International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-9 Issue-6, April 2020

# Semantic Similarity Analysis on Knowledge Based and Prediction Based Models

**Nisha Varghese, M Punithavalli**

***Abstract:*** ***The similarity between two synsets or concepts is a numeral measure of the degree to which the two objects are alike or not and the similarity measures say the degree of closeness between two synsets or concepts. The similarity or dissimilarity represented by the term proximity. Proximity measures are defined to have values in the interval [0, 1]. Term Similarity, Sentence similarity and Document similarity are the areas of text similarity. Term similarity measures used to measure the similarity between individual tokens and words, Sentence similarity is the similarity between two or more sentences and Document similarity used to measure the similarity between two or more corpora. This paper is the study between Knowledge based, Distribution based and prediction based semantic models and shows how knowledge based methods capturing information and prediction based methods preserving semantic information.***

***Keywords: Path similarity, LIN, LCH, JCN, WUP, RES, PPMI, LSA, Word2vec.***

## I. INTRODUCTION

The degree of proximity between two entities or tokens is estimated by similarity measures and which helps to identify similar entities and detect how the entities are different from each other. Various types of scoring or ranking algorithms have also been developed based on the distance or similarity measures. Semantic measures are used to estimate the strength of the semantic relationship between units of language, concepts or instances, through a numerical value obtained as the proximity according to the comparison of information supporting their meaning [1]. Semantic relatedness is the strength of the semantic interactions between two elements with no restrictions on the types of the semantic links considered and Semantic similarity is the subset of the notion of semantic relatedness only considering taxonomic relationships in the evaluation of the semantic interaction between two elements [2]. Similarity and distance are inversely proportional, that is greater the distance smaller the similarity. Text similarities are divided into various types such as Morphological similarity, Spelling Similarity, Synonymy, Homophony, semantic similarity, Sentence similarity, Document Similarity and Cross-lingual Similarity.

**Revised Manuscript Received on April 30, 2020.**
***** Correspondence Author
**Nisha Varghese***, Department of Computer Applications, Bharathiar University, Coimbatore, Email: nisha.varghes@gmail.com
**M Punithavalli**, Department of Computer Applications, Bharathiar University, Coimbatore, Email: punithavalli@buc.edu.in



Semantic relations are holds between word senses and not between words. Lexical semantics is the meaning of words and Compositional semantics is the meaning of sentences. Semantic measures are used to compare semantic entities such as tokens, sentences and even corpora. The key issue is that meaning is a multifaceted concept and thus there are multiple axes, along with which two words can be similar[3].

## II. TYPES OF TEXT SIMILARITIES

Text similarities are in various types such as Morphological Similarity, Spelling Similarity, Synonymy Similarity, Semantic Similarity, Sentence Similarity, Document Similarity and Cross-Lingual Similarity. Morphological similarity means the morphological changes of words with respect to root word (beauty-beautiful). Spelling similarity is the changes in spell but carries different meanings (pear-pair). Synonymy is the phenomenon with a word or phrase that means exactly or nearly the same as another word or phrase in the same language (close-shut). Homonyms are words with similar sound or pronunciation but have different meanings. Homophones are a type of homonym that also sound alike and have different meanings, but have different spellings (raise-rays-raze). Homographs are the words with same spellings, but have different meanings. Heteronyms similar to homographs with same spelling but have different meanings, but sound different. Hyponyms and Hypernyms are the relationship between word meanings, words belong to a node with a single parent node. The parent node is the hypernym and the child nodes are the hypernyms. Meronyms and holonyms also represent the relationship between word meanings to describe the part and whole relationships. Polysemy means a single word having different meanings depending on the Context. Semantic similarity means the similarity in the meaning of words (cat-tabby). Sentence and Document similarities are the similarities between the sentences and documents respectively. Cross-lingual Similarity is the changes in words with respect to the languages (Japan-Nihon) [4].

In WordNet, a lexical reference system, a word has an average of 1.4 senses [5]. Verbs have the highest average senses per word (2.1), adjectives (1.45), Adverbs (1.25) and nouns (1.24). Verbs often change their meaning depending on the noun [6]. The word break has a higher number of senses with 74 senses followed by cut with 73. The words out, round, still and down has a higher number of Parts of Speech with 5. A word such as lot has 27 synonyms. WordNet is the semantic dictionary of English words with interlinked semantic relations.

WordNet has an average of 1.75 words are used to express a single meaning. WordNet organizes information in the form of a hierarchy with a dummy root.

Verbs, nouns and adjectives all have separate hierarchies. The relationship between words and meanings is many to many. Polysemy and synonymy make the problem of understanding language more complex than an assembly of words that have unique meanings. Computers are extremely logical and expressions of humor or sarcasm are hard to detect automatically [6]. Table 1.1 listed some words and the WordNet senses.

**TABLE I. WORD SENSES**

| Words | Senses |
|---|---|
| Good | commodity, full, estimable, beneficial, adept, dear, dependable, effective, well, thoroughly |
| Out | come_out_of_the_closet, extinct, forbidden, knocked_out, away |
| Round | cycle, beat, round_of_golf, turn, rung, circle, attack, polish, round_off, orotund |
| Still | hush, distillery, calm, inactive, silent, placid, however, even |
| down | toss_off, devour, polish, depressed, gloomy |
| light | luminosity, sparkle, inner_light, lighter, light_up, alight, ignite, fall,unhorse, unaccented, faint, abstemious, idle, easy, lightly |
| bear | give_birth, digest, hold, yield, wear, behave, have_a_bun_in_the_oven |

## III. SEMANTIC SIMILARITY APPROACHES

Semantic Similarity approaches are classified into knowledge based model and corpus based models.

### A.Knowledge Based Models

#### 1. Path Similarity

Path similarity finds the shortest path between two synsets or concepts. The resultant score in discrete and not normalized, with no weights assigned on edges.

$$path_{similarity(c1,c2)} = no.of\ edges\ in\ the\ shortest\ path \quad (1)$$

#### 2. LEAKCOCK-CHODOROW(LCH) Similarity

LCH similarity is similar to path similarity with continuous score, denoting count of edges between two words/senses with negative log smoothing.

$$lch_{similarity} = -log\frac{spath(synset1, synset2)}{2 * path} \quad (2)$$

#### 3. WU & PALMER(WUP Similarity)

WUP Similarity is a score that takes into account the position of concepts or synsets c1 and c2 in the taxonomy relative to the position of the Least Common Subsumer (c1, c2). LCS finds the closest ancestor to both concepts.

$$LCS(c1, c2) = Lower\ node\ in\ hierarchy\ that\ is\ a\ hypernym\ of\ c1, c2 \quad (3)$$

WUP similarity between two concepts is the function of path length and depth, in path-based measures. The similarity score is normalized and never be zero. It is heavily dependent on the quality of a graph and no distinction between similarity and relatedness.

$$WUP_{Similarity(c1,c2)} = \frac{2 * Dep\big(LCS(c1, c2)\big)}{Len(c1, c2) + 2 * \big(LCS(c1, c2)\big)} \quad (4)$$

#### 4. RESNIK SIMILARITY(RES Similarity)

RES Similarity [7] is based on the Information Content (IC) of the Least Common Subsumer. The information content is the frequency counts of concepts or tokens as found in a corpus of text or in the dataset and computed for nouns and verbs in WordNet.

$$IC(c) = -logP(c) \quad (5)$$

$$RES_{Similarity(c1,c2)} = IC\big(LCS(c1, c2)\big) \quad (6)$$

The frequency of a concept is incremented in WordNet each time, when the concept and ancestor concepts in the hierarchy. The score is non-negative and normalized. IC-based scores provide more accuracy than path based measures.

#### 5. Dekang Lin method(LIN Similarity)

LIN Similarity [8] between c1 and c2 needs to do more than measure common information and based on the information contained in the LCS of the two concepts. If more difference between c1 and c2 then less similar they are. Commonality shows more information between c1 and c2 and more difference shows less similar between c1 and c2.

$$LIN_{Similarity} = 2 \times \frac{logp\big(LCS(c1, c2)\big)}{logp(u) + logp(v)} \quad (7)$$

$$Commonality = IC\big(Common(c1, c2)\big) \quad (8)$$

$$Difference = IC(Descreption(c1, c2) - IC\big(common(c1, c2)\big) \quad (9)$$

#### 6. JIANG-CONRATH DISTANCE(JCN Similarity)

JCN Similarity is similar to LIN Similarity, which is inversely proportional to the JCN Distance. The resultant score says the amount of information needed to state the commonality between the two concepts or synsets.

$$JCN_{Similarity} = \frac{1}{JCN_{Distance(c1,c2)}} \quad (10)$$

### B. Corpus Based Models_Distributed Semantic Models

$$JCN_{Distance(c1,c2)} = 2 * logP\big(LCS(c1, c2)\big) - \big(logP(c1) + logP(c2)\big) \quad (11)$$

#### 1. PPMI (Positive Pointwise Mutual Information)

Pointwise Mutual Information or Point Mutual Information (PMI) is a measure of the information overlap between two random variables which means the association between a feature (term) and a class(the window the corpus vocabulary)[4,5]. The range of the PMI is -∞ to ∞.

$$pmi(w; c) \equiv log\frac{p(w, c)}{p(w)p(c)} = log\frac{p(w|c)}{p(w)} = log\frac{p(c|w)}{p(c)} \quad (12)$$

When x and y are perfectly correlated the P(x|y)=P(y|x)=1. Here will be the same with contexts and words.

$$PMI(w, c) = log\frac{p(c|w)}{p(c)} = log\frac{count(w, c) * N}{count(c) * count(w)} \quad (13)$$

The text normalization and stop words removal affect the accuracy of these values, but in the above sentence the stop words are not handled properly and then they may be a problem of the relatively high frequency. PMI is biased towards the infrequent values and the negative words are also problematic.

To overcome this problem we can choose the Positive Pointwise Mutual Information (PPMI) considers the relative occurrences with the size of the vocabulary.

PPMI can provide semantic information about the context word is particularly informative about the target word [9]. In PPMI all negative values are replaced by 0 [10].

$$PPMI(w,c)=MAX(pmi(w,c),0) \quad (14)$$

But even with these methods, it is unable to solve the high dimensionality problem (|v|=20000-50000), sparse nature (most of the elements are zero) and redundancy.

### 1. Latent Semantic Analysis LSA

Latent Semantic Analysis (LSA) [11] is the powerful and widely used dimensionality reduction technique and it is a variation of Singular Valued Decomposition, so it also termed as Truncated SVD. In LSA it keeps the top k singular values instead of all dimensions. LSA preserves the semantics of the sentences and corpus as the sum of the meaning of all words occurring in it and assumes that the semantic associations between words. (LSA is also known as LSI (Latent Semantic Index) learns latent topics by performing a matrix decomposition (SVD) on the term-document matrix In practice, LSI or LSA is much faster to train than LDA ( latent Dirichlet allocation), but has lower accuracy but LDA computationally expensive on large data sets[12]. LSA is unable to capture the polysemes of words. When LSA is applied to texts for calculating the similarity, the vector representation of each text is transformed into a reduced dimensional space, while the similarity between two texts is obtained from calculating the two vectors of the reduced dimension [13] . Latent Semantic Analysis (LSA) is a variation of Singular value decomposition (SVD).

SVD is a matrix decomposition method which factorizes a matrix into the product of three matrices, the decomposition is possible for any number of $w\times c$ [14] $X(w\times c)=W(w\times m).S(m\times m).C(m\times c)$ is of the form. The diagonal values in the Sigma matrix S are diagonal m×m matrix of singular values of the original matrix X. The columns of the W are the left-singular vectors of X, These m column vectors of W are orthogonal to each other which represent a dimension in new latent space and ordered by the amount of variance in the dataset and the columns of C are the right singular vectors of X.

$$\begin{bmatrix} \uparrow & \uparrow & \uparrow \\ x_1 & \ldots. & x_c \\ \downarrow & \downarrow & \downarrow \end{bmatrix}_{w\times c} = \begin{bmatrix} \uparrow & \uparrow & \uparrow \\ w_1 & \ldots. & w_m \\ \downarrow & \downarrow & \downarrow \end{bmatrix}_{w\times m} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \end{bmatrix}_{m\times m} \begin{bmatrix} \leftarrow & c_1 & \rightarrow \\ & \vdots & \\ \leftarrow & c_m & \rightarrow \end{bmatrix}_{m\times c} \quad (15)$$

In LSA it keeps the top k singular values instead of all dimensions. So (4) will be changed like (5).

$$\begin{bmatrix} \uparrow & \uparrow & \uparrow \\ x_1 & \ldots. & x_c \\ \downarrow & \downarrow & \downarrow \end{bmatrix}_{w\times c} = \begin{bmatrix} \uparrow & \uparrow & \uparrow \\ w_1 & \ldots. & w_k \\ \downarrow & \downarrow & \downarrow \end{bmatrix}_{w\times k} \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix}_{k\times k} \begin{bmatrix} \leftarrow & c_1 & \rightarrow \\ & \vdots & \\ \leftarrow & c_k & \rightarrow \end{bmatrix}_{k\times c} \quad (16)$$

For example the SVD evaluation of the matrix X,

$$\begin{bmatrix} 5 & 2 & 1 \\ 3 & 8 & 4 \\ 7 & 5 & 1 \end{bmatrix} = \begin{bmatrix} -0.376 & 0.487 & -0.788 \\ -0.671 & -0.73 & -0.131 \\ -0.639 & 0.48 & 0.602 \end{bmatrix} \begin{bmatrix} 13.056 & & \\ & 4.78 & \\ & & 0.817 \end{bmatrix} \begin{bmatrix} -0.641 & -0.714 & -0.283 \\ 0.754 & -0.515 & -0.408 \\ -0.145 & 0.475 & -0.86 \end{bmatrix}$$

SVD can improve the retrieval performance of a search engine. The accuracy of the approximation is depending on the dimensions. If a lower number of dimensions the accuracy will be low, but with the higher number the approximation will be closed to the original matrix. When using an optimal number of dimensions, it is possible to capture keyword relationships. The largest advantage of the indexing method is solving the problem of synonymy.

### C.Prediction Based Models_Word Embeddings

### 1. Word2Vec

One of the important prediction based embedding is Word2vec, which is a two-layered neural network to generate word embeddings for a large corpus and results in a high dimensional vector space of data. It allows many operations using these word vectors such as add, subtract, and distance calculation and these operations help to preserve the relationships among the words. A word vector representation is associated with n-grams and the words represented as the sum of the representations [15]. Word2vec contains two architectures to produce for distributed word representations: Continous Bag-of-Words (CBOW) [16]. and Continous Skip-gram.

- **Continuous bag of words model**

The CBOW model predicts the current word from a window of surrounding context words without considering the order of the context words. The context may contain a single or multiple words. Skip gram weighs the adjacent context words strongly than the distant context words, that is predicts the context words from the target word [12,16].

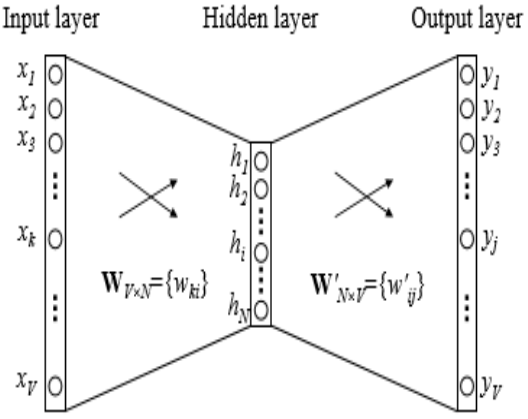



**Fig.1. Single context words of CBOW**

The fig.1. is a neural network that contains three layers: an input layer, a hidden layer and an output softmax layer. The softmax layer is used to sum the probabilities obtained in the output layer to 1. To calculate the hidden layer activation forward propagation is used and there is no other activation function between any layers.

The input layer and the target layer are one hot encoded in the form of [1×V]. One set of weights assigned between the input and hidden layer and the one between the hidden and softmax layer.

The hidden activation is the product of input and the input-hidden weights and the hidden input gets multiplied by hidden- output weights and output is calculated. Backpropagation is used to readjust the weights. The word vector representation of the word is taken as the weight between the hidden layer and the output layer.

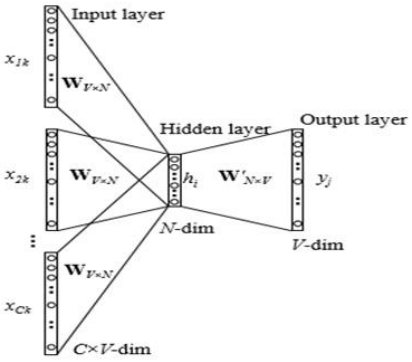



**Fig.2. Multiple context words of CBOW**

In fig.2. shows the CBOW of multiple context words and predicts the probability of a target word and it takes multiple one-hot encoded vectors in the input layer.

The advantages of CBOW are, It executes faster for a small dataset, it is probabilistic works efficiently for deterministic methods, does not require huge RAM as compared with the co-occurrence matrices we discussed earlier in this paper. But it takes the average of context words that may affect the semantic meaning as shown in fig.4 and fig.5. Improper training will not optimize better.

- **Skip-Gram**

Skip-gram predicts the context words from the target word and it weighs the adjacent context words strongly than the distant context words [12,16]. Skip gram is similar to CBOW but the difference will be in the target variable, which is there will be more than one hot encoded target variables and outputs.

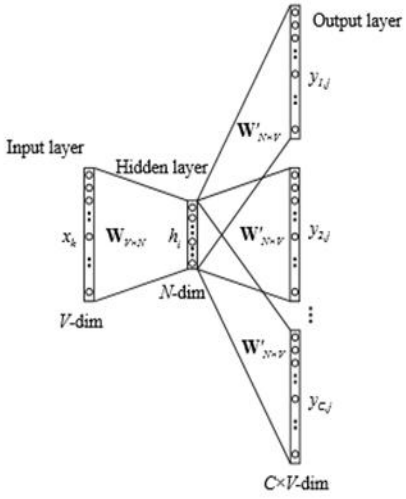



**Fig.3. Skip-gram model**

High dimensionality, the large training set and increasing the window size are helpful to improve the accuracy of vector representations because increased dimensionality can preserve more information, increase the training set and increasing the window size are preserve more semantic information, but both cause difficulty in training.

## IV. RESULTS AND DISCUSSIONS

Fig.4, Fig.5, and Fig.6 show the result of the various similarity measures with synsets of the WordNet interface.


**Fig.4. Path Similarity and LCH Similarity**

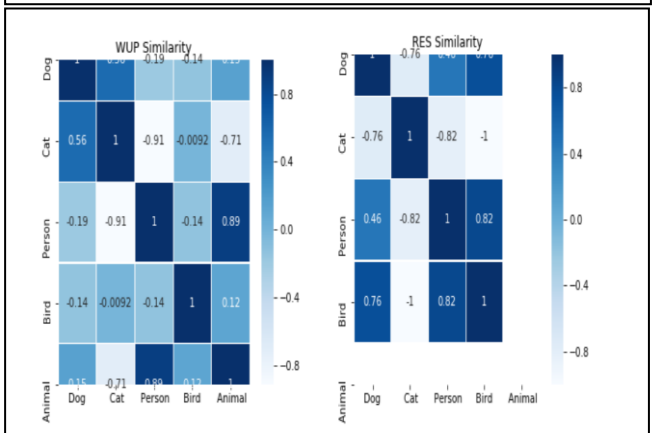



**Fig.5. WUP Similarity and RES Similarity**

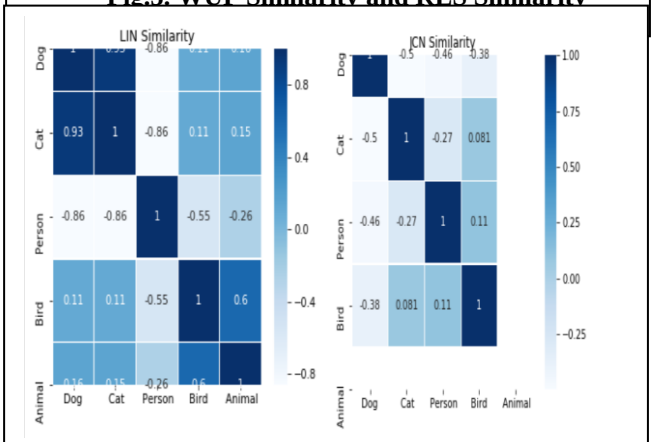
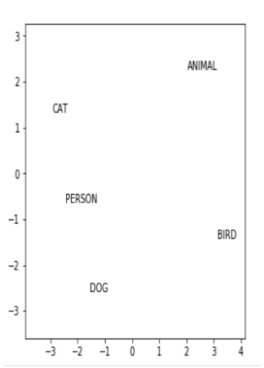



**Fig.6. LIN Similarity and JCN Similarity**

Fig.4 shows Path similarity and LEAKCOCK-CHODOROW(LCH) Similarity, both results shows approximately the same results for tokens. WUP Similarity is also showing same as Path and LCH Similarity. RES Similarity, LIN Similarity and JCN Similarity show various types of similarity measures using synsets. Fig.7, Fig.8 and Fig.9 show the word vector representations of various. These are the results from the prediction of CBoW training with window size 2 and 3, with softmax, loss function calculated by cross-entropy and loss minimization by Gradient Descent optimizer. In Fig.9, the semantic meaning is not preserved properly, but in Fig.8 the semantics captured effectively with respect to the window size 3 or Trigram.

Increased training data set can improve the semantics between the vectors.



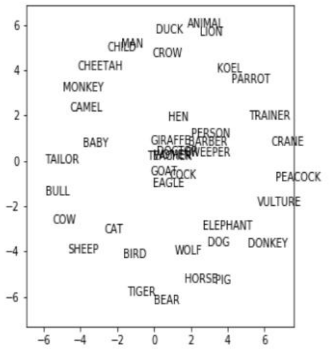




**Fig.7. Vector Representation** **Fig.8. Trigram Representation**

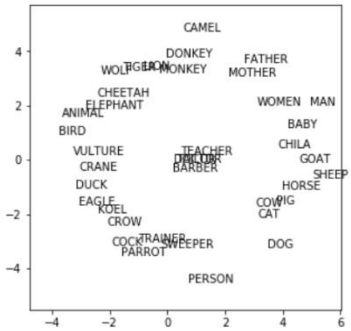



**Fig.9. Bigram Representation.**

## V. CONCLUSION

This paper focused on various types of Semantic Similarity measures based on knowledge and dense vectors. The corpus based dense vector representation models show that the accuracy depends on the window size and context and the accuracy can improve to the size of training data set, dimensionality and sub-sampling. Techniques like GloVe[17], Fast Text[18] and BERT(Bidirectional Transformers for Language Understanding) [19] can also use for the efficient representation of the Word Vector Representation as a future enhancement.

## REFERENCES

1. Sébastien Harispe, Semantic Similarity from Natural Language and Ontology analysis, arXiv:1704.05295v1 [cs.AI] 18 Apr 2017
2. Sébastien Harispe, Knowledge-based Semantic Measures: From Theory to Applications. Computer Science [cs]. Université de Montpellier, 2014.
3. RyanCotterell et al, MorphologicalWord-Embeddings, Human Language Technologies: The 2015 Annual Conference of the North American Chapter of the ACL, pages 1287–1292, Denver, Colorado, May 31 – June 5, Association for Computational Lingusitcs.
4. IvanVuli´c and Marie-FrancineMoens, Cross-Lingual Semantic Similarity of Words as the Similarity of Their Semantic Word Responses, Proceedings of NAACL-HLT 2013, pages 106–116, Atlanta, Georgia, 9–14 June 2013
5. K. Miller et al, “Five Papers on Wordnet”, Technical report, Prinston University, Prinston, 1993
6. Manu konchady, Text Mining Application Programming, Cengage Learning, 2009
7. Philip Resnik, Using Information Content to Evaluate Semantic Similarity in a Taxonomy, Sun Microsystems Laboratories, COGNITIVE MODELLING.
8. Dekang Lin, An information-theoretic definition of similarity, ICML, 1998
9. John A Bullinaria and Joseph P Levy. “Extracting semantic representations from word co-occurrence statistics: a computational study”. Behavior Research Methods, 39(3):510–526, 2007.
10. OmerLevy, “NeuralWordEmbedding asImplicitMatrixFactorization”, NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2,December 2014 pp 2177–2185
11. Scott Deerwester et al, “Indexing by Latent Semantic Analysis” JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE, 1990
12. Tomas Mikolov et al, “Efficient Estimation of Word Representations in Vector Space”, arXiv:1301.3781 [cs. Computation and Language], 2013
13. MingCheLee et al, A Grammar-Based Semantic Similarity Algorithm for Natural Language Sentences, The Scientific World Journal Volume 2014, Article ID 437162
14. G. H. Golub, “Linear Algebra Singular Value Decomposition and Least Squares Solutions”, Numer. Math. 14, 1970, pp403--420
15. Piotr Bojanowski et al, “Enriching Word Vectors with Subword Information”, arXiv:1607.04606 [cs.CL],2016
16. Thomas Mikolov et al, “Distributed Representations of Words and Phrases and their Compositionality”, arXiv:1310.4546 [cs.CL], 2013
17. Jeffrey Pennington, Richard Socher, Christopher D. Manning, “GloVe: Global Vectors for Word Representation”, Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014
18. Armand Joulin, “Bag of Tricks for Efficient Text Classification”, 2016, arXiv:1607.01759 [cs.CL].
19. Jacob Devlin, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”, 2018, arXiv:1810.04805 [cs.CL].

## AUTHORS PROFILE

**Nisha Varghese,** is a full time research scholar in the Department of Master of Computer Application in Bharathiar University, Tamilnadu. She received MSc in Computer Science from Calicut University, Kerala, India and Completed Mhil. In Computer Science, from Bharathiar University, Coimbatore, Tamil Nadu, India. Her research interests including Data Mining, Machine Learning, Text Data Analytics, Information Retrieval and Natural Language Processing. E-mail: nisha.varghes@gmail.com.

**M Punithavalli** is working as a Professor in the Department of Computer Applications, Bharathiar University, Coimbatore, Tamilnadu. India. She has 25 years of research and academic experience. She has published more than 80 research paper articles in international journals and reputed journals. She has authored three books. Her areas of interest are Software Engineering, Data mining and Machine Learning. Email: punithavalli@buc.edu.in