Data & Knowledge Engineering 144 (2023) 102110

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Ontology-based semantic retrieval of documents using Word2vec model

Anil Sharma [a,b,*], Suresh Kumar [c]

[a] USIC&T, Guru Gobind Singh Indraprastha University, Dwarka, Delhi, 110078, India
[b] College of Computing Sciences and IT, Teerthanker Mahaveer University, Moradabad, Uttar Pradesh, 244001, India
[c] Department of CSE, Netaji Subhas University of Technology East Campus, Geeta Colony, Delhi, 110031, India

ARTICLE INFO

Keywords:
Document retrieval
Keyword matching
Similarity value
Relevancy score
Word2vec model and query accuracy

ABSTRACT

Semantic retrieval of engineering knowledge is crucial in various engineering activities, such as process development and product model planning. To make a solution for this issue, previously, a few word-based semantic enabling existing approaches such as Lexical Retrieval Model with Semantic Residual Embedding's (LR-SRE), Document Retrieval Model through Semantic Linking (DR-MTS), Semantic Term Matching in Axiomatic Approaches to Information Retrieval (STM-IR) and Hybrid Ontology for Semantic Information Retrieval Model using Keyword Matching Indexing System (HOS-IR) were utilized. But, they all have shown less query-based accuracy results than the required value. In our proposed Information Retrieval (IR) design, the semantic knowledge-based retrieval scheme has been implemented. For query, entered by a user and processed for finding the dominated word. Word is then compared for its similarity equations, and similarity values are then computed to give output. Highly similar values are obtained as the class value. From class, the respective clusters are selected. Then, documents in that cluster are retrieved and ranked according to the relevance of the user. To support the accuracy level performance of this IR system, a word2vec model has been employed with the benefits of Horse Herd Optimization (HHO), which helps to extract vectors as features for classification. These results are stored as a .csv file for further retrieval. By implementing the proposed IR-word2vec model, the results showed that it outperforms other existing techniques by improved similarity index and accuracy for query results in an execution time of 1.7 s.

1. Introduction

In today's world, there is an enormous amount of data saved in digital form on the internet, putting a strain on information retrieval (IR) systems [1]. The goal of IR is to locate relevant content from vast document libraries based on consumer requests. Semantic information retrieval aims to improve traditional retrieval models by using semantic-based definitions of words in context instead of basic keyword matching [2]. Word similarity facilitates information retrieval, which is an important task. Enabling models to comprehend the fundamental semantic meaning among phrases and paragraphs could help with this endeavour. It would enable a process to determine resemblance among two textual information even though the basic phrases, words, or sentences are different. Yet, because words can vary in syntax and length, this task remains challenging [3]. The following processes must be executed in order to achieve the main goal of information retrieval, (i) Data are represented in summary content form throughout the indexing process. (ii) All stop words and frequent terms are deleted throughout the filtering process. (iii) The primary method of retrieving information is continuing to search. Algorithms like bag-of-words [4] have been used in the past to solve these semantic similarity

[*] Corresponding author.
E-mail address: anilsharma@aiactr.ac.in (A. Sharma).

https://doi.org/10.1016/j.datak.2022.102110
Received 1 November 2021; Received in revised form 18 November 2022; Accepted 18 November 2022
Available online 24 November 2022

difficulties. They are common in semantic similarity, information retrieval, and sentiment investigation tasks, although they are just for specific word instances.

Numerous methods for finding context-related word similarities are presented. LSTM [5] is one of the methods used to show that with enough data, it is possible to learn a complicated set of phrase vector representations as well as their deep meaning. Because of their memory units or cells, LSTMs are especially excellent in NLP tasks [6] because they can store data and context in the term of large paragraphs and sentences. A variety of experiments were conducted to evaluate the impact of varied proportions of a supporting component and primary element in the word representation. The addition of a supporting component in the basic representation can improve the performance of the sentence representation. Another method is to use Word2vec to convert words into word vectors that can reflect the semantic link between words [7]. Information retrieval could be improved by optimizing the semantics of words and grouping word vectors into themes. To find the IR, another way is to use the paragraph vector (PV) model [8]. Paragraph vector originates two components: the Distributed Memory Model of Paragraph Vectors (PV-DM) and the Distributed Bag of Words version of Paragraph Vector (PVDBOW).

Furthermore, some summarization approaches to the text have been used to improve the IR: semantic approaches [9], statistical approaches [10], meta-heuristic approaches [11], and machine learning approaches [12]. In the proposed method, the data are preprocessed to remove the unwanted things like web tag, stop-word etc., which is fit for words similarity analysing format. The semantic retrieval process is enhanced by the use of an advanced horse herd optimization algorithm (HHOA). HHOA is a meta-heuristic optimization algorithm that was recently established. The Horse Herd Optimization Algorithm (HOA) uses six key aspects to mimic the social behaviours of horses of various ages: grazing, hierarchy, sociability, mimicry, defence mechanism, and roam. Prior to its implementation in feature selection for high-dimensional data, the benefits of HOA should be adapted to discrete optimization problems. The proposed method gives a 0.96% similarity score and 1.6 s execution time that is a high similarity score and low processing time in contrast with previous methods.

The main contribution of the paper:

- Semantic document retrieval has been developed by representing documents in knowledge retrieval systems. With the help of domain corpus, the ontology concepts have been used to select the link between documents to create maps.
- Query-based semantic retrieval is proposed to give a solution with issues in the semantic-based methods, and it is analysed to extract the query features then converted to vector value for similarity comparison.
- A semantic retrieval method that adopts the word2vec model for finding target words from the context words to give better features for the documents indexed has been presented.
- Comparisons were made based on relevancy to feedback the user with the ranked documents.

The rest part of the paper contains: Section 2 present the literature review that is related to finding the similarity between words. In Section 3, the methodology and architecture of the proposed part are present. Section 4 present the result and discussion of the proposed outcome as well as a comparison of proposed and previous approaches. At last, Section 5 contain the conclusion of the paper.

2. Literature review

To retrieve the necessary document from a document corpus, several ideas and procedures are used. Some of the approaches that are related to finding the similarity between words are listed below.

Ilyes Khennak and Habiba Drias [13] had suggested the APSO technique for efficiently solving the problem of Web Information Retrieval (IR) Query Expansion (QE). APSO was used to solve the problem of QE as a combinatorial optimization problem. An APSO designed for QE based on MEDLINE, the world's largest medical library on the Internet. To tune the Accelerated Particle Swarm Optimization (APSO) variables, first, run a preliminary experiment. Next, the performance was analysed to the Firefly Algorithm (FA), a swarm intelligence algorithm. In addition, the performance was compared to Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and Bat Algorithm (BA), three recently released QE approaches. Compared to proposed methodology, the APSO model cannot find better similarity score between words in low vector dimensions because in proposed methodology, the cosine similarity is used that finds similarity scores of the words among two similar documents which are far apart by the Euclidean distance. As per cosine similarity equation, the smaller dimensions will produce higher cosine similarity. Hence, the Word2vec model has been used in the proposed methodology to find the similarity scores between the words

Haolin Wang et al. [14] had represented a unique medical information retrieval method that provided a two-stage query expansion technique to properly model and utilize latent semantic relationships in order to become more efficient. The method was separated into two segments: initially, a heuristic technique was utilized to increase the commonly utilized pseudo relevance feedback technique for much more appropriate query expansion by continuously expanding questions to increase the similarity score among documents and queries. Secondly, a word embedding relevance model is developed based on tensor factoring to identify semantic association patterns in sparse environments has been designed to expand retrieval performance with organized knowledge bases. The method is tested and verified for TREC CDS 2014 dataset. This method gives a good outcome and reduces the IR challenges but it is not suitable for finding words similarity in a large dataset.

Youcef Djenouri et al. [15] had suggested overcoming the basic Document Information Retrieval challenge, and an advanced data mining field was developed. Data mining methods are used to find useful knowledge, which was subsequently used by swarms to intelligently search the entire space of documents. In the preprocessing stage, two types of data mining approaches were used. The first was used the K-means algorithm to divide the group of documents among similar clusters, and the second type was used the DCI_Closed method to identify the most closed frequent phrases for each cluster that was previously created. Bees Swarm Optimization (BSO) is employed in the resolving step to thoroughly investigate the collection of texts. This method retrieves the document information but it takes more time for execution as well as a poor similarity score.

Kang Jae Lee et al. [16] has suggested a location-based service (LBS) focused on indoor activities inside a university setting that used ontology-based semantic queries. For exchanging, querying and organizing data semantically, an ontology model named "University activity ontology" was created with relation to indoor activities at a university. For semantic searches, for instance, reasoning methods are developed to retrieve and display information about the location relevant to a destination using keywords given by customers. The lowest trip from an indoor/outdoor location to an indoor destination of interest picked by users among suggested possibilities is calculated using a 3D network-based topological database structure created by linking a road system model with an indoor topological network design. Yet, the method has some drawbacks. For example, utilizing a search algorithm and extra reasoning rules, the LBS specified in the method's query processing may be better.

FEI LI et al. [17] had suggested creating a similarity measure method based on WordNet and Wikipedia that was efficient. An edge weight model for merging density and edge data that allocated a weight to each edge adaptively depending on the number of straight hyponyms of the subsumer to enhance the precision of WordNet edge-based measurements. Second, in order to enhance the computational efficiency of existing Wikipedia connect vector-based measures, a new Wikipedia link feature-based semantic similarity technique which transformed Wikipedia links into semantic knowledge as well as need to replace the previous measures' TF-IDF statistical weight method. This method is not fit for finding an accurate similarity word due to the weight-based statistical approach

Nurul Husna Mahadzir et al. [18] had suggested how to adjust many semantic similarity measures to disambiguate words from two separate languages. Two path-based as well as three information content (IC)-based measurements are included. These five metrics were compared to a test platform of 40-word pairings that included eight confusing Malay–English terms. Synset was a collection of synonyms for a single word found in WordNet. The acquired accuracy rate suggested that semantic similarity measurement was a good technique for disambiguating Malay and English terms when matched to human judgments. Yet, there are a few significant issues when specific terms may not exist in the given source, WordNet, resulting in an incorrect score

Oscar Araque et al. [19] had represented for text terms as well as lexica vocabulary, a semantic similarity measure was calculated. The study offered a predictive sentiment model based on this metric, which employed the semantic similarity measure in conjunction with embedding form. An exhaustive evaluation was conducted to determine the model's effectiveness. The overall efficiency of the proposed feature extraction and its coupling with embedding representations was empirically verified by different statistical analyses. The impact of lexical properties on extracted features was investigated in-depth using cross-lexicon and cross-dataset assessment in order to better characterize the feature extraction approach. The impact of the method is poor feature extraction, so the outcome is also poor.

Jingxiang Zhang et al. [20] have suggested a set of features of food safety incidents (FSIs) recorded by China's mainstream media, including food categories, spatial distribution, supply chain and risk factors interconnections. Initially, they created a semantic template for text data relevant to FSIs based on the findings. Also developed a multi-layer, multi-level semantic structure of rank (MMSS-Rank) method to assess the similarity of acquired food safety information to the semantic template. Then, to evaluate the correctness of the FSI data, they computed the overall results (text layer weight, semantic template weight, and keyword density matrix) as well as chose a suitable threshold. This approach is not suitable for finding similarities between words in large data due to high execution time.

From these above mentioned review techniques, some of drawbacks such as similarity score error, excess execution time and poor feature extraction are still a major issue for most of the information retrival model. Hence, an Ontology-Based Semantic Retrieval of Documents Using Word2vec Model has been implemented to overcome the above mentioned drawbacks.

3. Proposed methodology

The proposed system was developed for the semantic retrieval of documents from the database that consist of four stages such as collection of web documents, extracting keywords and page ranking, converting word to vector and finding its relevancy score for retrieval [21–23]. An ontology-based module consists of three sub-modules, such as cluster-based classification, page ranking, and Latent Dirichlet Allocation (LDA) approach to topic modelling. Initially, the topics from the dataset are given to CONtextual QUery-awarE Ranking (CONQUER) model and then to the Continuous Bag Of Words (CBOW) model of W2C-HHO, respectively. After converting the words to vectors, an encoding is done with the fixed window size. Varying this window size and its dimensions with respect to the error is necessary and it is done by using HHO optimization. Finally, semantic indexing of these vectors with respect to similarity is given to the same ontology as they are assumed to have similar contexts in the document [24–27].

From this Fig. 1, it is understandable that documents are collected for indexing, then the semantic graph for respective documents are mapped. Pre-processing has been done to remove the unwanted entities from the document, and they are then mapped. Mapped documents are then clustered, which converts the document into a word2vec model. In the word2vec model, the collected document topics are given as context words, and then a keyword for that word is subjected to convert them into vector values. Vector values are then considered as feature values. With the help of these features, the classification has been done using K-nearest Neighbour (KNN) classifier. At the end of the classification result, the classes are labelled for each document. These labelled documents are then stored in a database for future retrieval. On the other side, the query is entered by the user and processed for finding the dominated word. Word is then compared for its similarity equations, and similarity values are then computed to give output. Highly similar values are obtained as the class value. From the class, the respective clusters are selected, and documents in that cluster are retrieved and ranked according to relevance for the user.

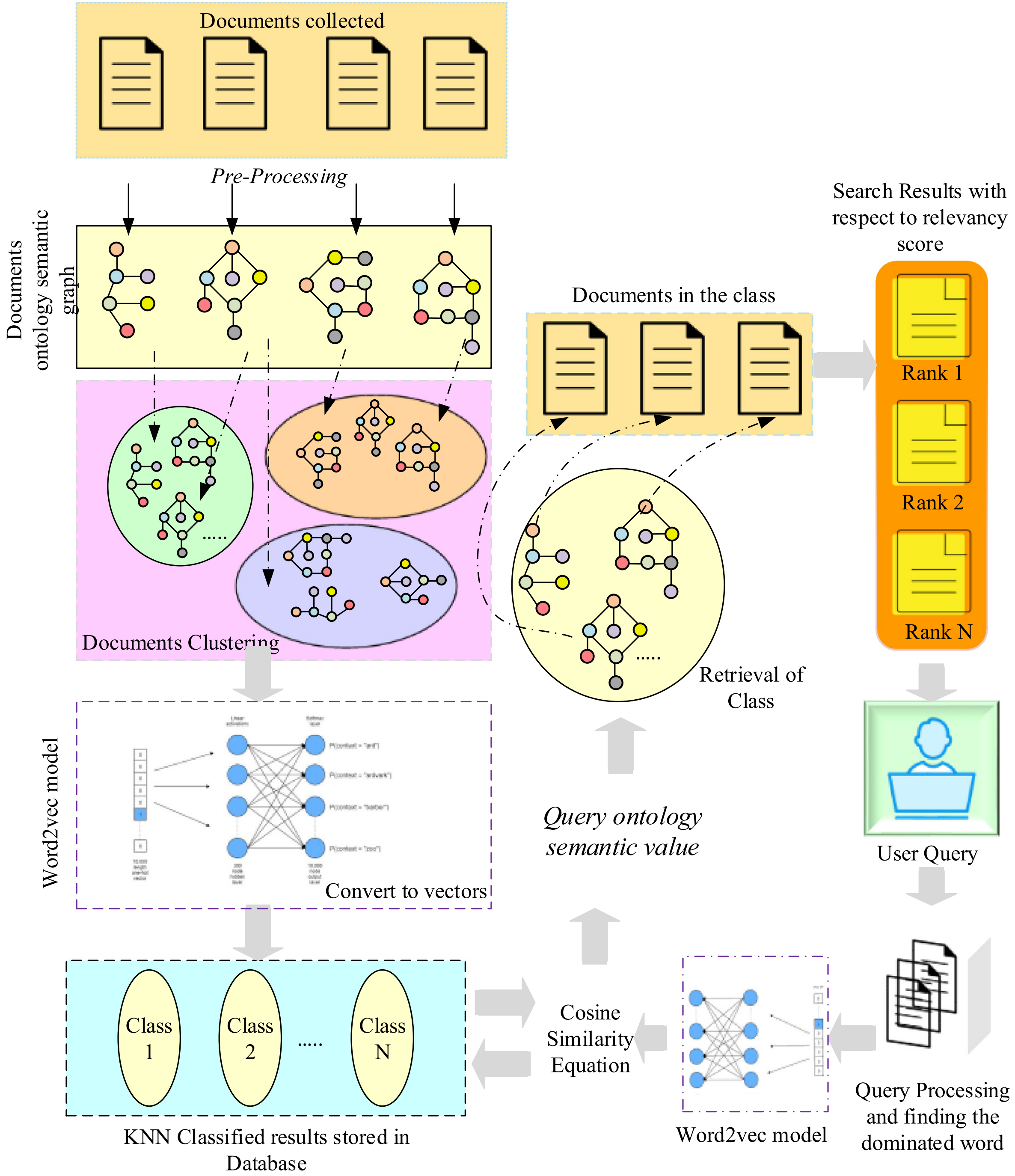


Fig. 1. Overall architecture of information retrieval system.

3.1. Semantic approach for document representation

On document-concept matching, a semantic representation method is used. As an alternative to keyword matching, concept-based matching might be considered. When both documents and queries are expressed using semantic, this may be utilized to solve polysemy and synonym issues. The method entails constructing from a set of semantic knowledge sources. Concepts are used to represent information resources and queries, allowing ambiguous terms to be disambiguated. The knowledge repository contains data about ideas and their connections to other concepts. With the aid of a knowledge repository, this will enable conceptual matching between appropriate extracted ideas. The definition of a logical semantic view is made possible by using LSI to comprehend the semantic content of a document collection. These are used to determine the set of words, the strength of word relationships, and the set of terms. LSI is a similarity metric that may be used instead of dimensionality reduction and word overlap measures. LSI methods are used to sort search results into meaningful categories, but they do not take into account the semantic links that exist between the groupings. Common phrases, on the other hand, are retrieved using the suffix tree clustering approach, and concept induction is performed using latent semantic analysis.

Domain Ontology, which is based on the WordNet ontology, may be used to aid in the formulation of a user query and give access to a document collection. From the ontological hierarchy, the medical hierarchy would have been inferred. The inquiry is mapped to the ontological user profile in this way. Each query context is semantically linked to ideas from the user's ontology. The news domain was utilized to broaden the query by incorporating various forms of normalization for the semantic component in terms of lexical chain length and document size. Basically, start with a word by document co-occurrence matrix and apply normalization to weights of uninformative words (Think tfidf). Finally, apply SVD (Singular value decomposition) to this matrix to reduce the number of features from ∼10 000 features to around 100 to 300 features which will condense all the important features into small vector space.

3.2. Pre-processing

In this stage, data is transformed into a set of word arrays that permits more preprocessing in subsequent stages. These subsequent stages are padding, stemming, tokenization, and case folding, tokenization and stop word removal. Case folding is a process of converting all words to lower case. Thus to give the same form of representation will be fed to the word2vec technique. During tokenization, the text is broken into tokens that are in the form of words with a distinct meaning, symbols and definite words. In order to remove unintended words that gives less information to the model is done stop the word removal process is done. Inflexion words that are reduced to the root are done with the help of the stemming process. The final stage of preprocessing is padding, where all the documents more or less endure the same length. Until the document length reaches the maximum limit, the token "<pad>" has been used to allow it up to the maximum number. This stage used CONQUER model [28] to identify the words for similarity score, "noun phrases" from the sentence is predicted and isolated to build synonyms, hyponyms, hypernyms and antonyms for that phrases also it again analysed for unwanted words and pruned them. At the end of preprocessing, the query starts executing in the word2vec model. In the preprocessing step, the annotated named words and the ontology concept names with their synonyms are tokenized, and the stop words are removed from the named mentions and the ontology concept names. Lowercase conversion is an essential step in the text classification model while preprocessing. Meanwhile, lowercase or uppercase forms of words are considered to have no difference; therefore, all the uppercase characters are generally converted to their lower case forms. Here the unwanted characters are removed for making the text in a structured format. 'I', 'we', 'us', 'a', 'an', 'the' etc. these types of stop words possess less important at the time of emotion recognition with respect to this, stop word removal is also an essential process.

Tokenization is performed to separate punctuation from the words for preserving their intended meaning. Also, it is carried out to separate the words/terms of each sentence. The stemming process defines that the chopping off of the end of words to a common base form. Here, a poster stemming algorithm is used to carry out the stemming process. It applies some set of rules to an input word for removing suffix and prefix and generating its stem that will be shared with other related words too. From Fig. 2, it is understandable that preprocessing has been done for both query sentences and collected documents to remove unwanted entities.

3.3. W2C-HHO model

Word Embeddings is a combined name for a set of feature modelling for language for learning techniques at Natural Language Processing that has phrases or words denoted in the form of vectors with real numbers. Word embedding used in this proposed design in the word2vec model that presents words into vector depends on various features that consist of window size and vector dimensions. It was known that similar words faced having the same vector values that are collected in the same block. Word2vec model performs similarity values between words that are trained from large corpus to resulting similarity value. Similarity values are generated from the ranges between −1 to 1 as maximum similarity values that provide an efficient implementation of the Skip-gram model and Continuous Bag of words (CBOW) to compute vector representations of words, and these are used to perform different tasks in language processing. This method first generates a vocabulary from the input words, and the word vectors are learned via backpropagation using optimization.

In order to configure the word2vec model, as shown in Fig. 3, it is trained with a large source of medical datasets. Here, all the features are generated based on window size and vector dimensions. Priory vocabulary size is V and hidden layer size is N. Adjacent units of layers are fully connected, and input is a one-hot encoded vector, that means for a given input as a context word, the output is V units, $\{x_1, \dots, x_v\}$ is set as one. Other values are set as zero, weights between input and output layer are denoted as $V \times N$ matrix W. In W, each row, represents N dimension vector representation v_w with respect to the neighbouring word of the input layer and formally i denotes row and W is taken from v_w^T. Assuming that $x_k = 1$ and $x_{k'} = 0$ for $k' \neq k$, thus it results in Eq. (1)

$$h = W^T x = {W_k}^T = v_w^T \tag{1}$$

In this Eq. (1), $W^T x$ which is essentially copying the kth row of W to h. v_w^T is the vector representation of the input word v_w^T. From the hidden layer to output layer weight matrix $\times V$, tends to compute the score u_j for each word in vocabulary as shown in Eq. (2)

$$u_j = {V'_{wj}}^T h \tag{2}$$

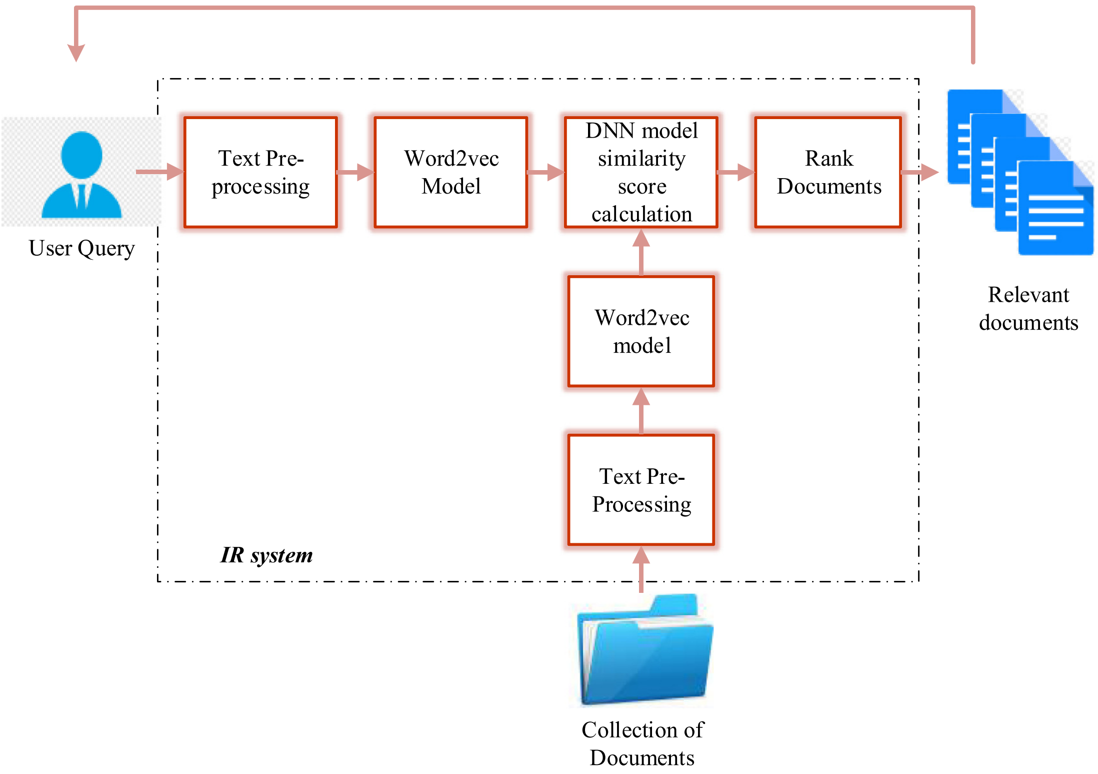


Fig. 2. Block diagram of word2vec model.

In this Eq. (2), V'_{wj} is the jth column of matrix W' and then the post-normal distribution of this activation layer is given as shown in Eq. (3)

$$P(wj|wI) = y_j = \frac{\exp(u_j)}{\sum_{j'=1}^{V}\exp(u_j{}')} \tag{3}$$

In this Eq. (3), where y_j is the output of the j-the unit in the output layer. Substitution of both (1) and (2) gave a result of Eq. (4)

$$p(wj|wI) = \frac{\exp(V'_{wj}{}^{T}V_{wI})}{\sum_{j'=1}^{V} exp(V'_{wj}{}^{T}V_{wI})} \tag{4}$$

Optimal efficiency of the Training process of word2vec embedding is enhanced by negative sampling can be given as shown in Eq. (5)

$$E = -log\sigma\left(-v'_{wo}{}^{T}h\right) - \sum_{w_j \in W_{neg}} log\sigma(-v'_{wj}{}^{T}h) \tag{5}$$

In this Eq. (5), wo is the output word and v'_{wo} is the output vector. The output value is denoted as h and the output value from the hidden layer $h = \frac{1}{C}\sum_{c=1}^{C} V_{wc}$ in CBOW model and $h = V_{wI}$, the output of CONQUER model. $W_{neg} = \left\{w_j|j = 1,\dots,M\right\}$ represents a set of words that are continuously sampled depending on $P_n(w)$ represents negative samples. In order to obtain updated equations, word vector under negative sampling takes a derivative of E with respect to net input of output unit w_j

$$\frac{\partial E}{\partial v'_{wj}{}^{T}h} = \begin{cases} \sigma\left(-v'_{wj}{}^{T}h\right) - 1 \quad if\ w_j = w_o \\ \sigma\left(-v'_{wj}{}^{T}h\right) if\, w_j \in W_{neg} \end{cases} \tag{6}$$

$$= \sigma\left(-v'_{wj}{}^{T}h\right) - t_j \tag{7}$$

In this Eqs. (6) and (7), t_j denotes label of word wj and $t = 1$ as wj is a positive sample. $t = 0$ In other cases. Further, the derivative of E with respect to an output vector of word wj that is given as denoted in Eq. (8)

$$\frac{\partial E}{\partial v'_{wj}} = \frac{\partial E}{\partial v'_{wj}{}^{T}h}.\frac{\partial v'_{wj}{}^{T}h}{\partial v'_{wj}} = \left(\sigma\left(v'_{wj}{}^{T}h\right) - t_j\right)h \tag{8}$$

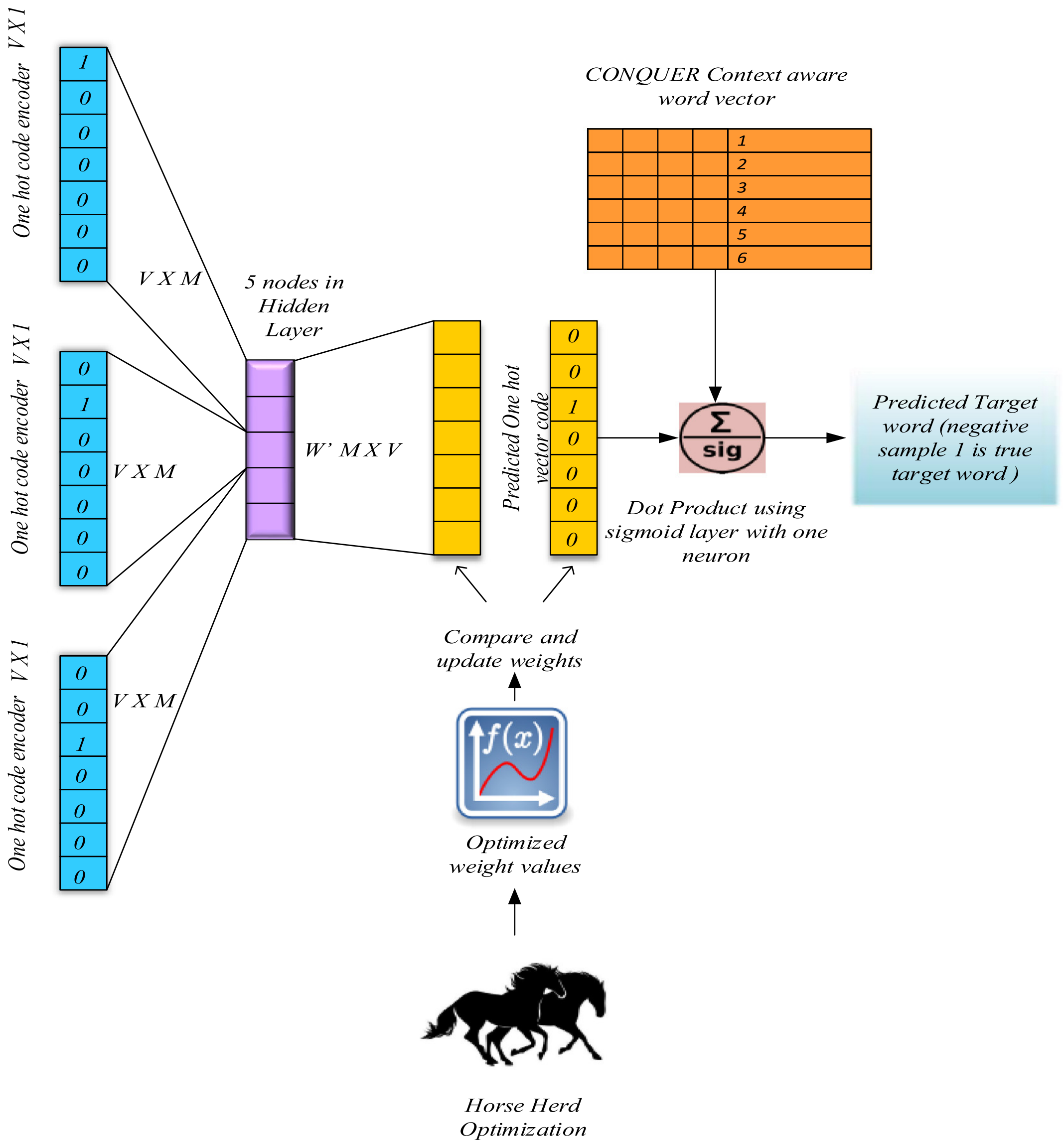

One hot code encoder V X 1
1 0 0 0 0 0 0
V X M
One hot code encoder V X 1
0 1 0 0 0 0 0
V X M
One hot code encoder V X 1
0 0 1 0 0 0 0
V X M
5 nodes in Hidden Layer
W' M X V
Predicted One hot vector code
0 0 1 0 0 0 0
CONQUER Context aware word vector
1
2
3
4
5
6
Σ
sig
Dot Product using sigmoid layer with one neuron
Predicted Target word (negative sample 1 is true target word)
Compare and update weights
f(x)
Optimized weight values
Horse Herd Optimization

Fig. 3. Word2vec model for the proposed IR.

This results in an update equation for its output vector that can be written as shown in Eq. (9)

$$v'_{wj}{}^{(new)} = v'_{wj}{}^{(old)} - \eta\left(\sigma\left(v'_{wj}{}^{T}h\right) - t_j\right)h \tag{9}$$

This output result was only needed to be employed to $wj \in \{wO\} \cup W_{-ve}$ rather than each word in the source database. Even though these neural network models get trained, it presents some errors at the hidden layer. In order to backpropagate error, an update is required for input vectors of words, and it requires a derivative E with respect to hidden layer output. The objective while training the sample is to maximize the conditional probability of observing the respective actual output word that denotes its index in the output layer. It is given as a context word W_1 with respect to weights as shown in Eq. (10)

$$fitness\ functionf\,(x) = \max p\left(w_j|w_I\right) = \max {y_j}^{*} \tag{10}$$

$$= \max log y_j * \tag{11}$$

$$= {u_j}^{*} - log \sum_{j'=1}^{V} \exp\left(u'_j\right); = -E \tag{12}$$

In order to obtain the update equations for W, the weights are now moved to W. The derivative of E on the output of the hidden layer obtained an equation of (13)

$$\frac{\partial E}{\partial h_i} = \sum_{j=1}^{V} \frac{\partial E}{\partial u_j} \cdot \frac{\partial u_j}{\partial h_i} = \sum_{j=1}^{V} e_j . w_{ij}{}' := EH_i \tag{13}$$

Here,

$$H_i = \sum_{k=1}^{V} x_K . w_{ki} \tag{14}$$

Now we can take the derivative of E with regard to each element of W, obtaining

$$\frac{\partial E}{\partial w_{ki}} = \frac{\partial E}{\partial h_i} \cdot \frac{\partial h_i}{\partial w_{ki}} = EH_i . x_k \tag{15}$$

From Eqs. (14) and (15), the objective for optimization has been clearly defined as to optimize weights, and its probability in order to minimize weights are given. With these documents, a Word2Vec model with a context distance of 6 words was trained. Words that appeared at least 10 times in the corpus were selected from the vocabulary to train the model. This subset of the corpus vocabulary (or model's vocabulary) was composed of 55,168 words. Word2Vec represents each document (the average of the embedded vector words composing the document) as a vector in a vector space whose cardinality was set to 150 dimensions.

(i) HHO optimization

The primary inspiration of HOA is the hierarchical organization of horse herds that have been evolved to live below herds. More animals are found to live in large groups that form a constant hierarchical system. For this CBOW model, network neurons are weight-optimized by maximizing the probability of word vectors as stated in the training phase. By maximizing the probability of input and output vectors, the values are optimized by selecting the appropriate weight value at the end. The following derivation shows the method to derive the weight value using HHO optimization (see Fig. 4).

Horse herd consists of a mare with the hierarchical order of horses that has specification of priority permissions calculated in the initial phase of algorithm that consists of fitness value of horses represented as a herd. Let us assume that there are K horses and P can be given as shown in Eqs. (16) and (17)

$$Herd = \left\{H_1, \dots ., H_k\right\} \tag{16}$$

$$P = Herd \rightarrow \{1 \dots K\} \tag{17}$$

If fitness $\left(H_x\right) < fitness \left(H_y\right)$ $where\ x \neq y$, here x and y $\in \{1 \dots K\}$ then

$$P\left(H_x\right) > P\left(H_y\right) \tag{18}$$

If fitness $\left(H_x\right) = fitness(H_y)$ $where\ x \neq y$, here x and y $\in \{1 \dots K\}$ then

$$\left[P\left(H_x\right) - P\left(H_y\right)\right](x - y) > 0 \tag{19}$$

Rank of each horse H_x is described by the formula

$$H_x - Rank\ of\ each\ horse = \frac{P(H_x)}{K} \tag{20}$$

Each herd has a centre that is equated with a weighted average of position to horses from which the weights otherwise represented rank of horses as shown in Eq. (20). The Centre of the horse is computed using the Eq. (21)

$$H_c = \frac{\sum_{x=1}^{k} Z_x H_x.rank}{\sum_{x=1}^{k} H_x.rank} \tag{21}$$

Distance between positions of stallion at the centre of horse herd H is calculated with the help of Euclidean distance that has the following equation of (22)

$$Dim\,(stallion, herd) = \sqrt{\sum_{y=1}^{dim} (stallion_y - herd_c)^2} \tag{22}$$

In this equation, dim is the number of dimensions of search space, and the horse consists of a set herd of horses. Then it updates velocity using the following equation of (23) and (24)

$$Vel_{x,y}^{T+1} = Vel_{x,y}^{T} + H_{x,rank} * (Herd_{center.y}^{T} - Z_{x,y}^{t}) \tag{23}$$

$$Vel_{x,y}^{T+1} = Vel_{x,y}^{T+1} + Rand * (Herd_{center.y}^{T} - Z_{x,y}^{t}) \tag{24}$$

In this Eq. (24), Rand denotes random number ranges from [0, 1]. T Denotes current iteration and new iteration $T+1$, and horse memory is a matrix that has a number of rows equal to the value of Horse Memory Pool (HMP) of horse in D columns.

$$Mem_x^{T+1} = \begin{bmatrix} Mem_{1,x,1}^{T+1} & \cdots & Mem_{1,x,D}^{T+1} \\ \vdots & \ddots & \vdots \\ Mem_{HMP,x,1}^{T+1} & \cdots & Mem_{HMP,x,D}^{T+1} \end{bmatrix} \tag{25}$$

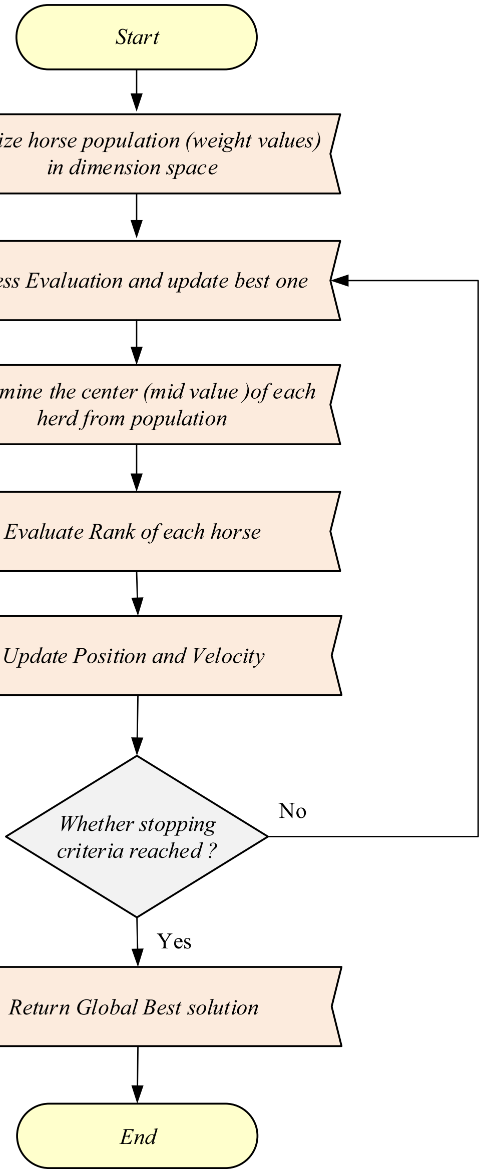

Start
Initialize horse population (weight values) in dimension space
Fitness Evaluation and update best one
Determine the center (mid value)of each herd from population
Evaluate Rank of each horse
Update Position and Velocity
Whether stopping criteria reached ?
No
Yes
Return Global Best solution
End

Fig. 4. Flow chart of HHO optimization algorithm.

The formula that is applied with respect to updating of cells in memory of matrix is given as Eq. (26)

$$Mem_{K,x,y}^{T+1} = Z_{x,y}^{T+1} * N(0, SD) \tag{26}$$

In this Eq. (26), N is a normal distribution with the mean value set as 0, and SD denotes standard variation.

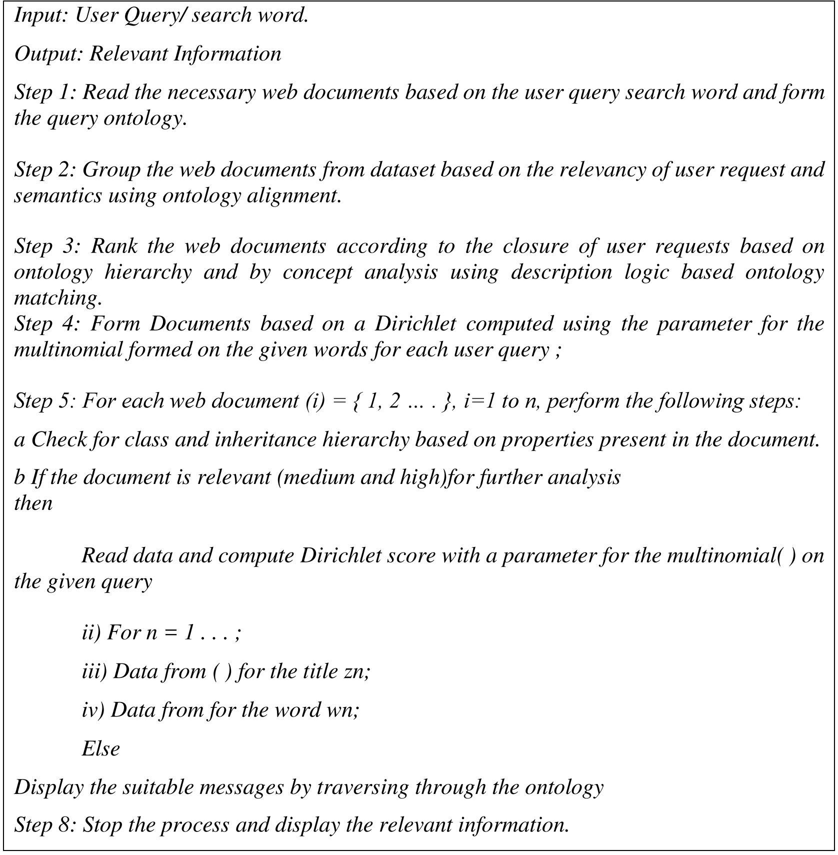
3.4. Feature extraction

A wide variety of feature descriptions may be used to represent documents. Document indexing and term weighting are two types of procedures involved in text document representation. The initial stage in document representation is to extract text documents from the collection in order to apply text preprocessing techniques to them. The token is processed by the part-of-speech tagger, which assigns a part of speech to each word. The tagger is used to tag the recovered document so that Natural Language Toolkit may execute linguistic transformations on it. Sentences are tagged with the right meanings of the words they contain to generate a training set. The POS tagging provides information about a word's semantic constituents, and the structure correlates to meaning units, allowing for semantic analysis. Word normalization entails stemming and lemmatization, but the results are a normalized version of the retrieved online content. Using semantic indexing techniques are used to identify the meaningful concepts. The documents are allocated to these concepts using cosine similarities. The techniques naturally create overlapping groups and well handle cross topic documents. In the document, the vocabulary extraction process used concept frequency-inverse document frequency $(cf-idf)$ for its representation. The concept in the documents is represented as the concept frequency document. The document (Di) represents the document vector, and each document selected is related to the cluster chosen. The weight (W_{ij}) is a function of the term frequency, collection frequency and normalization factors. Hence, the weight of each concept is calculated. The weight of concept j in the document i, T is represented in the form of Matrix A. The Word2Vec model can be calculated using the value of word vectors obtained using the Cosine Similarity equation. Cosine Similarity is the calculation of the similarity between two n-dimensional vectors by looking for a cosine value from the angle between the two and is often used to compare documents in text mining 8. Results of this equation are then recalculated using the Pearson correlation formula to compute what is the value of accuracy obtained with the selected dataset. Similarity calculations are evaluated using the Pearson correlation coefficient that has output value ranges from 0 to 1. Using the similarity equation, the values of similarity between obtained vectors are given as output. This output shows that the proposed model is enhanced more than previous methods.

3.5. KNN classifier based document classification

K-nearest neighbour is a largely used classification method, generally used for simple interpretation and its minimum calculation time. The document is classified based on the obtained vector values that are considered as features. Neighbours are taken from a set of samples that have correct classification and its Euclidean distance measured to perform classification. After obtaining feature value of K, types of distance metric with KNN classifier trained. The training phase of KNN consists of stored instances and their class labels. Consisting of m training instances of n features. The number of training instances (m) is equal to the number of samples. After training the KNN, each record of vector values is tested for detection and labelled as an output. At the end of testing, all the outputs are then classified for getting all the labels.

Algorithm 1. Pseudocode for the proposed IR design



```
Input: User Query/ search word.
Output: Relevant Information
Step 1: Read the necessary web documents based on the user query search word and form the query ontology.
Step 2: Group the web documents from dataset based on the relevancy of user request and semantics using ontology alignment.
Step 3: Rank the web documents according to the closure of user requests based on ontology hierarchy and by concept analysis using description logic based ontology matching.
Step 4: Form Documents based on a Dirichlet computed using the parameter for the multinomial formed on the given words for each user query ;
Step 5: For each web document (i) = { 1, 2 … . }, i=1 to n, perform the following steps:
a Check for class and inheritance hierarchy based on properties present in the document.
b If the document is relevant (medium and high)for further analysis
then
        Read data and compute Dirichlet score with a parameter for the multinomial( ) on the given query
        ii) For n = 1 . . . ;
        iii) Data from ( ) for the title zn;
        iv) Data from for the word wn;
        Else
Display the suitable messages by traversing through the ontology
Step 8: Stop the process and display the relevant information.
```

4. Experimental results

In the IR-Word2vec model, documents are gathered for indexing, and the semantic graphs for each document are mapped. Pre-processing has been performed to eliminate unnecessary elements from the document, after which they are mapped. Documents that have been mapped are then clustered, which transforms the document into a word2vec model. The gathered document subjects are supplied as context words in the word2vec model, and then a keyword for that word is used to transform them into vector

Table 1
Actual and predicted data of clustering techniques.

Clustering technique	Actual data			Predicted data		
	0	1	2	0	1	2
K-means	228	292	1480	409	540	1051
Agglomerative	228	292	1480	755	729	816

Table 2
Comparison of clustering techniques with accuracy, precision and specificity.

Clustering technique	Accuracy	Precision	Specificity
K-means	75	67	70
Agglomerative	62	56	61

values. Vector values are then treated as feature values, and classification is done with the help of these features using the K-nearest Neighbour (KNN) classifier. The classifications for each document are labelled at the conclusion of the categorization result. These labelled papers are then saved in a database for future access. On the other hand, the user enters a query, which is then analysed in order to locate the dominant term. The word is then compared for similarity equations, and similarity values are generated to provide output. As the class value, highly comparable results are produced. The class's corresponding clusters are chosen, and documents in that cluster are retrieved and rated based on their relevance to the user. The proposed IR-word2vec model was implemented in the Python 3.7 version software, and its performance has been analysed. Results are then compared with conventional retrieval techniques such as Lexical Retrieval Model with Semantic Residual Embeddings (LR-SRE), Document Retrieval Model through Semantic Linking (DR-MTS), Semantic Term Matching in Axiomatic Approaches to Information Retrieval (STM-IR) and Hybrid Ontology for Semantic Information Retrieval Model Using Keyword Matching Indexing System (HOS-IR). A huge data corpus is necessary to produce effective results with Word2Vec. According to the Word2Vec documentation, as the size of the data grows, so does the efficiency of the outcomes. The information was gathered from a variety of online sites that provide data and metadata for use.

4.1. Pre-processing

The process of preparing data for the technique's execution is referred to as preprocessing. The data must first be read into the proper format to prepare it for processing. After obtaining the data in the necessary format, it must be cleansed to remove any noisy or invalid information.

4.2. Clustering

The first method of preprocessing is clustering, and it is used for data exploration and grouping the objects with similar characteristics together for further processing. The K-means algorithm is a popular data clustering algorithm, and it requires a number of clusters in the data to be pre-specified.

Table 1 illustrates the actual and predicted data of clustering techniques. In this, the actual data is a manual clustered dataset, and the predicted data is clustered by the k-means and agglomerative algorithm. For class 0, 1 and 2, the actual data and predicted data is 228, 292, 1480 (k-means) 228, 292, 1480 (agglomerative) and 409, 540, 1051 (k-means) and 755, 729, 816 (agglomerative)

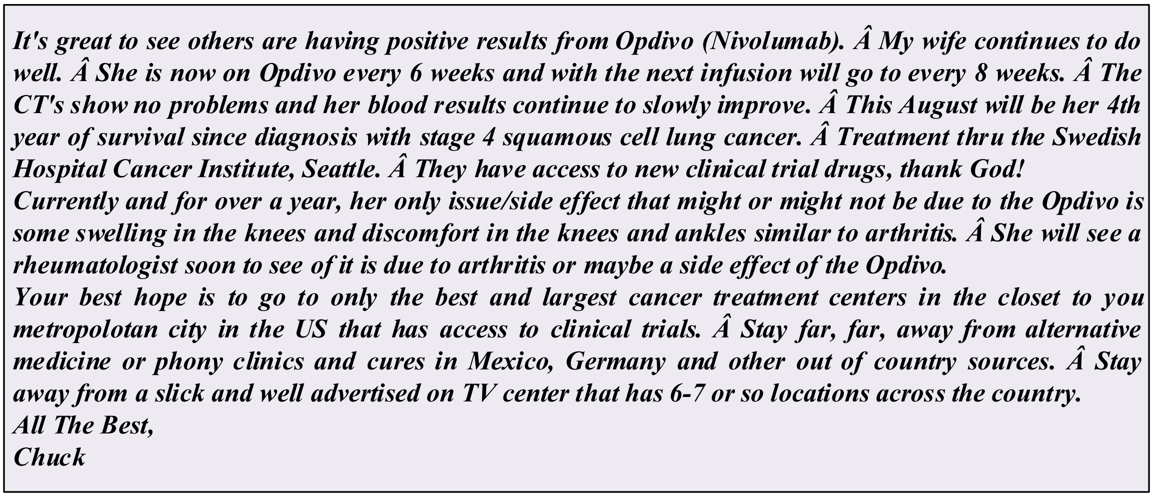
Table 2 illustrates the comparison of clustering techniques with performance metrics. In this, the accuracy, precision and specificity algorithm of K-means and agglomerative is 75, 67, 70 and 62, 56, 61. The performance metrics of the K-means algorithm is better compared to the agglomerative algorithm. In the proposed work, a large paragraph is taken as a dataset to find the similarities values, and Fig. 6 shows the sample input data.

4.3. Word2vec model

In the proposed work, a large paragraph is taken as a dataset from the clustering to find the similarities values, and the Fig. 5 shows the sample input data.

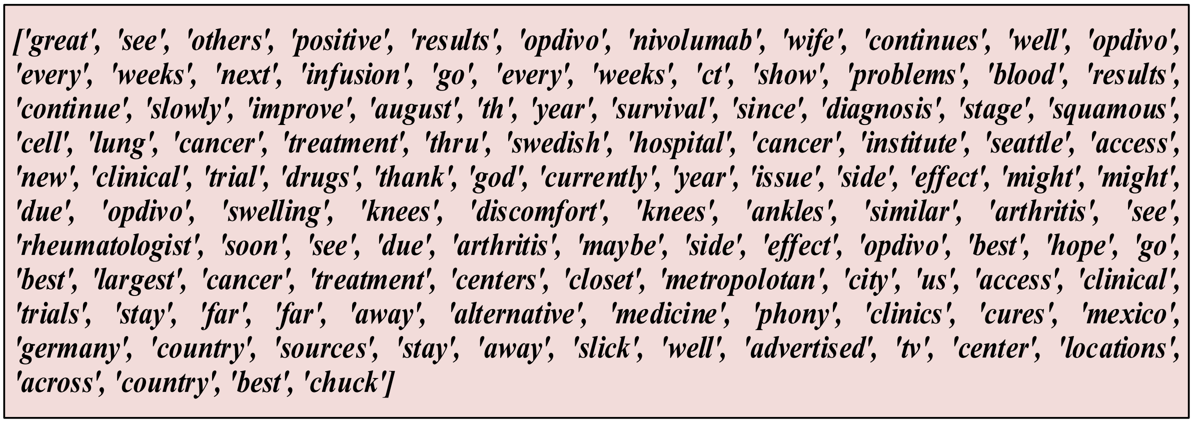
The first step of preprocessing is the conversion of given data into the lower case and the second step is the removal of punctuation. Next, the data is converted from sentence to word because words similarities are determined. In addition, stop words are removed from the data. Stop words are crucial in determining text-similarity in Word2Vec utilizing CBOW and in determining similarity in general.

The preprocessed texts were turned into a vocabulary or a bag of words model, which was then turned into a corpus. After converting a sentence to a word, the similarities of the word is checked. Initially, the window size 4 and vector size 3 is set. For example, the term 'opdivo' is searched for similarity value. As a result, it gets 10 similar words and 3 vector values because the size of the vector is set at 3. The vector values are −0.0186072, 0.00782822 and 0.170648.



It's great to see others are having positive results from Opdivo (Nivolumab). Â My wife continues to do well. Â She is now on Opdivo every 6 weeks and with the next infusion will go to every 8 weeks. Â The CT's show no problems and her blood results continue to slowly improve. Â This August will be her 4th year of survival since diagnosis with stage 4 squamous cell lung cancer. Â Treatment thru the Swedish Hospital Cancer Institute, Seattle. Â They have access to new clinical trial drugs, thank God!
Currently and for over a year, her only issue/side effect that might or might not be due to the Opdivo is some swelling in the knees and discomfort in the knees and ankles similar to arthritis. Â She will see a rheumatologist soon to see of it is due to arthritis or maybe a side effect of the Opdivo.
Your best hope is to go to only the best and largest cancer treatment centers in the closet to you metropolotan city in the US that has access to clinical trials. Â Stay far, far, away from alternative medicine or phony clinics and cures in Mexico, Germany and other out of country sources. Â Stay away from a slick and well advertised on TV center that has 6-7 or so locations across the country.
All The Best,
Chuck

Fig. 5. Sample input data.



['great', 'see', 'others', 'positive', 'results', 'opdivo', 'nivolumab', 'wife', 'continues', 'well', 'opdivo', 'every', 'weeks', 'next', 'infusion', 'go', 'every', 'weeks', 'ct', 'show', 'problems', 'blood', 'results', 'continue', 'slowly', 'improve', 'august', 'th', 'year', 'survival', 'since', 'diagnosis', 'stage', 'squamous', 'cell', 'lung', 'cancer', 'treatment', 'thru', 'swedish', 'hospital', 'cancer', 'institute', 'seattle', 'access', 'new', 'clinical', 'trial', 'drugs', 'thank', 'god', 'currently', 'year', 'issue', 'side', 'effect', 'might', 'might', 'due', 'opdivo', 'swelling', 'knees', 'discomfort', 'knees', 'ankles', 'similar', 'arthritis', 'see', 'rheumatologist', 'soon', 'see', 'due', 'arthritis', 'maybe', 'side', 'effect', 'opdivo', 'best', 'hope', 'go', 'best', 'largest', 'cancer', 'treatment', 'centers', 'closet', 'metropolotan', 'city', 'us', 'access', 'clinical', 'trials', 'stay', 'far', 'far', 'away', 'alternative', 'medicine', 'phony', 'clinics', 'cures', 'mexico', 'germany', 'country', 'sources', 'stay', 'away', 'slick', 'well', 'advertised', 'tv', 'center', 'locations', 'across', 'country', 'best', 'chuck']

Fig. 6. Sample dataset after removal of stop-word.

Table 3
Results of similarity measure from documents.

Document value	Keyword	sim_ score	vec_value of the keyword									
Doc1	gilenya	0.71	−0.004	−0.083	−0.05	0.07	0.033	0.072	0.068	0.075	−0.037	−0.005
Doc2	fingolimod	0.55	−0.070	−0.048	−0.03	−0.08	0.079	−0.048	0.084	0.052	−0.065	0.039
Doc3	ocrevus	0.42	0.054	0.083	−0.01	−0.09	0.043	0.005	0.074	−0.008	−0.026	−0.087
Doc4	cladribine	0.59	−0.081	0.044	−0.04	0.008	0.084	−0.044	0.045	−0.067	−0.035	0.094
Doc5	humira	0.76	−0.008	0.028	0.05	0.070	−0.056	0.018	0.061	−0.047	−0.031	0.068
Doc6	tagrisso	0.71	−0.080	−0.076	0.02	−0.028	−0.069	−0.081	0.083	0.02	−0.093	−0.048
Doc7	lucentis	0.65	−0.095	0.050	−0.08	−0.043	−0.0006	−0.0033	−0.075	0.096	0.049	0.091
Doc8	surgery	0.798	0.0547412	−0.07432	−0.07408	−0.02482	−0.08618	−0.01589	−0.00387	0.033243	0.014267	−0.0088
Doc9	remicade	0.667	0.0067177	−0.0382	−0.07139	−0.02088	0.039246	0.0882	0.092621	−0.05976	−0.09404	0.097606
Doc10	stelara	0.543	0.0738924	−0.01536	−0.04534	0.065381	−0.04853	−0.01819	0.028834	0.009869	−0.08293	−0.09449

The ten similar words are nivolumab, treatment, discomfort, rheumatologist, lung, infusion, clinical, god, weeks and cancer, its similarity score is 0.956, 0.950, 0.932, 0.814, 0.785, 0.775, 0.756, 0.729, 0.714 and 0.702 respectively. Table 3 shows the similarity values calculated from each document.

From Table 3, it is understandable that the documents get classified and labelled for the reason of retrieval. From the query context, a word is selected as a keyword and analysed in the documents that are shown in the first column of the table. Keywords are represented in the second column. The words that are converted to vector for comparison is given in the next column that is the third one. In the next column, the comparison results that are similar to the keywords are presented. Hence match will be found

Table 4
Keywords and their similarity value.

Keywords	Similarity value	Keywords	Similarity value	Keywords	Similarity value	Keywords	Similarity value
'gilenya'	254	'stelara'	31	'durvalumab'	11	'imfinzi'	6
'ocrevus'	248	'eylea'	28	'nivolumab'	11	'ustekinumab'	6
'ocrelizumab'	170	'avastin'	21	'osimertinib'	11	'lemtrada'	5
'entyvio'	114	'alectinib'	19	'pemetrexed'	11	'atezolizumab'	4
'humira'	112	'lucentis'	17	'ranibizumab'	11	'renflexis'	4
'opdivo'	101	'crizotinib'	17	'vitrectomy'	11	'siponimod'	4
'fingolimod'	94	'simponi'	16	'afatinib'	10	'upadacitinib'	4
'remicade'	91	'cimzia'	15	'yervoy'	10	'alecensa'	4
'cladribine'	76	'tecentriq'	14	'mavenclad'	9	'bevacizumab'	3
'tarceva'	74	'erlotinib'	13	'ozurdex'	9	'entrectinib'	3
'keytruda'	70	'vedolizumab'	12	'inflectra'	8	'gefitinib'	3
'tagrisso'	63	'tysabri'	12	'tofacitinib'	7	'ipilimumab'	3
'alimta'	55	'xalkori'	53	'dexamethasone'	6		

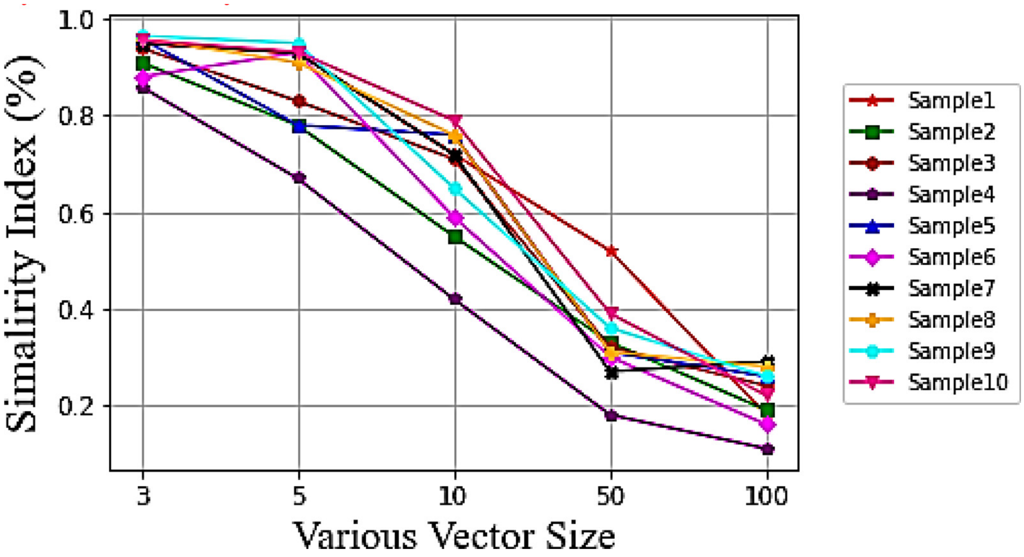


Fig. 7. Various vector dimensions and their similarity score for various words.

and retrieved with respect to the class present in the document. Values in the query keyword "gilenya" matches with the values of −0.083, −0.05, 0.07, 0.033, 0.072, 0.068, 0.075, −0.037 and −0.005.

Table 4 summarizes the Keywords and their similarity value in the document. The similarity values for different keywords vary based on the word that repeats on the document. The word 'gilenya' is repeated for 254 times in that document. Likewise, the similarity values for the other keywords are analysed.

Fig. 7 illustrates the similarity score values for different words and vector dimensions. Initially, the sample 1 word "gilenya" is analysed for similarity values such as 0.96%, 0.95%, 0.71%, 0.50% and 0.18% respectively based on the vector sizes of 3, 5, 10, 50 and 100. Sample 2 denote the word "fingolimo", whose similarity score for each vector size like 3, 5, 10, 50 and 100 is 0.90%, 0.78%, 0.57%, 0.30%, and 0.2%. Likewise, the similarity score for other samples like "tagrisso" as sample 3, "ocrevus" as sample 4, "remicade" as sample 5, "cladribine" as sample 6, "opdivo" as sample 7, "humira" as sample 8, "lucentis" as sample 9 and "surgery" as sample 10 are analysed for different vector size.

4.4. Comparison of optimization techniques

In the proposed model, the word2vec conversion is optimized using HHO (Horse Herd optimization). The convergence of the proposed HHO model is compared with the existing optimization such as PSO (Particle Swarm Optimization) and GA (Genetic algorithm). The below graphs represents the convergence graph for the three optimizations (a) convergence of HHO, (b) Convergence of GA, and (c) convergence of PSO.

Fig. 8(a) illustrates the convergence graph of the HHO. In this, the fitness value for iteration from 0–5 is 0.041244, and the rest of the fitness value for iteration from 0–98 is 0.041210 (remains constant). The Fig. 8(b) illustrates the convergence graph of the GA. In this, the fitness value for iteration from 0–25 is 0.383, and the rest of the fitness value for iteration from 0–98 is 0.369 (remains constant). The Fig. 8(c) illustrates the convergence graph of the PSO, in this the fitness value for iteration from 0–20 is

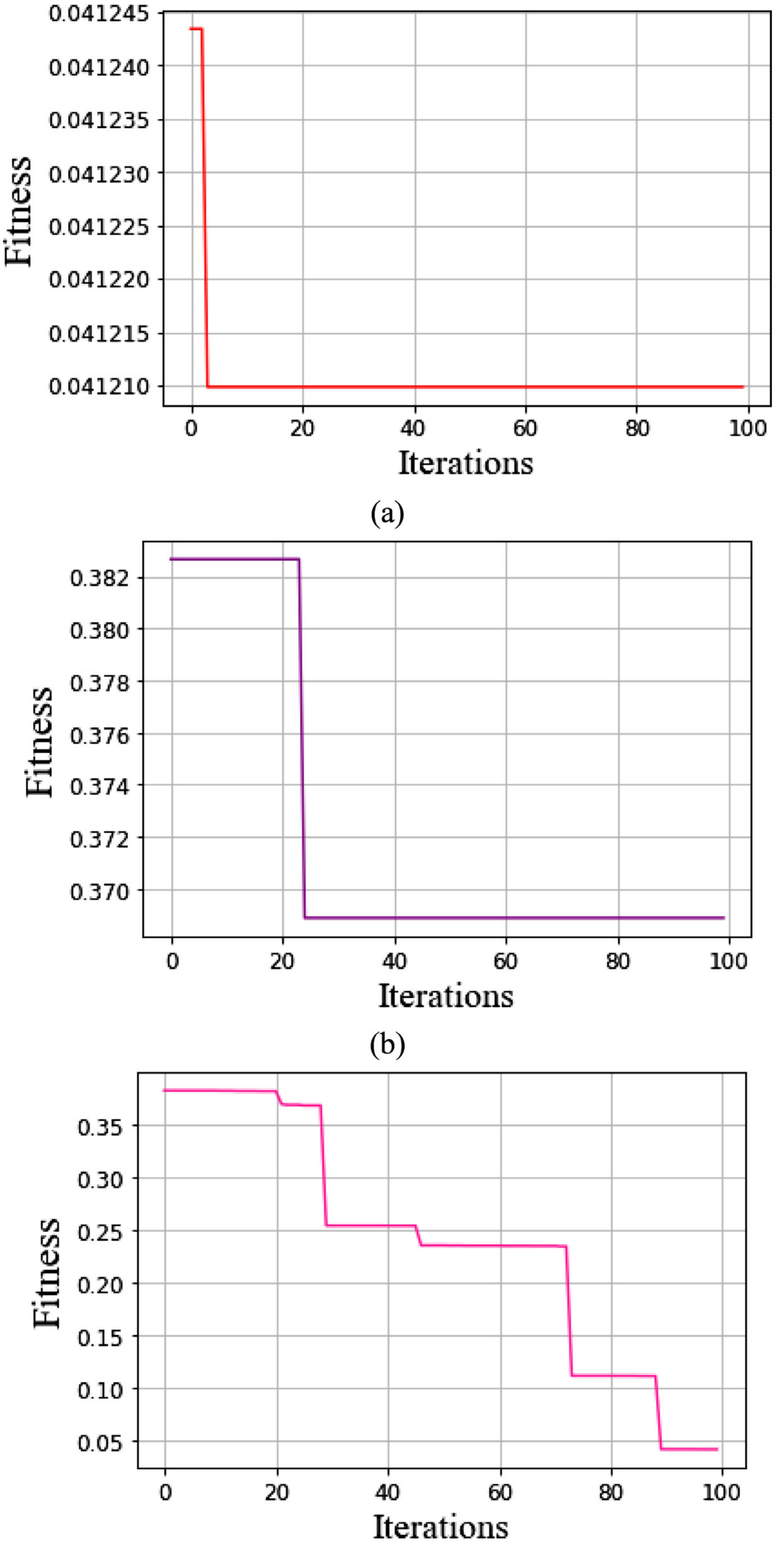


Fig. 8. (a) Convergence graph of HHO. (b) Convergence graph of GA. (c) Convergence graph of PSO.

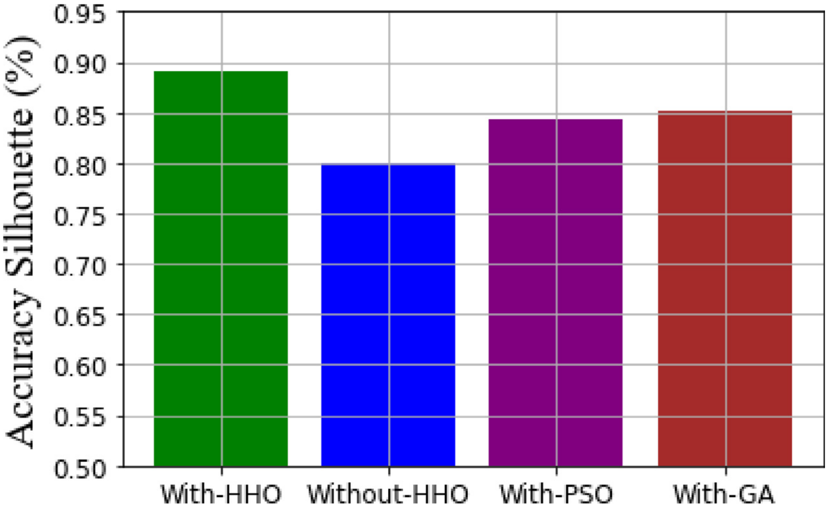


Fig. 9. Comparison of accuracy silhouette with existing techniques.

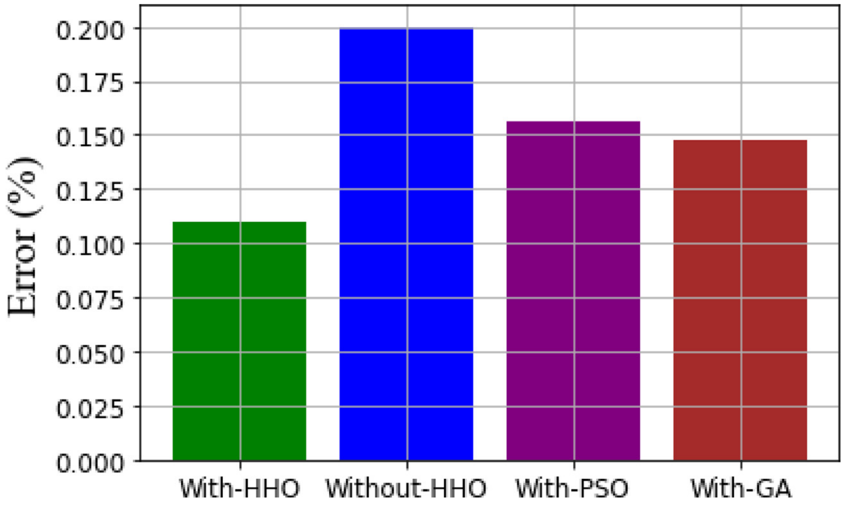


Fig. 10. Comparison of error with existing techniques.

0.38, 20–30 is 0.37, 30–45 is 0.25, 45–72 is 0.24, 72–88 is 0.11 and 88–98 is 0.04. The above three graphs determine that the fitness value of HHO is performed better compared to the existing model of GA and PSO.

Figs. 9 and 10 illustrates the comparison of accuracy silhouette and error with existing techniques, in the accuracy and error of with HHO is 0.89 and 0.110, the accuracy silhouette and error of the existing techniques such as without HHO, PSO, GA is 0.80, 0.84, 0.85 (accuracy silhouette) and 0.200, 0.155, 0.148 (error). The comparison of accuracy and error for proposed and existing techniques determines that the proposed model HHO performed better accuracy and error.

Figs. 11 and 12 illustrates the comparison of precision and specificity with existing techniques. In this, the precision and specificity of HHO is 0.70 and 0.76, the precision and specificity of the existing techniques such as without HHO, PSO, GA is 0.62, 0.65, 0.66 (precision) and 0.68, 0.73, 0.70 (specificity). The comparison of precision and specificity for proposed and existing techniques determines that the proposed model HHO performed better accuracy and error.

Table 5 illustrates the performance metrics of the proposed model and existing models. The performance metrics such as accuracy, precision, F1_score, specificity, execution time and similarity index are performed. From that, the performance of the proposed model (IR-word2vec) is better compared to the existing techniques.

Hence, the proposed method accomplishes outstanding coverage and is easily adaptable to the representation of word senses is commonly used lexical resources like WordNet, Wikipedia, and Wiktionary, without requiring additional manual work. For the

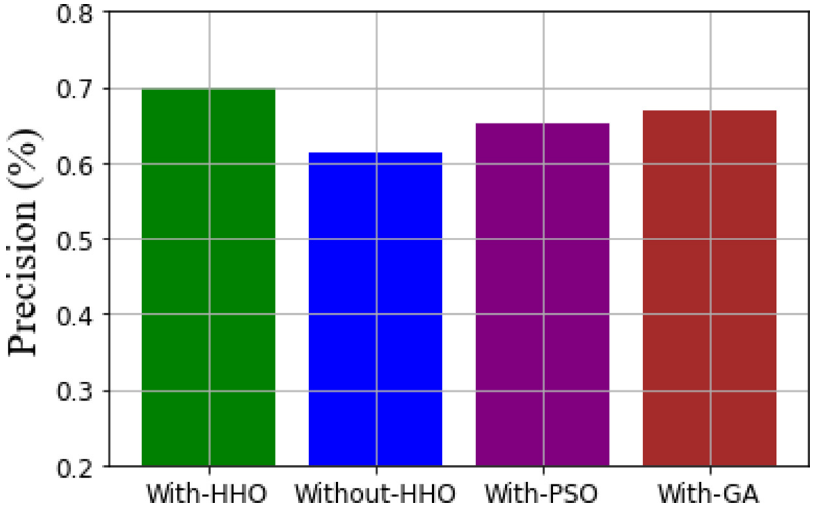


Fig. 11. Comparison of precision with existing techniques.

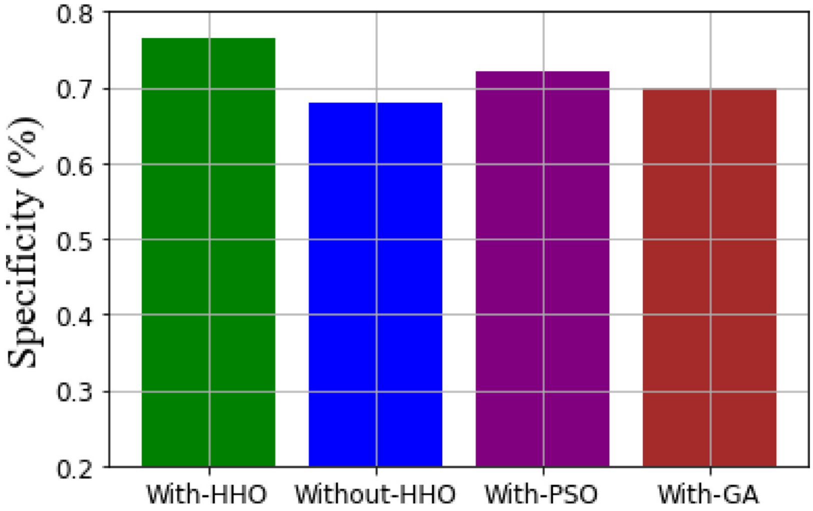


Fig. 12. Comparison of specificity with existing techniques.

best results, Word2Vec requires a large amount of data. Finally, the outcome proves that the Word2Vec model requires less time in contrast with and without the optimization process and other existing techniques.

5. Conclusion

Semantic retrieval of engineering information is essential in a variety of engineering activities such as process development and product model design. In existing techniques some of drawbacks play a vital role such as similarity scores errors between words in low vector dimensions, high execution time, poor feature extraction that ruins the performance of the model. Hence, to address these difficulties, a semantic knowledge-based retrieval system was devised. The query is entered by the user and analysed to discover the dominant term. The word is then compared for similarity equations, and similarity values are generated to provide output. As the class value, highly comparable results are produced. The appropriate clusters are chosen from the class, and documents in that cluster are retrieved and rated based on their relevance to the user. To support the accuracy level performance of this IR system, a word2vec model with the benefits of Horse Herd Optimization (HHO) was used, which aids in the extraction of vectors as features for classification. This data is saved as a .csv file for later access. The results of applying the suggested IR-word2vec model revealed that it outperforms other current approaches in terms of similarity index and accuracy for query results.

Table 5
Performance metrics comparison for IR-word2vec with existing models.

Optimization techniques	Performance metrics					
	Accuracy	Precision	F1_Score	Specificity	Execution time	Similarity index
IR-word2vec	0.85	0.66	0.56	0.75	1.6	0.96
LR-SRE	0.78	0.59	0.53	0.69	1.7	0.85
DR-MTS	0.80	0.55	0.47	0.64	1.9	0.83
STM-IR	0.73	0.51	0.45	0.69	2.1	0.75
HO-SIR	0.68	0.43	0.42	0.50	2.3	0.62

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability statement

There is no availability of data or materials available or report for the manuscript.

Funding

There is no funding provided to prepare the manuscript.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

Informal consent

Informed consent was obtained from all individual participants included in the study.

Consent to participate

I have read and I understand the provided information.

Consent to publish

This article does not contain any Image or video to get permission.

Data availability statement

There is no availability of data or materials available or report for the manuscript.

Code availability

No code is available for this manuscript.

References

[1] N.T. James, R. Kannan, A survey on information retrieval models, techniques and applications, Int. J. Adv. Res. Comput. Sci. Softw. Eng. (2017) ISSN.

[2] J. Wang, M. Pan, T. He, X. Huang, X. Wang, X. Tu, A pseudo-relevance feedback framework combining relevance matching and semantic matching for information retrieval, Inf. Process. Manage. 57 (6) (2020) 102342.

[3] J. Li, R. Jia, H. He, P. Liang, Delete, retrieve, generate: A simple approach to sentiment and style transfer, 2018, arXiv preprint arXiv:1804.06437.

[4] Y. HaCohen-Kerner, D. Miller, Y. Yigal, The influence of preprocessing on text classification using a bag-of-words representation, PLoS One 15 (5) (2020) e0232525.

[5] E.L. Pontes, S. Huet, A.C. Linhares, J.M. Torres-Moreno, Predicting the semantic textual similarity with siamese CNN and LSTM, 2018, arXiv preprint arXiv:1810.10641.

[6] M.K. Nammous, K. Saeed, Natural language processing: Speaker, language, and gender identification with LSTM, in: Advanced Computing and Systems for Security, Springer, Singapore, 2019, pp. 143–156.

[7] I. Budiman, D.T. Nugrahadi, M.R. Faisa, M. Rusli, A Study on Effect of Generated Features From Word2Vec Vectors For Text Classification.

[8] T. Thongtan, T. Phienthrakul, Sentiment classification using document embeddings trained with cosine similarity, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 2019, pp. 407–414.

[9] M.A. Raza, R. Mokhtar, N. Ahmad, M. Pasha, U. Pasha, A taxonomy and survey of semantic approaches for query expansion, IEEE Access 7 (2019) 17823–17833.

[10] A. Berger, J. Lafferty, Information retrieval as statistical translation, in: ACM SIGIR Forum, Vol. 51, ACM, New York, NY, USA, 2017, pp. 219–226, (2).

[11] L. Abualigah, A.H. Gandomi, M.A. Elaziz, H.A. Hamad, M. Omari, M. Alshinwan, A.M. Khasawneh, Advances in meta-heuristic optimization algorithms in big data text clustering, Electronics 10 (2) (2021) 101.

[12] M. Frasca, G. Tortora, Visualizing correlations among Parkinson biomedical data through information retrieval and machine learning techniques, Multimedia Tools Appl. (2021) 1–19.

[13] I. Khennak, H. Drias, An accelerated PSO for query expansion in web information retrieval: application to medical dataset, Appl. Intell. 47 (3) (2017) 793–808.

[14] H. Wang, Q. Zhang, J. Yuan, Semantically enhanced medical information retrieval system: a tensor factorization based approach, IEEE Access 5 (2017) 7584–7593.

[15] Y. Djenouri, A. Belhadi, R. Belkebir, Bees swarm optimization guided by data mining techniques for document information retrieval, Expert Syst. Appl. 94 (2018) 126–136.

[16] K. Lee, J. Lee, M.P. Kwan, Location-based service using ontology-based semantic queries: A study with a focus on indoor activities in a university context, Comput. Environ. Urban Syst. 62 (2017) 41–52.

[17] F. Li, L. Liao, L. Zhang, X. Zhu, B. Zhang, Z. Wang, An efficient approach for measuring semantic similarity combining WordNet and Wikipedia, IEEE Access 8 (2020) 184318-184338.

[18] N.H. Mahadzir, M.F. Omar, M.N.M. Nawi, Semantic similarity measures for Malay-English ambiguous words, J. Telecommun. Electron. Comput. Eng. (JTEC) 10 (1–11) (2018) 109–112.

[19] O. Araque, G. Zhu, C.A. Iglesias, A semantic similarity-based perspective of affect lexicons for sentiment analysis, Knowl.-Based Syst. 165 (2019) 346–359.

[20] J. Zhang, M. Chen, E. Hu, L. Wu, Data mining model for food safety incidents based on structural analysis and semantic similarity, J. Ambient Intell. Humaniz. Comput. (2020) 1–15.

[21] A. Gomathi, J. Jayapriya, G. Nishanthi, K. Pranav, G.P. Kumar, Ontology based semantic information retrieval using particle swarm optimization, Int. J. Appl. Inf. Commun. Eng. 1 (4) (2015) 5–8.

[22] R.Z. Al-Abdallah, A.T. Al-Taani, Arabic text summarization using firefly algorithm, in: 2019 Amity International Conference on Artificial Intelligence, AICAI, IEEE, 2019, pp. 61–65.

[23] R.T. Jose, S.L. Poulose, Semantic Web Query Join Optimization Using Modified Grey Wolf Optimization Algorithm.

[24] H. Kusniyati, A.A. Nugraha, Analysis of matric product matching between cosine similarity with term frequency-inverse document frequency (TF-IDF) and Word2Vec in PT. Pricebook digital Indonesia, Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. 6 (1) (2020) 105–112.

[25] P. Sitikhu, K. Pahi, P. Thapa, S. Shakya, A comparison of semantic similarity methods for maximum human interpretability, in: 2019 Artificial Intelligence for Transforming Business and Society, Vol. 1, AITB, IEEE, 2019, pp. 1–4.

[26] D. Bollegala, Y. Matsuo, M. Ishizuka, A web search engine-based approach to measure semantic similarity between words, IEEE Trans. Knowl. Data Eng. 23 (7) (2010) 977–990.

[27] S. Wang, R. Koopman, Semantic embedding for information retrieval, in: BIR@ ECIR, 2017, pp. 122–132.

[28] J. Chen, Y. Saad, Divide and conquer strategies for effective information retrieval, in: Proceedings of the 2009 SIAM International Conference on Data Mining, Society for Industrial and Applied Mathematics, 2009, pp. 449–460.

Anil Sharma holds master and bachelor degree in Computer Science & Engineering. Currently, he is pursuing Ph.D. from GGSIPU Delhi. He has an experience of 10 years in academics and research. He has published and presented 22 research papers in reputed international journals and conferences. He has authored one book on Management Information System and holds one Indian patent on IoT. He is a member and reviewer for professional body Internet Society Delhi Chapter. His research interest includes information retrieval, semantic web, natural language processing, machine learning and rough set theory.

Suresh Kumar holds Ph.D. and master degree in Computer Science & Engineering. He is currently working as an Associate Professor with Department of Computer Engineering, Netaji Subhas University of Technology East Campus, Delhi. He has published and presented more than 50 research papers in reputed international journals and conferences. He has served as session chairs for many international conferences and delivered expert talks in Faculty Development Programs. He has been on panel on many expert committees for technical consultancy. His research areas include semantic web, machine learning, information retrieval, information systems and natural language processing.