected context words were placed in sets SA and SFR, respectively. The golden words of the ukWaC1 and ukWaC4 corpora were extracted using the binary weighting method and put in set G. In addition, the common words of set G were considered for ukWaC1 and ukWaC4 as the golden context words in set $G_c$.The golden context words in set $G_c$ were added to sets SA and BA. Subsequently, the word-context matrices were constructed and the word vectors were evaluated. The golden words of ukWaC1 and ukWaC4 could be extracted for the interested researchers. A significant increase in Spearman's $\rho$ for the test sets may occur by adding the abovementioned gold context words to the sets SA. It is noteworthy that the resulting word vectors had a significant increase in Spearman's $\rho$ in comparison to $X_{baseline}$ and $X_A$ matrices. The ukWaC1 corpus was used to extract the rules. The normalized rules were applied on the ukWaC4 corpus to test the generalizability of the normalized rules. Significant improvements occurred in both ukWaC1 and ukWaC4 corpora. The extracted golden context words in set $G_c$ are as follows: $G_c$= {Coventry, intellectual, philosophy, contemporary, outcome, oxford, dr, player, dynamic, guess, objective, moment, suggest, station, take, wing, aircraft, prime, announce, several}.

## Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

## References

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

[2] Rezaeinia SM, Rahmani R, Ghodsi A, Veisi H. Sentiment analysis based on improved pre-trained word embeddings. Expert Systems with Applications 2019; 117: 139-47. doi:10.1016/j.eswa.2018.08.044

[3] Ali F, Kwak D, Khan P, El-Sappagh S, Ali A et al. Transportation sentiment analysis using word embedding and ontology-based topic modeling. Knowledge-Based Systems 2019; 174: 27-42. doi:10.1016/j.knosys.2019.02.033

[4] Deho BO, Agangiba AW, Aryeh LF, Ansah AJ. Sentiment analysis with word embedding. In: 2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST); Accra, Ghana; 2018. pp. 1-4.

[5] Pota M, Marulli F, Esposito M, De Pietro G, Fujita H. Multilingual POS tagging by a composite deep architecture based on character-level features and on-the-fly enriched Word Embeddings. Knowledge-Based Systems 2019; 164: 309-23. doi:10.1016/j.knosys.2018.11.003

[6] Nozza D, Manchanda P, Fersini E, Palmonari M, Messina E. LearningToAdapt with word embeddings: Domain adaptation of Named Entity Recognition systems. Information Processing & Management 2021; 58 (3): 102537. doi:10.1016/j.ipm.2021.102537

[7] 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 2020; 514: 88-105. doi:10.1016/j.ins.2019.12.002

[8] Bagheri E, Ensan F, Al-Obeidat F. Neural word and entity embeddings for ad hoc retrieval. Information Processing & Management 2018; 54 (4): 657-73. doi:10.1016/j.ipm.2018.04.007

[9] Sundermann C, Antunes J, Domingues M, Rezende S. Exploration of word embedding model to improve context-aware recommender systems. In: 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI); Santiago, Chile; 2018. pp. 383-388.

[10] Khattar D, Kumar V, Varma V, Gupta M. Weave&rec: A word embedding based 3-d convolutional network for news recommendation. In: 27th ACM International Conference on Information and Knowledge Management; New York, United States; 2018. pp. 1855-1858.

[11] Dobó A, Csirik J. A comprehensive study of the parameters in the creation and comparison of feature vectors in distributional semantic models. Journal of Quantitative Linguistics 2020; 27 (3): 244-71. doi:10.1080/09296174.2019.1570897

[12] Levy O, Goldberg Y, Dagan I. Improving distributional similarity with lessons learned from word embeddings. Transactions of the association for computational linguistics 2015; 3: 211-25. doi:10.1162/tacl_a_00134

[13] Eberhart R, Kennedy J. Particle swarm optimization. In: ICNN'95-international conference on neural networks; Perth, WA, Australia; 1995; pp. 1942-1948.

[14] Garg H. A hybrid PSO-GA algorithm for constrained optimization problems. Applied Mathematics and Computation 2016; 274: 292-305. doi:10.1016/j.amc.2015.11.001

[15] Kennedy J, Eberhart RC. A discrete binary version of the particle swarm algorithm. In: 1997 IEEE International conference on systems, man, and cybernetics. Computational cybernetics and simulation; Orlando, FL, USA; 1997. pp. 4104-4108.

[16] El-Maleh AH, Sheikh AT, Sait SM. Binary particle swarm optimization (BPSO) based state assignment for area minimization of sequential circuits. Applied soft computing 2013; 13 (12): 4832-40. doi:10.1016/j.asoc.2013.08.004

[17] Bruni E, Boleda G, Baroni M, Tran NK. Distributional semantics in technicolor. In: 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Jeju Island, Korea; 2012. pp. 136-145.

[18] Rubenstein H, Goodenough JB. Contextual correlates of synonymy. Communications of the ACM 1965; 8 (10): 627-33. doi:10.1145/365628.365657

[19] Hill F, Reichart R, Korhonen A. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. Computational Linguistics 2015; 41 (4): 665-95. doi:10.1162/COLI_a_00237

[20] Gamallo P. Comparing explicit and predictive distributional semantic models endowed with syntactic contexts. Language Resources and Evaluation 2017; 51 (3): 727-43. doi:10.1007/s10579-016-9357-4

[21] Lenci A. Distributional models of word meaning. Annual review of Linguistics 2018; 4: 151-71. doi:10.1146/annurev-linguistics-030514-125254

[22] Biemann C, Riedl M. Text: Now in 2D a framework for lexical expansion with contextual similarity. Journal of Language Modelling 2013; 1 (1): 55-95. doi:10.15398/jlm.v1i1.60

[23] Padró M, Idiart M, Villavicencio A, Ramisch C. Nothing like good old frequency: Studying context filters for distributional thesauri. In: Empirical Methods in Natural Language Processing (EMNLP); Doha, Qatar; 2014. pp. 419-424.

[24] Gamallo P, Bordag S. Is singular value decomposition useful for word similarity extraction? Language resources and evaluation 2011; 45 (2): 95-119. doi:10.1007/s10579-010-9129-5

[25] Hofmann MJ, Jacobs AM. Interactive activation and competition models and semantic context: from behavioral to brain data. Neuroscience & Biobehavioral Reviews 2014; 46: 85-104. doi:10.1016/j.neubiorev.2014.06.011

[26] Kubat, Miroslav, and Kubat. An introduction to machine learning. Vol. 2. Cham, Switzerland: Springer International Publishing, 2017.

[27] Alpaydin E. Introduction to machine learning. MIT press; 2020.

[28] Yang K, Shahabi C. A PCA-based similarity measure for multivariate time series. In: 2nd ACM international workshop on Multimedia databases; New York, United States; 2004. pp. 65-74.

[29] 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 2009; 43 (3): 209-26. doi:10.1007/s10579-009-9081-4

[30] Şenel LK, Utlu I, Yücesoy V, Koc A, Cukur T. Semantic structure and interpretability of word embeddings. IEEE/ACM Transactions on Audio, Speech, and Language Processing 2018; 26 (10): 1769-1779. doi:10.1109/TASLP.2018.2837384

[31] Mikolov T, Sutskever I, Chen K, S.Corrado G, Dean J. Distributed representations of words and phrases and their compositionality. In: 26th International Conference on Neural Information Processing Systems; Red Hook, United States; 2013. pp. 3111–3119.

[32] Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. Transactions of the association for computational linguistics 2017; 5: 135-146. doi:10.1162/tacl_a_00051