



Impact of word embedding models on text analytics in deep learning environment: a review

Deepak Suresh Asudani¹ · Naresh Kumar Nagwani¹ · Pradeep Singh¹

Accepted: 1 February 2023 / Published online: 22 February 2023
© The Author(s), under exclusive licence to Springer Nature B.V. 2023

Abstract

The selection of word embedding and deep learning models for better outcomes is vital. Word embeddings are an n-dimensional distributed representation of a text that attempts to capture the meanings of the words. Deep learning models utilize multiple computing layers to learn hierarchical representations of data. The word embedding technique represented by deep learning has received much attention. It is used in various natural language processing (NLP) applications, such as text classification, sentiment analysis, named entity recognition, topic modeling, etc. This paper reviews the representative methods of the most prominent word embedding and deep learning models. It presents an overview of recent research trends in NLP and a detailed understanding of how to use these models to achieve efficient results on text analytics tasks. The review summarizes, contrasts, and compares numerous word embedding and deep learning models and includes a list of prominent datasets, tools, APIs, and popular publications. A reference for selecting a suitable word embedding and deep learning approach is presented based on a comparative analysis of different techniques to perform text analytics tasks. This paper can serve as a quick reference for learning the basics, benefits, and challenges of various word representation approaches and deep learning models, with their application to text analytics and a future outlook on research. It can be concluded from the findings of this study that domain-specific word embedding and the long short term memory model can be employed to improve overall text analytics task performance.

Keywords Word embedding · Natural language processing · Deep learning · Text analytics

1 Introduction

This research investigates the efficacy of word embedding in a deep learning environment for conducting text analytics tasks and summarizes the significant aspects. A systematic literature review provides an overview of existing word embedding and deep learning models. The overall structure of the paper is shown in Fig. 1.

✉ Deepak Suresh Asudani
dsasudani.phd2021.cs@nitrr.ac.in

¹ Department of Computer Science and Engineering, National Institute of Technology, Raipur, Chhattisgarh, India

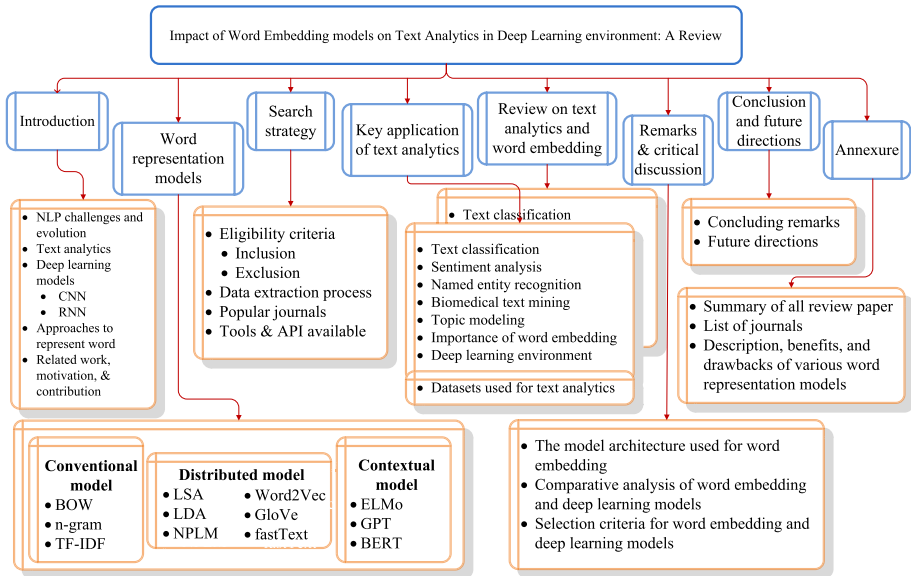
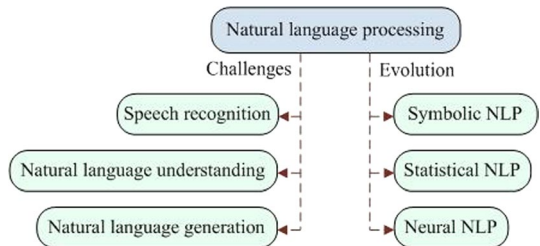


Fig. 1 Overall structure of the paper

Fig. 2 Challenges and evolution of natural language processing



1.1 Natural language processing (NLP)

NLP is a branch of linguistics, computer science, and artificial intelligence concerned with computer–human interaction, mainly how to design computers to process and evaluate huge volumes of natural language data. NLP integrates statistical, machine learning, and deep learning models with computational linguistics rules-based modeling of human language. Speech recognition, natural language interpretation or understanding (NLI or NLU), and natural language production or generation (NLP or NLG) are all common challenges in natural language processing, as shown in Fig. 2. These technologies allow computers to understand and process human language.

NLP research has progressed from punch cards and batch processing to the world of Google and others, where millions of web pages may be analyzed in under a second. NLP progresses from symbolic to statistical to neural NLP. Many NLP applications leverage deep neural network design and produce state-of-the-art results due to technological advancements, increased computer power, and abundant corpus availability (Young et al. 2018) (Lavanya and Sasikala 2021).

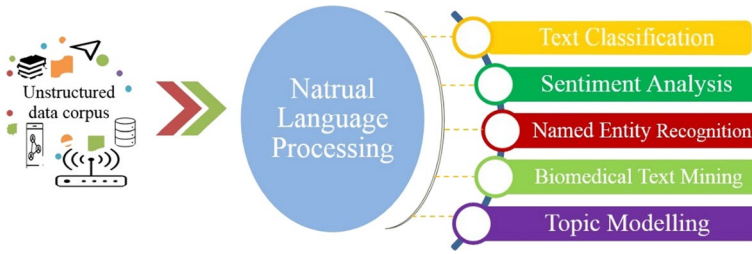


Fig. 3 NLP techniques

1.2 Text analytics

The majority of text data is unstructured and dispersed across the internet. This text data can yield helpful knowledge if it is properly obtained, aggregated, formatted, and analyzed. Text analytics can benefit corporations, organizations, and social movements in various ways. The easiest way to execute text analytics tasks is to use manually specified rules to link the keywords closely. In the presence of polysemy words, the performance of defined rules begins to deteriorate. Machine learning, deep learning, and natural language processing methods are used in text analytics to extract meaning from large quantities of text. Businesses can use these insights to improve profitability, consumer satisfaction, innovation, and even public safety. Techniques for analyzing unstructured text include text classification, sentiment analysis, named entity recognition (NER) and recommendation system, biomedical text mining, topic modeling, and others, as shown in Fig. 3. Each of these strategies is employed in a variety of contexts.

1.3 Deep learning models

Deep learning methods have been increasingly popular in NLP in recent years. Artificial neural networks (ANN) with several hidden layers between the input and output layers are known as deep neural networks (DNN). This survey reviews 193 articles published in the last three years focusing on word embedding and deep learning models for various text analytics tasks. Deep learning models are categorized based on their neural network topologies, such as recurrent neural networks (RNN) and convolutional neural networks (CNN). RNN detects patterns over time, while CNN can identify patterns over space.

1.3.1 Convolutional neural networks

CNN is a neural network with many successes and inventions in image processing and computer vision. The underlying architecture of CNN is depicted in Fig. 4. A CNN consists of several layers: an input layer, a convolutional layer, a pooling layer, and a fully connected layer. The input layer receives the image pixel value as input and passes it to the convolutional layer. The convolution layer computes output using kernel or filter values, subsequently transferred to the pooling layer. The pooling layer shrinks the representation size and speeds up computation. Local and location-consistent patterns

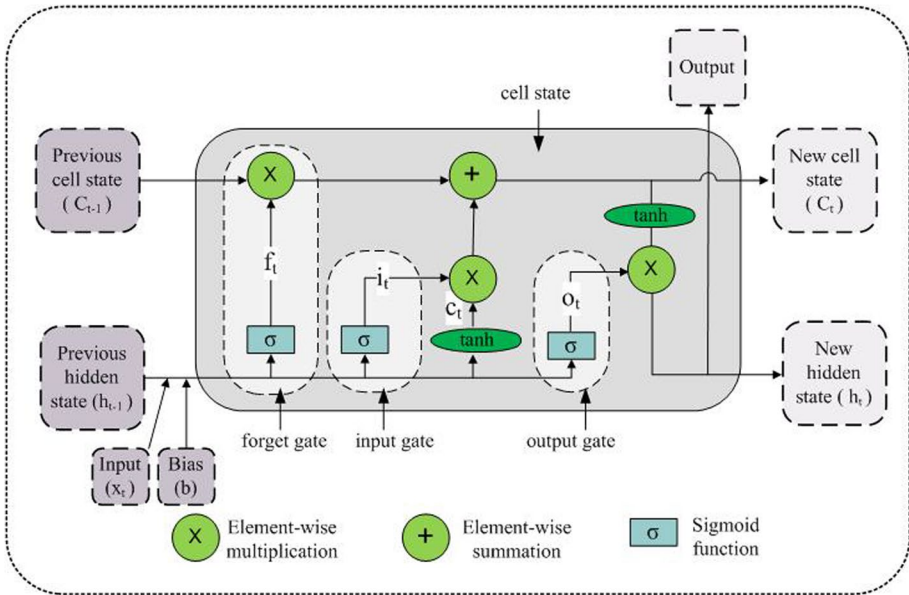


Fig. 6 The architecture of LSTM

RNN passes the intermediate information through a non-linear transformation function like tanh, as shown in Eq. (2). The intermediate output is passed through the softmax function, which output values 0 to 1 and adds up to 1, as represented using Eq. (3). RNN uses a backpropagation through time algorithm to learn from the data sequence and improve the prediction capabilities. Backpropagation is the recursive application of the chain rule, where it computes the total loss, L , as represented in Eq. (4) and shown in Fig. 5. RNN suffers due to vanishing and exploding gradients problems. The vanishing gradient problem can be addressed using the Gated Recurrent Unit (GRU) or Long Short Term Memory (LSTM) network architecture.

$$h_t = \tanh(WT_{hh}h_{t-1} + WT_{xh}x_t) \tag{2}$$

$$y_t = \text{softmax}(WT_{hy}h_t) \tag{3}$$

$$L = L_1 + L_2 + \dots + L_t \tag{4}$$

In an LSTM cell state, at a particular time t , the input vector x_t passed through the three gate vectors, hidden state, and cell state. The LSTM architecture is shown in Fig. 6. The input gate receives the input signal and modifies the values of the current cell state using Eq. (5).

The forget gate f_t updates its state using Eq. (6) and removes the irrelevant information. The output gate o_t generates the output using Eq. (7) and sends it to the network in the next step. Sigma represents the sigmoid function, and tanh represents the hyperbolic tangent function. The \odot operator defines the element-wise product. The input modulation gate, m_t is represented by Eq. (8). It uses weight matrices W and bias vector b to update the cell

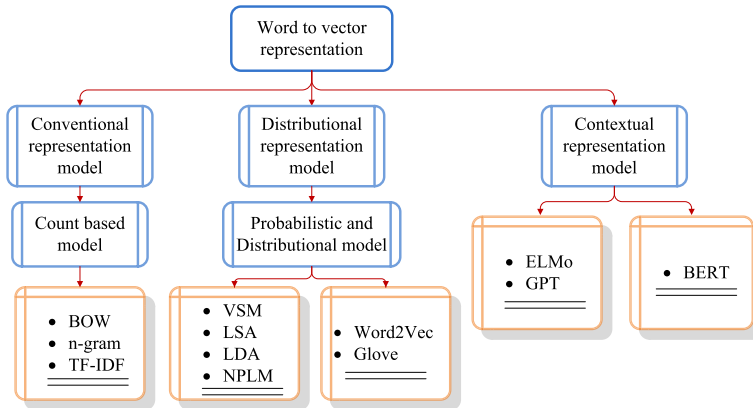


Fig. 7 Approaches to represent a word

state c_t at time t as defined by Eq. (9). The network updates the hidden states using these memory units, as shown in Eq. (10).

$$i_t = \text{sigma}(W_{xi}x_t + W_{hi}h_{(t-1)} + b_i) \tag{5}$$

$$f_t = \text{sigma}(W_{xf}x_t + W_{hf}h_{(t-1)} + b_f) \tag{6}$$

$$o_t = \text{sigma}(W_{xo}x_t + W_{ho}h_{(t-1)} + b_o) \tag{7}$$

$$m_t = \text{tanh}(W_{xc}x_t + W_{hc}h_{(t-1)} + b_c) \tag{8}$$

$$c_t = f_t \odot c_{(t-1)} + i_t \odot m_t \tag{9}$$

$$h_t = o_t \odot \text{tanh}(c_t) \tag{10}$$

1.4 Word to vector representation models

Recent breakthroughs in deep learning have significantly improved several NLP tasks that deal with text semantic analysis, such as text classification, sentiment analysis, NER and recommendation systems, biomedical text mining, and topic modeling. Pre-trained word embeddings are fixed-length vector representations of words that capture generic phrase semantics and linguistic patterns in natural language. Researchers have proposed various methods for obtaining such representations. Word embedding has been shown to be helpful in multiple NLP applications (Moreo et al. 2021).

Word embedding techniques can be categorized into conventional, distributional, and contextual word embedding models, as shown in Fig. 7. Conventional word embedding, also called count-based/frequency-based models, is categorized into a bag of words (BoW), n-gram, and term frequency-inverse document frequency (TF-IDF) models. The distributional word embedding, also called static word embedding, consists of probabilistic-distributional models, such as vector space model (VSM), latent semantic analysis (LSA), latent

Dirichlet allocation (LDA), neural probabilistic language model (NPLM), word to vector (Word2Vec), global vector (GloVe) and fastText model. The contextual word embedding models are classified into auto-regressive and auto-encoding models, such as embeddings from language models (ELMo), generative pre-training (GPT), and bidirectional encoder representations from transformers (BERT) models.

1.5 Related work

Selecting an effective word embedding and deep learning approach for text analytics is difficult because the dataset's size, type, and purpose vary. Different word embedding models have been presented by researchers to effectively describe a word's meaning and provide the embedding for processing. The word embedding model improved throughout the year to effectively represent out-of-vocabulary words and capture the significance of the contextual word. Previous studies have shown that a deep learning model can successfully predict outcomes by deriving significant patterns from the data (Wang et al. 2020).

The systematic studies on deep learning based emotion analysis (Xu et al. 2020), deep learning based classification of text (Dogru et al. 2021), and survey on training and evaluation of word embeddings (Torregrossa et al. 2021) focus on comparing the performance of word embedding and deep learning models for the domain-specific task. Studies also present an overview of other related approaches used for similar tasks. The focus of this research is to explore the effectiveness of word embedding in a deep learning environment for performing text analytics tasks and recommend its use based on the key findings.

1.6 Motivation and contributions

The primary motivation of this study is to cover the recent research trends in NLP and a detailed understanding of how to use word embedding and deep learning models to achieve efficient results on text analytics tasks. There are systematic studies on word embedding models and deep learning approaches focusing on a specific application. Still, no one includes a reference for selecting suitable word embedding and deep learning models for text analytics tasks and does not present their strengths and weaknesses.

The key contributions of this paper are as follows:

1. This study examines the contributions of researchers to the overall development of word embedding models and their different NLP applications.
2. A systematic literature review is done to develop a comprehensive overview of existing word embedding and deep learning models.
3. The relevant literature is classified according to criteria to review the essential uses of text analytics and word embedding techniques.
4. The study explores the effectiveness of word embedding in a deep learning environment for performing text analytics tasks and discusses the key findings. The review includes a list of prominent datasets, tools, and APIs available and a list of notable publications.
5. A reference for selecting a suitable word embedding approach for text analytics tasks is presented based on a comparative analysis of different word embedding techniques to perform text analytics tasks. The comparative analysis is presented in both tabular and graphical forms.

6. This paper provides a concise overview of the fundamentals, advantages, and challenges of various word representation approaches and deep learning models, as well as a perspective on future research.

The overall structure of the paper is shown in Fig. 1. Section 1 introduces the overview of NLP techniques for performing text analytics tasks, deep learning models, approaches to represent word to vector form, related work, motivation, and key contribution of the study. Section 2 presents the overall development of word embedding models. Section 3 explains the methodology of the conducted systematic literature review. It also covers the eligibility criteria, data extraction process, list of popular journals, and available tools and API. Sections 4 and 5 discuss studies on significant text analytics applications, word embedding models, and deep learning environments. Section 6 discusses a comparative analysis and a reference for selecting a suitable word embedding approach for text analytics tasks. Section 7 concludes the paper with a summary and recommendations for future work, followed by Annexures A and B, which contain an overview of all review papers and the benefits and challenges of various word embedding models.

2 Word representation models

This section will examine the techniques for word embedding training, describing how they function and how they differ from one another.

2.1 Conventional word representation models

2.1.1 Bag of words

The BoW model is a representation that simplifies NLP and retrieval. A text is an unordered collection of its words, with no attention to grammar or even word order. For text categorization, a word in a document is given a weight based on how frequently it appears in the document and how frequently it appears in different documents. The BoW representation for two statements consisting of words and their weights are as follows.

Statement 1: One cat is sleeping, and the other one is running.

Statement 2: One dog is sleeping, and the other one is eating.

	One	Cat	Is	Sleeping	And	The	Other	Dog	Running	Eating
S1	2	1	2	1	1	1	1	0	1	0
S2	2	0	2	1	1	1	1	1	0	1

The two statements have ten distinct words, representing each as ten element vector. Statement-1 is represented by [2,1,2,1,1,1,1,0,1,0], and statement-2 is represented by [2,0,2,1,1,1,1,1,0,1]. Each vector element is represented as a count of the corresponding entry in the dictionary.

BoW is suffering due to some limitations, such as sparsity. If the length of a sentence is large, it takes a more significant time to obtain its vector representation and needs considerable time to get sentence similarity. Frequent words have more power as a word occurs more times. Its frequency count increases, ultimately increasing its similarity scores,

ignoring word orders and generating the same vector for totally different sentences, losing the sentence’s contextual meaning out of vocabulary that cannot handle unseen words.

2.1.2 n-grams

It is a contiguous sequence of n tokens. For n= 1, 2, and 3, it is termed as 1-gram, 2-gram, and 3-gram, also termed as unigram model, bigram, and trigram. The n-gram model divides the sentence into word or character-level tokens. Consider two statements,

Statement-1: One cat is sleeping, the other is running.

Statement-2: One dog is sleeping, and the other one is eating.

The unigram and bigram word and character level representation is shown in the example below.

1-gram (unigram)	Word level tokens	[One, cat, is, sleeping, and, the, other, one, is, running] [One, dog, is, sleeping, and, the, other, one, is, eating]
	Character level tokens	[O, n, e, _, c, a, t, _, i, s, _, s, l, e, e, p, i, n, g, _, a, n, d, _, t, h, e, _, o, t, h, e, r, _, o, n, e, _, i, s, _, r, u, n, n, i, n, g] [O, n, e, _, d, o, g, _, i, s, _, s, l, e, e, p, i, n, g, _, a, n, d, _, t, h, e, _, o, t, h, e, r, _, o, n, e, _, i, s, _, e, a, t, i, n, g]
2-gram (bigram)	Word level tokens	[One cat, cat is, is sleeping, sleeping and, and the, the other, other one, one is, is running] [One dog, dog is, is sleeping, sleeping and, and the, the other, other one, one is, is eating]
	Character level tokens	[On, ne, e, _, c, a, at, t, _, i, is, s, _, s, sl, le, ee, ep, pi, in, ng, g, _, a, an, nd, d, _, t, th, he, e, _, o, ot, th, he, er, r, _, o, on, ne, e, _, i, is, s, _, r, ru, un, nn, ni, in, ng] [On, nn, ne, e, _, d, do, og, g, _, i, is, s, _, s, sl, le, ee, ep, pi, in, ng, g, _, a, an, nd, d, _, t, th, he, e, _, o, ot, th, he, er, r, _, o, on, ne, e, _, i, is, s, _, e, ea, at, ti, in, ng]

2.1.3 Term frequency-inverse document frequency

TF-IDF is used to find how relevant the word is in the document. Word relevance is the amount of information that gives about the context. Term frequency measures how frequently a term occurs in a document, and the term has more relevance than other terms for the document. Consider two statements,

Statement-1: One cat is sleeping, and the other one is running.

Statement-2: One dog is sleeping, and the other one is eating.

The TF score of a word in sentences is shown in the example below.

Statment 1	Words	One	Cat	Is	Sleeping	And	The	Other	Running
	TF score	2/10	1/10	2/10	1/10	1/10	1/10	1/10	1/10
	Value	0.2	0.1	0.2	0.1	0.1	0.1	0.1	0.1
Statment 2	Words	One	Dog	Is	Sleeping	And	The	Other	Eating
	TF score	2/10	1/10	2/10	1/10	1/10	1/10	1/10	1/10
	Value	0.2	0.1	0.2	0.1	0.1	0.1	0.1	0.1

$(d_{i1}, d_{i2}, \dots, d_{in})$, with weight assign using TF-IDF scheme for representing the difference in information provided by each terms. The term d_{ij} represents the weight assign to the j^{th} term in i^{th} document.

The similarity coefficient between two document D_i and D_j , represented as $S(D_i, D_j)$ is computed to express the degree of similarity between terms and their weights. Two documents with similar index terms are close to each other in the space. The distance between two document points in the space is inversely correlated with the similarity between the corresponding vectors (Salton et al. 1975). A distributional model represents a word or phrase in context, but a VSM represents meaning in a high-dimensional space (Erk 2012). VSM suffers due to the curse of dimensionality resulting from a relatively sparse vector space with a larger dataset.

2.2.1 Latent semantic analysis

LSA is an automatic statistical technique for extracting and inferring predicted contextual use relations of words in discourse sequences. Singular value decomposition (SVD) is computed using a latent semantic indexing technique. The term-document matrix is first created by determining the correlation structure that defines the semantic relationship between the words in a document. SVD extracts data-associated patterns, ignoring the less important terms. Consistent phrases emerge in the document, indicating that it is associated with the data. The SVD of the term-document ($t \times d$) matrix, X , is decomposed into three sub-matrices, such as $X = T_0 S_0 D'_0$. Where, T_0 and D'_0 are left, and right singular vectors matrices and have orthogonal unit-length columns, and S_0 is the diagonal matrix of singular values. The SVD takes a long time to map new terminology and documents and confront complex issues. The Latent Semantic Indexing (LSI) approach solves the synonymy problem by allowing numerous terms to refer to the same thing. It also helps with partial polysemy solutions (Scott Deerwester et al. 1990) (Flor and Hao 2021).

2.2.2 Latent dirichlet allocation

The LDA model is a probabilistic corpus model assigning high probability to corpus members and other comparable texts. It is a three-tier hierarchical Bayesian model in which each collection item is represented as a finite mixture across a set of underlying themes. Afterward, each topic is modeled as an infinite mixture of topic probability. For text modeling, topic probabilities provide an explicit description of a document. The latent topic is determined by the likelihood that a word appears in the topic. Even though LDA cannot collect syntactic data, it relies entirely on topic data. (Campbell et al. 2015) The LSA and LDA models construct embeddings using statistical data. The LSA model is based on matrix factorization and is subject to the non-negativity requirement. In contrast, the LDA model is based on the word distribution and is expressed by the Dirichlet prior distribution, which is the multinomial distribution's conjugate (Li and Yang 2018).

2.3 Neural probabilistic language model

Learning the joint probability function of sequences of words in a language is one of the goals of statistical language modeling. The curse of dimensionality is addressed with an NPLM that learns a distributed representation for words. Language modeling is the prediction of the probability distribution of the following word, given a sequence of words as

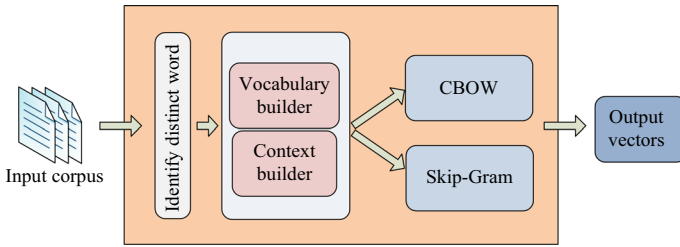


Fig. 8 Word2Vec model

shown in Eq. (11), and in each subsequent step, the product of conditional probabilities with the assumption that they are independent, as represented by Eq. (12).

$$P(x_{t+1}/x_t, \dots, x_1) \tag{11}$$

$$\begin{aligned}
 P(x_{t+1}/x_t, \dots, x_1) &= P(x_1)P(x_2/x_1)P(x_3/x_2, x_1) \dots P(x_t/x_{t-1}, \dots, x_1) \\
 &= \pi_1^t P(x_t/x_1^{t-1}) = \pi_1^t P(x_t/x_{t-1}, \dots, x_1)
 \end{aligned}
 \tag{12}$$

where the term x_t is the t th word. The conditional probability is represented by probability function C maps to the vocabulary V and maps function g to a conditional probability distribution over the word in V to obtain the following word x_t , as shown in Eq. (13). The conditional probability is decomposed into two sub-parts.

$$f(i, w_{t-1}, \dots, w_{t-n+1}) = g(i, C(w_{t-1}), \dots, C(w_{t-n+1})) \tag{13}$$

The output of function g represents the estimated probability $P(x_t = i/x_1^{t-1})$. Language models based on neural networks outperform n -gram models substantially (Bengio et al. 2003) (See 2019).

2.3.1 Word2Vec model

Conventional and static word representation methods treat words as atomic units represented as indices in a dictionary. These methods do not represent the similarity between words. The Word2Vec is a collection of model architectures and optimizations for learning word embeddings from massive datasets. The distributed representations technique uses neural networks to express word similarity adequately.

In several NLP applications, Word2Vec models such as continuous bag-of-words (CBOw) and Skip-Gram models are used to efficiently describe the semantic meanings of words (Mikolov et al. 2013a). The Word2Vec model takes a text corpus as input, processes it in the hidden layer, and outputs word vectors in the output layer. The model identifies the distinct word, creates a vocabulary, builds context, and learns vector representations of words in vector space using training data, as depicted in Fig. 8. Each unique word in the training set corresponds to a specific vector in space. Each word can have various degrees of similarity, indicating that words with similar contexts are more related.

The CBOw and Skip-Gram model architecture is shown in Fig. 9. The CBOw uses context words to forecast the target word. For a given input word, the Skip-Gram model predicts the context word.

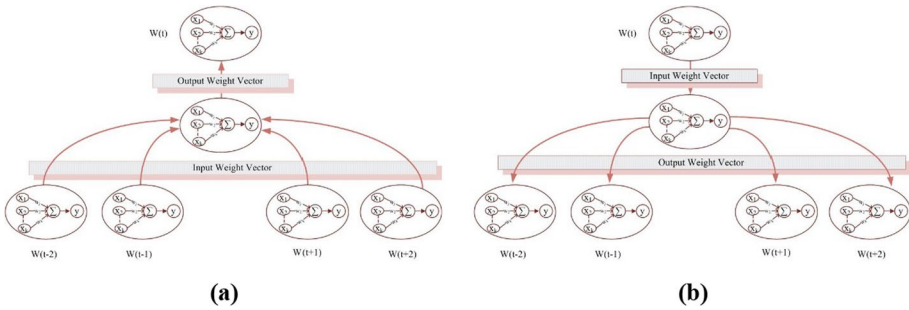


Fig. 9 The architecture of (a) CBOW model, (b) Skip-Gram model

The input is a one-hot encoded vector. The weights between the input and hidden layers are represented by the input weight vector, a $V \times N$ matrix, W . Each row of the matrix W represents the N -dimensional vector representation of the word input layers. The output weight vector represents the weights between the hidden and output layers, an $N \times V$ matrix, W' . The input and output weight vectors are used to award a score to each word in the vocabulary. In CBOW, the N -dimension vector representation v_w of the related word of the input layer is represented in each row of W . The i^{th} row of matrix W is $v_{w_i}^T$, given a context word, assuming $x_k, x_k = 1$ and $x_{k'} = 0$ for $k' \neq k$. The hidden layer activation function is linear, passing information from the previous layer to the next layer, i.e. copy the k^{th} row of matrix W to the hidden state value h . The vector representation of the input word w_1 is represented by v_{w_1} . The updated value of h is as shown in Eq. (14). The output weight matrix $W' = \{w'_{ij}\}$ is used to compute the score from vocabulary for each word u_j . The j th column of the matrix W' is represented by v'_{w_j} , as shown in Eq. (15).

$$h = W^T x = v_{w_1}^T \tag{14}$$

$$u_j = v'_{w_j}{}^T h \tag{15}$$

The output layer uses the softmax activation function to compute the multinomial probability distribution of words. The j^{th} unit output contains word representation from the input weight vector v_w and output weight vector $v'_{w'}$, as illustrated in Eq. (16).

$$p(w_j/w_1) = y_j = \frac{\exp(v'_{w_j}{}^T v_{w_1})}{\sum_{j'=1}^V \exp(v'_{w_{j'}}{}^T v_{w_1})} \tag{16}$$

For a window size of 2, the word $w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}$ are the context word for the target word w_t . Compared to the CBOW model, the Skip-Gram model is the polar opposite. Based on the input word, the Skip-Gram model predicts context words. For a window size of 2, the word w_t is the input word for the output context words $w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}$. The input weight vector is computed using a similar approach to the CBOW model. For the input w_1 the output of j^{th} word on C multinomial distribution is represented by $y_{c,j}$. Input to the j^{th} unit is represented by $u_{c,j}$. The j^{th} word of the output layer is $w_{c,j}$ from the c^{th} panel and the word $w_{o,c}$ represents the output context word. The output for each word is computed using the output weight vector, as represented in Eq. (17).

$$p(w_{c,j} = w_{o,c}/w_I) = y_{c,j} = \frac{\exp(u_{c,j})}{\sum_{j'=1}^v \exp(u_{j'})} \quad (17)$$

Multiplying the input by the input weights between the input and the hidden layer yields the input-hidden matrix. The output layer computes multinomial distributions using the hidden output weight matrix. The resulting errors are calculated by element-wise adding the error vectors. The error is propagated back to update the weight until the true element is found. The weights obtained between the hidden and output layers after training are called the word vector representation (Mikolov et al. 2013b).

2.3.2 GloVe

Word embeddings learned through Word2Vec are better at capturing word semantics and exploiting word relatedness. Word2Vec focuses solely on information collected from the local context window, whereas global statistic data is neglected. The GloVe is a hybrid of LSA and CBOW that is efficient and scalable for large corpora (Jiao and Zhang 2021). The GloVe is a popular model based on the global co-occurrence matrix, where each element x_{ij} in the matrix indicates the frequency with which the words w_i and w_j co-occur in a given context window. The number of times a particular word appears in the context of the word i , is denoted by X_i . The P_{ij} represents the likelihood of the word j appearing in the context of the word i , as presented in Eqs. (18)–(19).

$$X_i = \sum_k X_{ik} \quad (18)$$

$$P_{ij} = P(j/i) = \frac{X_{ij}}{X_i} \quad (19)$$

A weighted least squares regression model approximates the relationship between a word embedding and a co-occurrence matrix. The function $f(X_{ij})$ represents a weighting function for the vocabulary of size V . The w represents the word vectors and \tilde{w} represents the context word vectors. The term b_i and \tilde{b}_j are bias for words w_i and w_j to restore the symmetry. When the word frequency is too high, a weight function $f(x)$, as shown in Eqs. (20)–(21), ensures that the weight does not increase significantly.

$$J = \sum_{i,j=1}^V f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij})^2 \quad (20)$$

$$f(x) = \begin{cases} (x/x_{max})^{3/4} & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases} \quad (21)$$

The GloVe is an unsupervised learning technique for constructing word vector representations. The resulting illustrations highlight significant linear substructures of the word vector space, trained using a corpus's aggregated global word-word co-occurrence

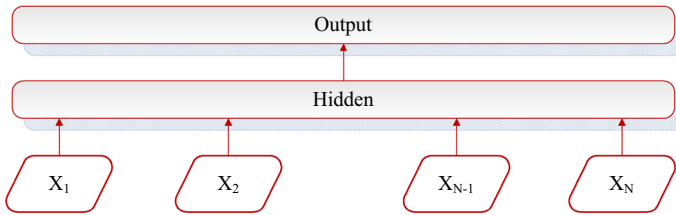


Fig. 10 The model architecture of fastText

information. Glove pre-trained word embedding is based on 400 K vocabulary words trained on Wikipedia 2014 and Gigaword 5 as the corpus and 50, 100, 200, and 300 dimensions for word display (Pennington et al. 2014).

2.3.3 fastText

The fastText model uses internal subword information in the form of character n-grams to acquire information about the local word order and allows it to handle unique, out-of-vocabulary terms. The method creates word vectors to reflect the grammar and semantic similarity of words and produce vectors for unseen words. The Facebook AI Research lab announced fastText, an open-source technique for generating vectors for unknown words based on morphology. Each word w is expressed as w_1, w_2, \dots, w_n in n-gram features and utilized as input to the fastText model. For example, the character trigram for the word “sleeping” is $\langle sl, sle, lee, eep, epi, pin, ing, ng \rangle$. Each n-gram will create a vector, and the original vector will be combined with the vector of all its related n-grams during the training phase, as shown in Fig. 10.

Input to the model contains entire word vectors and character-level n-gram vectors, which are combined and averaged simultaneously (Joulin et al. 2017). Pre-trained word vectors generated from fastText using standard crawl and Wikipedia are available for 157 languages. The fastText model is trained using CBOW in dimension 300, with character n-grams of length five and a size 5 and 10 negatives window.¹

2.4 Contextual representation models

The conventional and distributional representation approaches learn static word embedding. After training, each word representation is identified. The semantic meaning of the word polysemy can vary depending on the context. Understanding the actual context is required for most downstream tasks in natural language processing. For example, “apple” is a fruit but usually refers to a firm in technical articles. The vectors of words in the contextualized word embedding can be modified according to the input contexts utilizing neural language models.

¹ <https://fasttext.cc/docs/en/crawl-vectors.html>

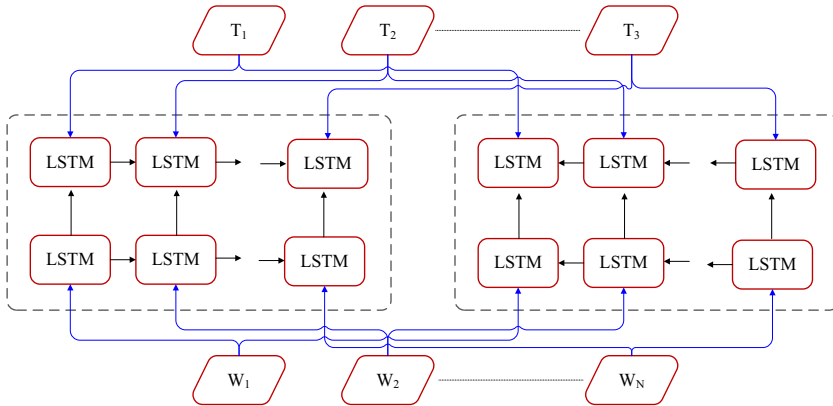


Fig. 11 The architecture of ELMo

2.4.1 Embeddings from language models

The ELMo representations use vectors derived from a bidirectional LSTM (BiLSTM) trained on a large text corpus. The ELMo model effectively addresses the problem of comprehending the syntax and semantic meaning of words and the language contexts in which they are used. ELMo considers the complete sentence when assigning an embedding to each word. It employs a bidirectional design, embedding depending on the sentence’s next and preceding words, as shown in Fig. 11.

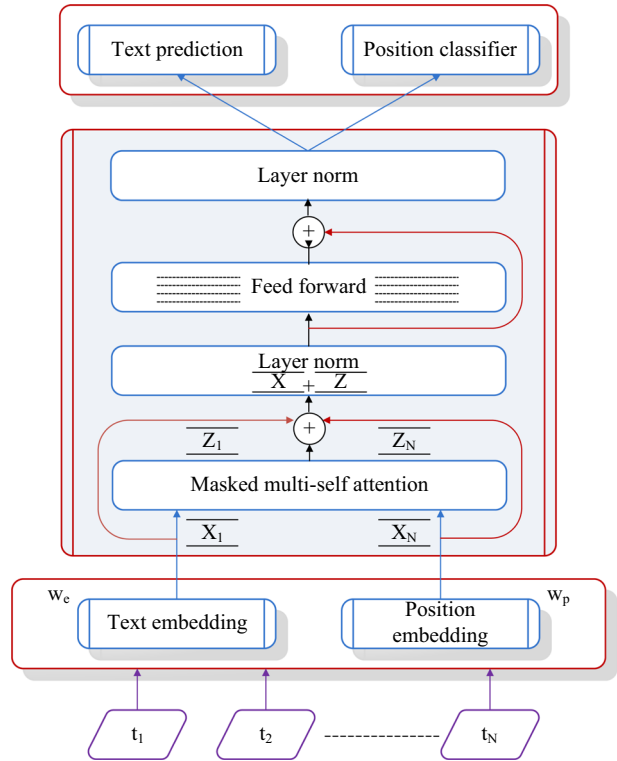
For a sequence of N tokens (t_1, t_2, \dots, t_N) , the aim is to find the language model’s greatest probability in both directions. The likelihood of the sequence is computed using a forward language model, which models the chance of token t_k considering the history $(t_1, t_2, t_3, \dots, t_k)$. A backward language model is identical to a forward language model but goes backward through the sequence, anticipating the previous token based on the future context. The forward and backward language model and the join expression that optimizes the log probability in both directions are shown in Eqs. (22)–(24) (Peters et al. 2018).

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_1, t_2, \dots, t_{k-1}) \tag{22}$$

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_{k+1}, t_{k+2}, \dots, t_N) \tag{23}$$

$$\sum_{k=1}^N (\log p(t_k | t_1, t_2, \dots, t_{k-1}) + \log p(t_k | t_{k+1}, t_{k+2}, \dots, t_N)) \tag{24}$$

Fig. 12 The architecture of GPT



2.4.2 Generative pre-training

The morphology of words in the application domain can be extensively exploited with GPT. GPT uses a one-way language model, transformer, to extract features, whereas ELMo employs a BiLSTM. The architecture of GPT is shown in Fig. 12.

A standard language modeling objective for a sequence of tokens (t_1, t_2, \dots, t_N) to maximize the likelihood is shown in Eq. (25). The language model employs a multi-layer transformer decoder with a self-attention mechanism to anticipate the current word through the first N-word (Vaswani et al. 2017). To achieve a proper distribution over target words, the GPT model employs a multi-headed self-attention operation over the input contextual tokens, accompanied by position-wise feed-forward layers, as shown in Eqs. (26)–(28).

$$L_1(X) = \sum_i \log P(t_i | t_{i-N}, \dots, t_{i-1}; \theta) \tag{25}$$

$$h_0 = UW_e + W_p \tag{26}$$

$$h_1 = \text{transformer}_{\text{block}(h_{i-1})} v_i \in [1, n] \tag{27}$$

$$P(u) = \text{softmax}(h_n W_e^T) \tag{28}$$

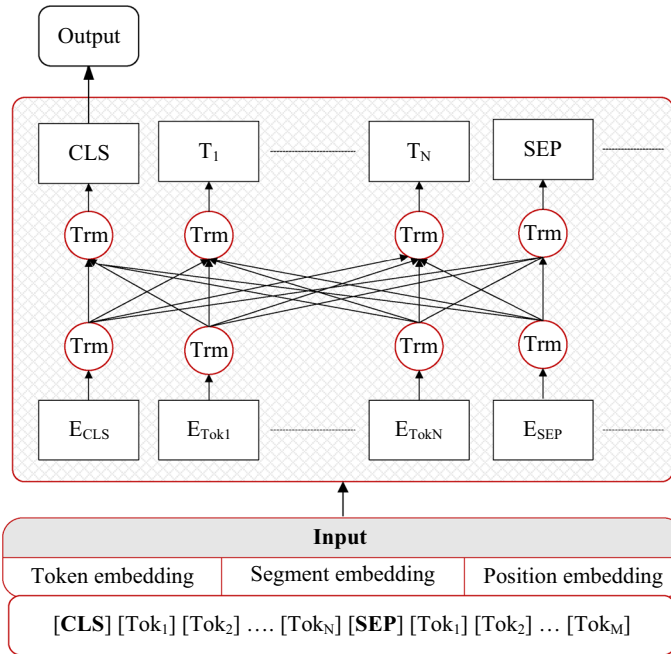


Fig. 13 BERT Architecture

The number of layers is represented as n , W_e represents the token embedding matrix, the position embedding matrix W_p and U is the context vector of tokens (Radford et al. 2018).

2.4.3 Bidirectional encoder representations from transformers

The ELMo model takes a feature-based approach and adds pre-trained representation as a feature. The GPT model uses a fine-tuning technique and only uses task-specific parameters that have been trained on downstream tasks. BERT model architecture includes a multi-layer bidirectional transformer encoder, as depicted in Fig. 13.

BERT employs masked language modeling to optimize and combine position embedding with static word embeddings as model inputs. It follows frameworks for both pre-training and fine-tuning. The model is trained on unsupervised learning from several pre-training tasks during pre-training. The BERT model is fine-tuned by first initializing it using the pre-trained parameters and then fine-tuning all parameters using labeled data from the downstream jobs (Devlin et al. 2019).

BERT uses word-piece embeddings. A special classification token [CLS] is always the first token in every sequence. Use the special token [SEP] to separate the sentences. BERT uses a deep, pre-trained neural network with transformer architecture to create dense vector representations for natural language. The BERT base or large category TF Hub model has $L=12/24$ hidden layers (transformer blocks), $H=768/1024$ hidden size, and $A=12/16$ attention heads (TensorFlow Hub).

Table 1 Set of search phrases and words for each of the EDS

EDS database	Search phrases and words
Scopus (Journal articles)	TITLE ("word embedding" OR word2vec OR glove) and TITLE-ABS-KEY("deep learning") AND (LIMIT-TO (SUBJAREA, "COMP")) AND (LIMIT-TO(PUBYEAR,2021) OR LIMIT-TO (PUBYEAR,2020) OR LIMIT-TO(PUBYEAR,2019)) AND (LIMIT-TO(LANGUAGE, "English")) AND (LIMIT-TO(SRCTYPE, "j"))
Scopus (Peer-reviewed conference articles)	TITLE-ABS-KEY("word embedding" OR "word2vec" OR "Glove") AND TITLE-ABS-KEY("deep learning") AND (LIMIT-TO(EXACTSRCTITLE, "Emnlp Ijcnlp Conference On Empirical Methods In Natural Language Processing And International Joint Conference On Natural Language Processing Proceedings Of The Conference") OR LIMIT-TO(EXACTSRCTITLE, "Aaai Conference On Artificial Intelligence") OR LIMIT-TO(EXACTSRCTITLE, "Acl Ijcnlp Association For Computational Linguistics And The International Joint Conference On Natural Language Processing Of The Asian Federation Of Natural Language Processing Proceedings Of The Conference") OR LIMIT-TO(EXACTSRCTITLE, "Cognitive Modeling And Computational Linguistics Proceedings Of The Workshop") OR LIMIT-TO(EXACTSRCTITLE, "Conference On Computational Natural Language Learning Proceedings")) AND LIMIT-TO(PUBYEAR, 2021) OR LIMIT-TO(PUBYEAR, 2020) OR LIMIT-TO(PUBYEAR, 2019)) AND (LIMIT-TO(LANGUAGE, "English")) AND (LIMIT-TO(SUBJAREA, "COMP"))
Science direct	Articles with these terms: word embedding; word2vec; glove, Title, abstract or author-specified keywords: deep learning, Year: 2019–2021
IEEE Xplore	("Document Title": word embedding) OR ("Document Title":word2vec) OR ("Document Title": glove) AND ("Document Title": deep learning) AND ("Abstract": deep learning) AND ("Author Keywords": deep learning), Filters Applied: Journals, Year range - 2019–2021

3 Search strategy

A comprehensive search for possibly relevant literature was undertaken in three electronic data sources (EDS), namely Institute of Electrical and Electronics Engineers (IEEE) Xplore, Scopus, and Science Direct, following the systematic guidelines outlined and declared by (Kitchenham 2004) (Okoli and Schabram 2010) for the journal and peer-reviewed conference articles published between the year 2019 to 2021. The search included the keywords “word embedding” or Word2Vec or GloVe in conjunction with deep learning. The set of search phrases and words used for each EDS is shown in Table 1.

3.1 Eligibility criteria

Article eligibility and inclusion is an essential and strict inspection method for including the best potential articles in the study. The following points are defined to choose research examining the impact of word embedding models on text analytics in deep learning environments. The primary study selection criteria are categorized into inclusion criteria and exclusion criteria.

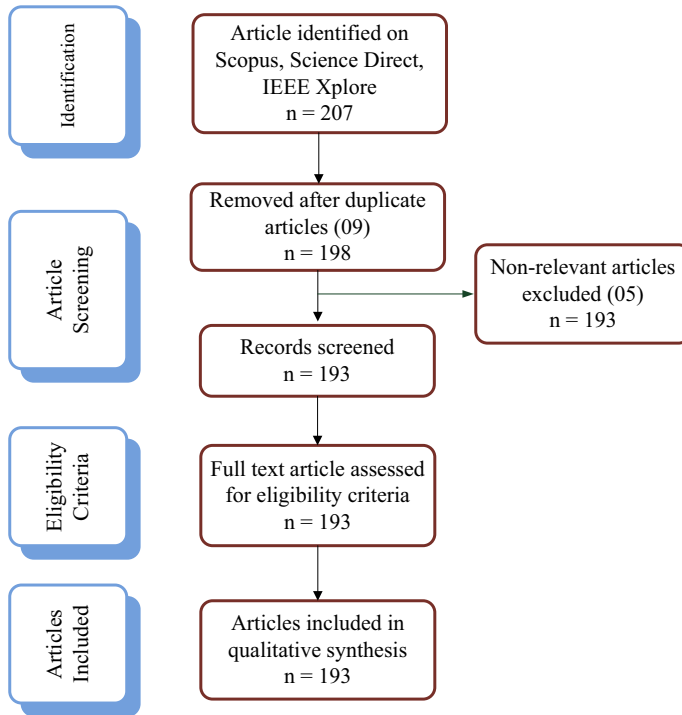


Fig. 14 PRISMA diagram

3.1.1 Inclusion criteria

- Studies focus primarily on word embedding models that have been applied or reviewed for analytics.
- Any analytics task, such as text classification, sentiment analysis, text summarization, and other text analysis activities utilizing word embedding models, will be included in the articles.
- The research article from the database is selected only from the subject of computer science.
- Research papers have been accepted and published in important and determinant peer-reviewed conferences focusing on word embedding and natural language processing and published in reputed journals.
- Studies were published from 2019 to 2021.

3.1.2 Exclusion criteria

- Studies not in the English language.
- Studies focused only on understanding deep learning models, such as their architectural behaviors or motivation to utilize them.
- Articles that do not meet the inclusion criteria are excluded.
- Articles that were already examined in other EDS will be excluded.

Table 2 Summary of articles selected for review

	Result by query	Articles selected for review	Articles repeated	Articles out of scope
Scopus (Journal articles)	72	69	00	03
Scopus (Peer-reviewed conference articles)	45	45	00	00
Science Direct	61	58	01	02
IEEE Xplore	29	21	08	00
	207	193	09	05

Table 3 Description of data extraction

Data items	Description
Bibliographic information	Author, title, name of the publisher/journal, year of publication
Embedding approach used	The various embedding techniques that have been employed
Task and method	Various forms of word embedding models and deep learning methodologies are used in analytics
Dataset used	Datasets used for training and testing the suggested method were gathered
Performance parameter and evaluation	Enlist the parameters used for evaluation and the actual performance achieved using the proposed approach
Implementation API or tools	Information about various tools and APIs used for experimentation
Key/Critical findings	Significant findings and critical discussion related to the proposed approach and techniques used
Limitations and Future work	Limitations of the current research work and proposed future work

The EDS database is used to find the literature with the keywords “word embedding OR Word2Vec OR GloVe” and “deep learning” used in the title, abstract, and keywords section. The overall number of articles shown by the database is huge. When the research is confined to 2019 to 2021, the number drops to 207. The process is needed to filter more for the quality of the review. The language is selected only English, and the subject area is chosen as computer science. The published articles in important and determinant peer-reviewed conferences focusing on word embedding and natural language processing and reputed journals are included for the study’s reliability and quality. The PRISMA diagram shown in Fig. 14 depicts the criteria for selecting articles and information about the article for review and record.

The summary of articles selected for review is shown in Table 2. The 09 studies are excluded as duplicate articles from different EDS, and the 05 studies irrelevant to this review are also excluded. The final 193 articles on word embedding models in conjunction with deep learning and its applications in text analytics are selected to analyze the literature and find the gap and research direction.

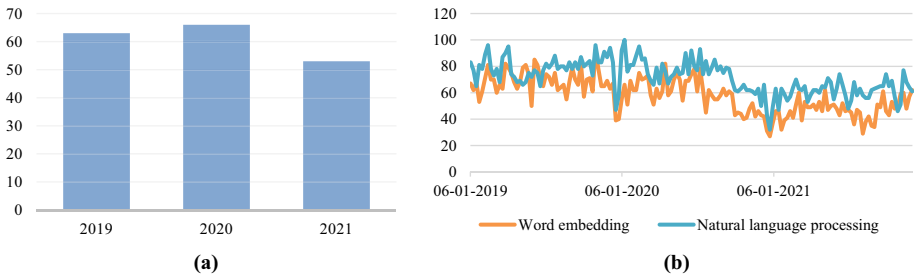


Fig. 15 (a) Year-wise publication records selected for review, (b) Analysis of search query on word embedding and NLP in Google Trends

3.2 Data extraction process

A detailed data extraction format is prepared in the spreadsheet to minimize any bias in the data extraction process. The spreadsheet was primarily used to extract and maintain each chosen research study data. A detailed overview of the data extraction procedure is discussed in Table 3.

3.3 Popular journals and year-wise studies

The research is restricted to important and determinant peer-reviewed conferences focusing on word embedding and natural language processing and reputed journal publications published between 2019 and 2021. The terms word embedding, deep learning, and their applications in text analytics were used in the search. Only papers that meet the inclusion and exclusion criteria are chosen for review. The study began in the fourth quarter of 2021; hence, fewer publications than in 2020. It is expected to have more publications in the coming years. Articles selected for the study are shown year-wise in Fig. 15(a). Google Trends² is used to analyze word embedding and NLP topics in Google search queries worldwide from 2019 through 2021. The comparison of the search volume of queries over time is displayed in Fig. 15(b). According to recent trends, the embedding technique for natural language processing jobs has evolved significantly. The choice of an effective embedding strategy is critical to the success of an NLP task.

For review, articles published in important and determinant peer-reviewed conferences focusing on word embedding and natural language processing and reputed journals are chosen. It has been discovered that Elsevier publishes nearly 50% of the selected publications, almost 25% are published by IEEE, and Springer Nature publishes nearly 10%.

The journals of Elsevier publications, Information Processing and Management, Knowledge-Based Systems, and Applied Soft Computing, had 34 papers selected for review, the most of any other publication. IEEE Access is ranked second on the list, with 27 articles chosen for evaluation. The third journal on the list is Springer's Neural Computing and Applications. A circular dendrogram depicting the name of peer-reviewed conferences and

² <https://trends.google.com/>

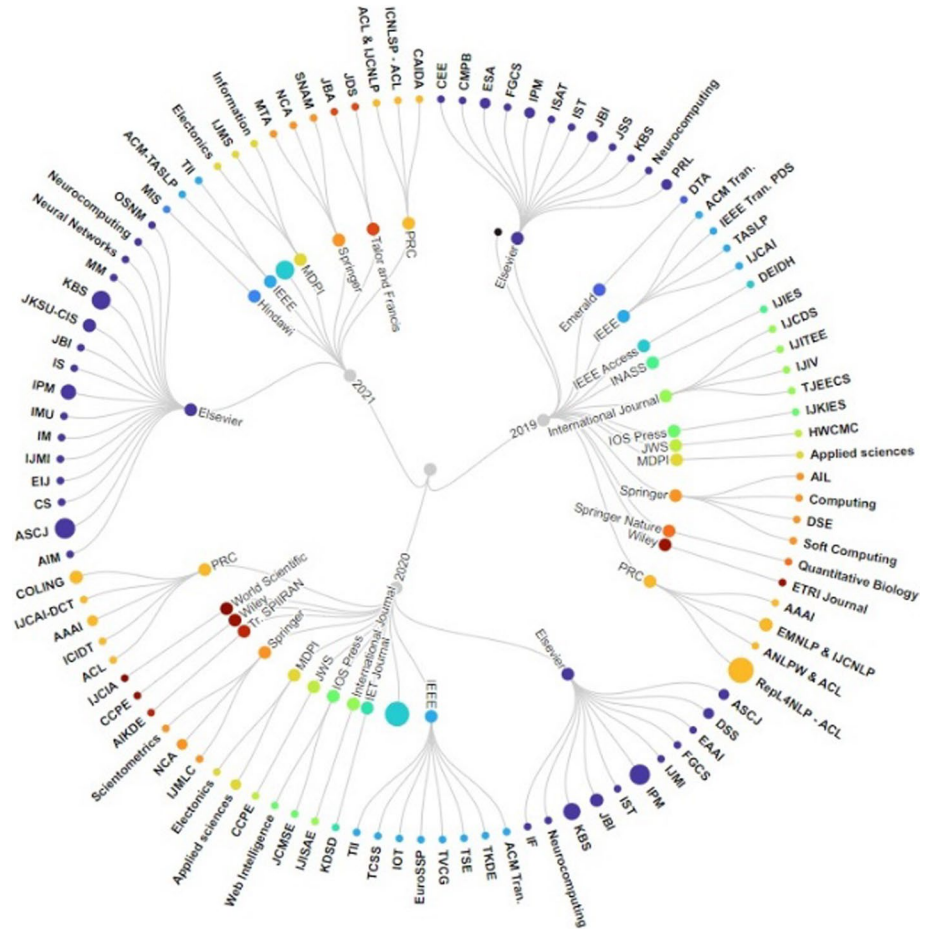


Fig. 16 Peer-reviewed conferences and journals selected for the current review

journals selected for current review by year is shown in Fig. 16. The peer-reviewed conference and journal’s names and abbreviations are listed in Table 13 in Annexure A.

3.4 Tools and APIs available for implementing word embedding models

This section provides an overview of the available tools and API for implementing word embedding models.

Natural Language Toolkit: Natural Language Toolkit (NLTK)³ is a free and open-source Python library for natural language processing. NLTK provides stemming, lower-case, categorization, tokenization, spell check, lemmatization, and semantic reasoning text processing packages. It gives access to lexical resources like WordNet.

³ <https://www.nltk.org/>

Scikit-learn: Scikit-learn⁴ is a Python toolkit for machine learning that supports supervised and unsupervised learning. It also includes tools for model construction, selection, assessment, and other features, such as data preprocessing. For the development of traditional machine learning algorithms, two Python libraries, NumPy and SciPy, are useful.

TensorFlow: Tensorflow⁵ is a free and open-source library for creating machine learning models. TensorFlow uses a Keras-based high-level API for designing and building neural networks. TensorFlow was created to perform machine learning and deep neural network research by researchers on the Google Brain team. Its flexible architecture enables computing to be deployed over various platforms like CPU, GPU, and TPU and makes it significantly easier for developers to transition from model development to deployment.

Keras: Keras⁶ is a Google-developed high-level deep learning API for implementing neural networks. It is built in Python and is used to simplify neural network implementation. It also enables the computation of numerous neural networks in the backend. Keras support the frameworks such as Tensorflow, Theano, and Microsoft Cognitive Toolkit. Keras allows users to create deep models for smartphones, browsers, and the java virtual machine. It also allows distributed deep-learning model training on clusters of GPU and TPU.

PyTorch: PyTorch⁷ is an open-source machine learning framework initially created by Facebook AI Research lab (FAIR) to speed up the transition from research development to commercial implementation. PyTorch has a user-friendly interface that allows quick, flexible experimentation and output. It supports NLP, machine learning, and computer vision technologies and frameworks. It enables GPU-accelerated Tensor calculations and the creation of computational graphs. The most recent version of PyTorch is 1.11, which includes data loading primitives for quickly building a flexible and highly functional data pipeline.

Pandas: Pandas⁸ is an open-source Python framework that supports high-performance, user-friendly information structures and analytic tools for Python. Pandas are applied in various scientific and corporate disciplines, including banking, business, statistics, etc. Pandas 1.4.1 is the most recent version and is more stable in terms of regression support.

NumPy: Travis Oliphant built Numerical Python (NumPy)⁹ in 2005 as an open-source package that facilitates numerical processing with Python. It has matrices, linear algebra, and the Fourier transform functions. The array object in NumPy is named ndarray, and it comes with a slew of helper functions that make working with it a breeze. The latest version of NumPy is 1.22.3, and it is used to interface with a wide range of databases smoothly and quickly.

SciPy: NumPy includes a multidimensional array with excellent speed and array manipulation features. SciPy¹⁰ is a Python library based on NumPy and is available for free. SciPy consists of several functions that work with NumPy arrays and are helpful for a variety of scientific and engineering tasks. The latest version of the SciPy toolkit is 1.8.0, and it offers excellent roles and methods for data processing and visualization.

⁴ <https://scikit-learn.org/stable/>

⁵ <https://www.tensorflow.org/>

⁶ <https://keras.io/>

⁷ <https://pytorch.org/>

⁸ <https://pandas.pydata.org/>

⁹ <https://numpy.org/>

¹⁰ <https://scipy.org/>

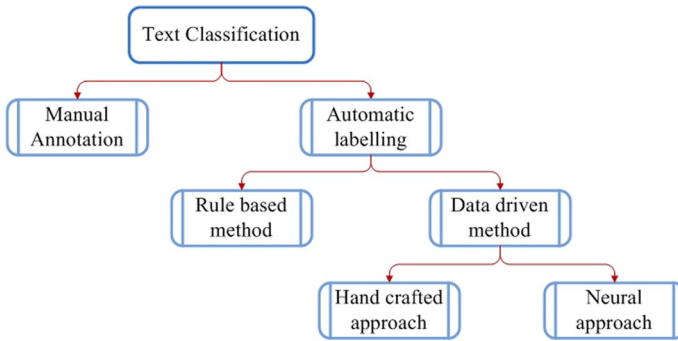


Fig. 17 Approaches for text classification

4 Key applications of text analytics

Techniques for analyzing unstructured text include text classification, sentiment analysis, NER and recommendation systems, biomedical text mining, and topic modeling.

4.1 Text analytics

4.1.1 Text classification

Text classification is the process of categorizing texts into organized groups. Text gathered from a variety of sources offers a great deal of knowledge. It is difficult and time-consuming to extract usable knowledge from unstructured data. Text classification can be done manually or automatically, as shown in Fig. 17.

Automatic text classification is becoming progressively essential due to the availability of enormous corpora. Automatic text classification can be done using either a rule-based or data-driven technique. A rule-based technique uses domain knowledge and a set of pre-defined criteria to classify text into multiple groups. Text is organized using a data-driven approach based on data observations. Machine learning or deep learning algorithms can be used to discover the intrinsic relationship between text and its labels based on data observation.

A data-driven technique fails to extract relevant knowledge from a large dataset using solely handmade characteristics. An embedding technique is used to map the text into a low-dimensional feature vector, which aids in extracting relationships and meaningful knowledge (Dhar et al. 2020).

4.1.2 Sentiment analysis

Sentences can be articulated in a variety of ways. It might be expressed through various emotions, judgments, visions or insights, or people's perspectives. The meaning of individual words has an impact on readers and writers. The writer uses specific words to communicate feelings, and the readers strive to interpret the emotion depending on their abilities to analyse. Deep learning systems have already demonstrated outstanding performance in NLP applications such as sentiment classification and emotion detection within many

datasets. These models do not require any predefined selected characteristics. Instead, it learns advanced representations of the input datasets on its own (Dessì et al. 2021). Sentiment analysis techniques are divided into lexicon-based approaches, machine-learning approaches, and a combination of the two (Mohamed et al. 2020). The internet is an unorganized and rich source of knowledge that contains many text documents offering thoughts and reviews. Personal decisions, businesses, and institutions can benefit from sentiment recognition (Onan 2021).

4.1.3 Named entity recognition

A named entity is a word used to differentiate one object from a set of entities that share similar features. It restricts the range of entities that describe a subject by using one or more restrictive identifiers. At the sixth Message Understanding Conference, the term Named Entity was first used to describe the problem of recognizing names of enterprises, persons, and physical locations in literature and price, timing, and proportion statements. Then there was a surge in interest in NER, with numerous researchers devoting significant time and effort to the subject (Grishman and Sundheim 1996), (Nasar et al. 2021). The extraction of intelligent information from text relies heavily on NER. The NER task is difficult due to the polymorphic behavior of many words (Khan et al. 2020). NER is used in various NLP applications, including text interpretation, information extraction, question answering, and autonomous text summarization. In NER, four main approaches are used: (1) Rule-based approaches, which rely on hand-crafted rules, (2) Unsupervised learning methods, which use unsupervised algorithms rather than hand-labeled training instances (3) Feature-based supervised learning techniques primarily depend on supervised learning algorithms that have been carefully engineered, (4) Deep-learning-based techniques that generate representations necessary for classification and identification from training dataset in an end-to-end way.

4.1.4 Biomedical text mining

Healthcare experts are struggling to classify diseases based on available data. Humans must recognize clinically named entities to assess massive electronic medical records effectively. Conventional rule-based systems require a significant amount of human effort to create rules and vocabulary, whereas machine learning-based approaches require time-consuming feature extraction. Deep learning models like LSTM with conditional random field (CRF) performed admirably in several datasets. Clinical named entity recognition is a process that identifies specific concepts from unorganized texts, medical tests, and therapies. It is crucial to convert unorganized electronic medical record material into organized medical information. (Yang et al. 2019).

4.1.5 Topic modeling

Topic modeling aims to ascertain how underlying document collections are structured. Topic models were first created to retrieve information from massive document collections. Without relying on metadata, topic models can be used to explore sets of journals by article subject. The LSA uses SVD to extract the fundamental themes from a term-document matrix, resulting in mathematically independent issues. Similar to how principal component analysis reduces the number of features in a prediction task, topic models are simply

Table 4 Dataset used for text analytics purpose

Sr. No	Name of dataset	References
1	Amazon product review	Liu et al. (2021b), Wang et al. (2021c), Hajek et al. (2020), Rezaeinia et al. (2019), Hao et al. (2020), Yang et al. (2021a), Dau et al. (2021), and Khan et al. (2021)
2	Arabic news datasets	Almuzaini and Azmi (2020), Alrajhi and ELAffendi (2019), Almuhareb et al. (2019), and Elnagar et al. (2020)
3	Fudan dataset	Zhang et al. (2021) and Zhu et al. (2020a)
4	i2b2: Informatics for Integrating Biology & the Bedside	Yang et al. (2019), Catelli et al. (2021), and Catelli et al. (2020)
5	IMDB	Li et al. (2021), Wang et al. (2021c), Jang et al. (2020), Hao et al. (2020), and Zhu et al. (2020a)
6	Yelp	Wang et al. (2021c), Alamoudi and Alghamdi (2021), Hao et al. (2020), Zhu et al. (2020a), Yang et al. (2021a), Dau et al. (2021), Sun et al. (2020a), and Khan et al. (2021)
7	SemEval	Wang et al. (2021c), Alamoudi and Alghamdi (2021), Naderalvojud and Sezer (2020), González et al. (2020), Rida-e-fatima et al. (2019), Zhu et al. (2020a), Liu and Shen (2020), and Sharma et al. (2021)
8	Sogou	Zhang et al. (2021) and Xiao et al. (2019)
9	Standford sentiment treebank	Wang et al. (2021c), Naderalvojud and Sezer (2020), and Rezaeinia et al. (2019)
10	Twitter	Amin et al. (2020), Alharthi et al. (2021), and Malla and Alphonse (2021)
11	Wikipedia	Li et al. (2021) and Zhang et al. (2021)
12	Word-Sim	Hammar et al. (2020), Li et al. (2019a), and Zhu et al. (2020a)

a compression technique that maximizes topic variance on a simplified representation of a document collection (Zhao et al. 2021). Text classification is the process of organizing text to extract valuable information from it. In contrast, topic modeling is determining an abstract topic for a group of texts or documents. Topic modeling is commonly used to extract semantic information from textual material (Kumar et al. 2021).

4.2 Datasets used for text analytics

This section outlines the datasets commonly used for text analytics purposes, as shown in Table 4. Researchers have offered several text analytics datasets. Text classification, sentiment analysis, NER, recommendation systems, and topic modeling are among the application fields found in the literature. The overview of attributes in terms of application area, datasets, model architecture, embedding methods, and performance evaluation are illustrated in Annexure A.

Amazon dataset: Customer reviews of products purchased through the Amazon website are included in the dataset. The dataset consists of binary and multiclass classifications for review categories. The data is arranged into training and testing sets for both product classification categories.

Arabic news datasets: The Arabic newsgroups dataset contains documents posted to several newsgroups on various themes. Different versions of this dataset are used for text classification, text clustering, and other tasks. The Arabic news texts corpus is organized into nine categories: culture, diversity, economy, international news, local news, politics, society, sports, and technology. It contains 10,161 documents with a total of 1.474 million words.

Fudan dataset: This is an image database containing pedestrian detection images. The photographs were taken in various locations around campus and on city streets. At least one pedestrian will appear in each photograph. The heights of tagged pedestrians lie between (180, 390) pixels. All of the pedestrians who have been classified are standing up straight. There are 170 photos in all, with 345 pedestrians tagged, with 96 photographs from the University of Pennsylvania and 74 from Fudan University.

i2b2: Informatics for Integrating Biology & the Bedside (i2b2) is a fully accessible clinical data processing and analytics exploration platform allowing heterogeneous healthcare and research data to be shared, integrated, standardized, and analyzed. All labeled and unannotated, de-identified hospital discharge reports are provided for academic purposes.

Movie review dataset: The movie review dataset is a set of movie reviews created to identify the sentiment involved with each study and decide whether it is favorable or unfavorable. There are 10,662 sentences, with an equal amount of negative and positive examples.

Yelp dataset: Two sentiment analysis tasks are included in the Yelp dataset. One method is to look for sentiment labels with finer granularity. The other predicts both excellent and negative emotions. Yelp-5 has 650,000 training data and 50,000 testing data for negative and positive classes, while Yelp-2 has 560,000 training datasets and 38,000 testing datasets.

SemEval: SemEval is a domain-specific dataset with reviews of laptops and restaurant services thoroughly annotated by humans. The overall aspect of a sentence, section, or text span, irrespective of the entities or their characteristics, the SemEval dataset, is frequently used. The dataset comprises over three thousand reviews in English for each product category.

Sogou dataset: The Sogou news dataset combines the news corpora from SogouCA and SogouCS. This Chinese dataset includes around 2.7 billion words and is published by a Chinese commercial search engine.

Stanford Sentiment Treebank (SST) dataset: The SST dataset is a more extended version of the movie review data. The SST1 includes fine-grained labels in a multiclass movie review dataset with training, testing, and validation sets. The binary label dataset in SST2 is split into three sections: training, testing, and validation.

Twitter dataset: With the tremendous increase in online social networking websites like blogs, vital information in sentiments, thoughts, opinions, and epidemic outbreaks is being conveyed. Twitter generates vast data about epidemic outbreaks, customer reviews about the product, and survey information. The Twitter Streaming API can be used to obtain a dataset from Twitter that includes disease information and a geographical study of Twitter users.

Wikipedia: Wikipedia pages are taken as the corpus to train the model. The preprocessing operations on the pages extract helpful information such as an article abstract. Processing takes place using a dictionary of selected terms.

WordSim: WordSim is a set of tests for determining the similarity or relatedness of words. The WordSim353 dataset consists of two groups: the first set includes 153-word

pairs for evaluating similarity assigned by 13 subjects, and the other contains 16-word pairs for evaluating relatedness given by 16 subjects.

5 Review on text analytics, word embedding application, and deep learning environment

For many domains, researchers have created numerous text analytics models. When creating text analytics models, the primary concern that comes to mind is “what type of embedding method is suited for which application area and the appropriate deep learning strategy”. A description of various text analytics strategies with different embedding methods and deep learning algorithms is shown in Annexure A. It depicts the multiple approaches utilized and their performance as a function of the application domain.

5.1 Text classification

Text categorization issues have been extensively researched and solved in many real-world applications. Text classification is the process of grouping together texts like tweets, news articles, and customer evaluations. The construction of text classification and document classification techniques includes extracting features, dimension minimization, classifier selection, and assessments (Jang et al. 2020). Recent advances have focused on learning low-dimension and continuous vector representations of words, known as word embedding, which may be applied directly to downstream applications, including machine translation, natural language interpretation, and text analytics (El-Alami et al. 2021) (Elnagar et al. 2020). Word embedding uses neural networks to represent the context and relationships between the target word and its context words (Almuzaini and Azmi 2020). An attention mechanism and feature selection using LSTM and character embedding achieve an accuracy of 84.2% in classifying Chinese text (Zhu et al. 2020b). Deep feedforward neural network with the CBOW model achieves an accuracy of 89.56% for fake consumer review detection (Hajek et al. 2020).

LSTM with the Word2Vec model achieves an F1-score of 98.03% for word segmentation in the Arabic language (Almuhareb et al. 2019). Neural network-based word embedding efficiently models a word and its context and has become one of the most widely used methods of word distribution representation (N.H. Phat and Anh 2020)(Alharthi et al. 2021).

Machine learning algorithms such as Naive Bayes classifier (NBC), support vector machine (SVM), decision tree (DT), and the random forest (RF) were famous for information retrieval, document categorization, image, video, human activity classification, bioinformatics, safety and security (Shaikh et al. 2021). Deep learning model such as CNN and GloVe embedding improves citation screening and achieves an accuracy of 84.0% (V Dinter et al. 2021). To classify meaningful information into various categories, the deep learning model GRU with GloVe embedding achieves an accuracy of 84.8% (Zulqarnain et al. 2019). Information retrieval systems are applications that commonly use text classification methods (Greiner-Petter et al. 2020), (Kastrati et al. 2019). Text classification can be used for a variety of purposes, such as the classification of news articles (Spinde et al. 2021), (Roman et al. 2021), (Choudhary et al. 2021), (de Mendonça and da Cruz Júnior 2020), (Roy et al. 2020). The performance of Word2Vec, GloVe, and fastText is compared to match the corresponding activity pair.

The experimental evaluation shows that the fastText embedding approach achieves the F1-score of 91.00% (Shahzad et al. 2019). Extracting meta-textual features and word-level features using the BERT approach gains an accuracy of 95% for classifying insincere questions on question-answering websites (Al-Ramahi and Alsmadi 2021). CNN with the Word2Vec model achieves an accuracy of 90% for text classification tasks (Kim and Hong 2021), (Ochodek et al. 2020). It is challenging to extract discriminative semantic characteristics from text that contains polysemic words. The construction of a vectorized representation of semantics and the use of hyperplanes to break down each capsule and acquire the individual senses are proposed using capsule networks and routing-on-hyperplane (HCapsNet) techniques. Experimental investigation of a dynamic routing-on-hyperplane approach utilizing Word2Vec for text classification tasks like sentiment analysis, question classification, and topic classification reveals that HCapsNet achieves the highest accuracy of 94.2% (Du et al. 2019). A hierarchical attention network based on Word2Vec embedding achieves an accuracy of 84.57% for detecting fraud in an annual report (Craja et al. 2020). Text classification by transforming knowledge from one domain to another using LSTM and Word2Vec embedding model achieves an accuracy of 90.07% (Pan et al. 2019a). Social media tweets analysis (Hammar et al. 2020). Domain-specific word embedding outperforms the BERT embedding model and achieves an F1-score of 94.45% (Grzeża et al. 2020), (Zuheros et al. 2019), (Xiong et al. 2021). Ensemble deep learning model with RoBERT embedding achieves an accuracy of 90.30% to classify tweets for information collection (Malla and Alphonse 2021), (Hasni and Faiz 2021), (Zheng et al. 2020). CNN with a domain-specific word embedding model, achieves an F1-score of 93.4% to classify tweets into positive and negative (Shin et al. 2020).

Text categorization algorithms have been successfully applied to Korean/French/Arabic/Tigrinya/Chinese languages for document/tweets classification (Kozłowski et al. 2020), (Jin et al. 2020). CNN with the CBOW model achieves an accuracy of 93.41% for classifying text in the Trigniya language (Fesseha et al. 2021). LSTM with Word2Vec achieves 99.55% for tagging morphemes in the Arabic language (Alrajhi and ELAffendi 2019). With word2vec, CNN achieves an accuracy of 96.60% on Chinese microblogs. This result demonstrates that word vectors employing Chinese characters as feature components produce better accuracy than word vectors (Xu et al. 2020). The lexical consistency of the Hungarian language can be improved by embedding techniques based on sub-word units, such as character n-grams and lemmatization (Döbrössy et al. 2019). To accurately assess pre-trained word embeddings for downstream tasks, it is necessary to capture word similarity. Traditionally the similarity is determined by comparing it to human judgment. A Wikipedia Agent Using Local Embedding Similarities (WALES) is proposed as an alternative and valuable metric for evaluating word similarity. The WALES metric depends on a representative traversing the Wikipedia hyperlink graph. A performance evaluation of a graph-based technique on English Wikipedia demonstrates that it effectively measures similarity without explicit human labeling (Giesen et al. 2022). A Doc2Vec word embedding model is used to extract features from the text and pass them through CNN for classification. The experimental evaluation of the Turkish Text Classification 3600 (TTC-3600) dataset shows that the model efficiently classifies the text with an accuracy of 94.17% (Dogru et al. 2021). LSTM with CBOW achieves an accuracy of 90.5% for comparing the semantic similarity between words in the Chinese language (Liao and Ni 2021). The review of text classification techniques in terms of data source, application area, datasets, and performance evaluation are illustrated in Table 7 of Annexure A.

5.2 Sentiment analysis

Sentiment analysis determines the sentiment and perspective of points of view in textual data. The problem can be expressed as a binary or multi-class problem. Multi-class sentiment analysis divides texts into fine-grained categories or multilevel intensities, whereas binary sentiment analysis divides texts into positive and negative classes (Birjali et al. 2021). Social communication platforms such as websites, which include comments, discussion forums, blogs, microblogs, and Twitter, are among the sources for sentiment analysis. Sentiment analysis provides information on what customers like and dislike, and the company better understands its product's qualities (Liu et al. 2021b). Using lexicon-based and Word2Vec embedding and a Bidirectional enhanced dual attention model, the aspect-based sentiment analysis task gets an F1-score of 87.21% (Rida-e-fatima et al. 2019). Sentiment analysis includes emotion classification, qualitative or quantitative analysis, and opinion extraction. Consumer data are evaluated to actively analyze public opinion and aid decision-making (Harb et al. 2020), (Vijayvergia and Kumar 2021). Sentiments and opinion analyses are examined at the document level, sentence level, or aspect level (Liu and Shen 2020), (Alamoudi and Alghamdi 2021). Using a hybrid framework of Word2Vec, GloVe, and BOW with an SVM classifier, an extended ensemble sentiment classifier approach achieves an accuracy of 92.88% (Mohamed et al. 2020). Sentiment analysis efficiently determines customer opinion to analyze patient mental health via social media posts (Dadkhah et al. 2021), (Agüero-Torales et al. 2021), (Sharma et al. 2021). An LSTM model with imitated and polarised word embedding yields an F1-score of 96.55% for human–robot interaction (Atzeni and Reforgiato Recupero 2020).

The advancement of big data, cloud technology, and blockchain has broadened the scope of applications, allowing sentiment analysis to be employed in virtually any subject. Customers' impressions of goods or services are evaluated to make informed decisions (Ayu and Khotimah 2019), (Onan 2021). Bidirectional GRU with refined global word embedding achieves an F1-score of 91.3% for the sentiment analysis task (Wang et al. 2021a). Aspect-based sentiment analysis for Arabic/Korean/Russian/Turkish language can efficiently classify text into lexicon-based, machine learning-based, and deep learning-based categories (Song et al. 2019), (Smetanin and Komarov 2021), (Kilimci and Duvar 2020), (Alwehaibi et al. 2021). Sentiment analysis on Arabic Twitter data using domain-specific embedding and the CNN model achieves an accuracy of 73.86% (Fouad et al. 2020).

Researchers confront significant problems, such as handling context, mocking, statements expressing many emotions, expanding Web jargon, and semantic and grammatical ambiguity, despite several moods and emotion recognition approaches (Naderalvojud and Sezer 2020). Establishing an effective technique to express the feeling and emotions of people is a time-consuming undertaking (Hao et al. 2020), (Naderalvojud and Sezer 2020). In a low-resource language, extracting numerous features and emotions from a multi-opinion statement is challenging. Word embedding approaches are used to acquire meanings, compare text, and determine the text's relevance for decision-making (Wang et al. 2021c). Profanity detection using LSTM and fastText achieves an accuracy of 96.15% (Yi et al. 2021). Contextualized word embedding is based on the context of a particular word, and its representation changes dynamically depending on the context. The use of a word embedding strategy in conjunction with deep learning models can detect hate, toxicity, irony, and objectionable content in text and categorise it into a specific category (Kapil and Ekbal 2020), (Alatawi et al. 2021), (González et al. 2020), (Beddiar et al. 2021). Machine learning and deep learning models such as DT, RF, Multilayer perceptron (MLP),

CNN, LSTM, and BiLSTM are compared utilizing Word2Vec, BERT, and a domain-specific embedding technique in terms of performance. The LSTM model with domain trained embedding achieves an accuracy of 95.7% to detect whether reviews on social media contain toxicity comments (Dessi et al. 2021). An offensive stereotype technique is suggested as a systematic way to detect hate speech and profanity on social media platforms. The proposed method locates the quantitative indicator of bias in the pre-trained embedding model, which effectively classifies the text as containing hate speech (Elsafoury et al. 2022). The prejudices connected to various social categories are investigated. The study demonstrates how the biases associated with multiple social categories are mitigated and how they overlap over a one-dimensional subspace for each individual (Cheng et al. 2022). Metric learning is mapping the embedding space that places comparable data adjacent to each other and vice versa. The pre-trained transformer-based language model is suggested to be used self-supervised to generate appropriate sentence embedding. Deep Contrastive Learning for Unsupervised Textual Representations (DeCLUTR) requires fewer trainable parameters. The universal sentence encoder performed well in the unsupervised evaluation of the SentEval task (Giorgi et al. 2021). A deep canonical correlation analysis-based network called the Interaction Canonical Correlation Network is suggested to learn correlations between text, audio, and video. The features that are retrieved from all three modes are then used to create the multimodal embedding, which performs multimodal sentiment analysis and emotion recognition. On the CMU-MOSI movie review dataset, the suggested network attains the best accuracy of 83.07% (Sun et al. 2020b). An unordered structure model is suggested to build phrase embedding for sentiment analysis tasks in various Arabic dialects, independent of the order and grammar of the context's words. On the Arabic Twitter Dataset, the suggested method outperforms others in classifying the sentiment of various dialects with an accuracy of 88.2% (Mulki et al. 2019). To learn the contextual word relationships within each document and the inductive learning of new words. Graph Neural Network (GNN) is created for a document and generates the embedding for all the words in the document. The TextING and Glove are used for inductive learning utilising the GNN. The experimentation is performed on four datasets: the movie reviews dataset, the Reuters newswire 8 and 52 categories dataset, and the cardiovascular diseases dataset. The result shows that the TextING approach achieves the highest accuracy of 98.04% on the R8 dataset in modeling local word-word relations and word significances in the text (Zhang et al. 2020). To predict Bitcoin price using text sentiment, the LSTM model with fastText embedding achieved the most remarkable accuracy of 89.13% compared to Word2Vec, GloVe with RNN and CNN (Kilimci 2020). Compared to GloVe, ELMo with LSTM, the CNN model with BERT embedding extracts linguistic and psycholinguistic information with an accuracy of 72.10% to detect a person's personality (El-Demerdash et al. 2022), and the multilayer CNN model with BERT embedding is 80.35% (Ren et al. 2021). The review of sentiment analysis techniques in terms of data source, application area, datasets, and performance evaluation are illustrated in Table 8 of Annexure A.

5.3 Biomedical text mining

Integrating deep learning and an NLP model in a healthcare environment improves diagnosis. Massive amounts of health-related information are available for processing, including digital text in electronic health records (EHR), medical text on social networks, and text in a computerized report. Image annotation and labeling are done using medical images and radiological reports. NLP can be used to complete annotations and labeling in less time

with less effort. NLP assists in exiting relationships between entities, allowing for a more accurate medical diagnosis (Pandey et al. 2021), (Moradi et al. 2020). The biomedical literature's unique character, quantity, and complexity present challenges for automated classification algorithms. In a multilabel situation, word embedding techniques can be helpful for biomedical text categorization. Medical Subject Headings (MeSH) are represented as ontologies, giving machine-readable labels and specifying the issue space's dimensionality. ELMo embedding-based automated biomedical literature classification efficiently classifies biomedical text and gets an F1-score of 77% (Koutsomitropoulos and Andriopoulos 2021). A biomedical word sense disambiguation strategy using the BiLSTM model obtains a macro average of 96.71% to improve medical text classification (Li et al. 2019b). The BiLSTM model with Word2Vec embedding yields an F1-score of 98% regarding acronyms within the text and is classified into respective diseases. (Magna et al. 2020).

The performance of the deep contextualized attention BiLSTM model utilizing ELMo, fastText, Word2Vec, GloVe, and TF-IDF is compared. The BiLSTM model correctly classifies malignant and normal cells with an accuracy of 86.3% (Jiang et al. 2020a). Using an ontology-based strategy to preserve data-driven and knowledge-driven information in pre-trained embedding enhances the model's similarity measure (Racharak 2021). Domain-specific embedding is used for disease diagnosis to analyze patients' medical inquiries and structured symptoms. The fusion-based technique obtains the maximum accuracy of 84.9% and effectively supports telemedicine for meaningful drug prescriptions (Faris et al. 2021).

The LSTM with the CBOW model achieves the highest accuracy of 94% in recognizing disease-infected people from tweets about disease outbreaks on online social networking sites (OSNS) (Amin et al. 2020). Colloquial phrases are collected from tweets available on OSNS using BERT embedding, and the model achieves an accuracy of 89.95% in categorizing health information. (Kalyan and Sangeetha 2021). An attention-based BiLSTM-CRF (Att-BiLSTM-CRF) model with ELMo achieves an F1-score of 88.78% to efficiently analyze electronic health information and clinical named entity recognition (CNER) challenge (Yang et al. 2019). Similarly, BiLSTM with CRF and BERT embedding performs F1-score of 98.32% for the CNER task (Catelli et al. 2021). EHR analysis for identifying cause and effect relationships using CNN and Att-BiLSTM models achieves F1-score of 52% (Akkasi and Moens 2021). The use of domain-specific embeddings BioWordVec improves visual prognostic predictions from EHR and reaches a 99.5% accuracy (Wang et al. 2021b). Domain-specific embedding, ClinicalBERT enhances the performance of EHR categorization into clinical and non-clinical categories (Goodrum et al. 2020), (Pattisapu et al. 2019). Multi-label classification of health records using bidirectional GRU (BiGRU) and ELMo achieves an accuracy of 63.16% and enhances the EHR classification based on diseases (Blanco et al. 2020). BiLSTM with CRF and GloVe embedding achieves F1-score of 75.62% for biomedical NER tasks (Ning and Bai 2021). In a Spanish clinical case, domain-specific embedding achieves an F1-score of 90.84% to improve NER (Akhtyamova et al. 2020). The CNN with Word2Vec embedding achieves an accuracy of 90.20% in predicting a therapeutic peptide's illness (Wu et al. 2019). A deep learning model such as CNN with Word2Vec embedding achieves an accuracy of 90.31% for predicting protein family (Yusuf et al. 2021). For type III secreted effector prediction, a model combining CNN and Word2Vec embedding and a position-specific scoring matrix for feature extraction obtains an accuracy of 81.20%. (Fu and Yang 2019). An enhancer comprises CNN with a Word2Vec embedding that achieves an accuracy of 77.50% for detecting eukaryotic gene expression. (Khanal 2020). An enhancer made up of a sequence generative adversarial network (GAN) with a Skip-Gram model obtains an accuracy of 95.10% (Yang et al. 2021b). A model comprising an Att-CNN, BiGRU with Word2Vec embedding yields an accuracy of

92.14% in predicting chromatin accessibility (Guo et al. 2020). A model utilizing BERT with language embedding obtained an accuracy of 94% in detecting adverse medication events (Fan et al. 2020). The review of biomedical text mining techniques in terms of data source, application area, datasets, and performance evaluation are illustrated in Table 9 of Annexure A.

5.4 Named entity recognition and recommendation system

Information retrieval, question answering, machine translation, and other downstream applications use NER as a pre-processing step. In an end-to-end multitasking context, word embedding methods like Word2Vec and fastText are used to improve speech translation (Chuang et al. 2021). Cross domains adversarial learning models comprised of CNN, BiLSTM, and Word2Vec embedding are utilized to categorize the information from EHR available in the Chinese language and achieve F1-score of 74.39%. (Wen et al. 2020). The Chinese word embedding-based model with LSTM acquires an F1-score of 95.53% to understand the semantics of words and efficiently analyze the features. (Zhang et al. 2021). A domain-specific word embedding approach with a fuzzy metric that focuses on a unique entity recognition task is proposed to adopt cooking recipes from a set of all available recipes. The model achieves 95% confidence in selecting appropriate recipes (Morales-Garzón et al. 2021). For the Chinese clinical NER task, the LSTM, CRF, and BERT models obtain an accuracy of 91.60% for EHR categorization. (Li et al. 2020b). An LSTM with domain-specific word embedding Tex2Vec is utilized to extract valuable insides from Urdu literature and attain an F1-score of 81.10%. (Khan et al. 2020). The BiLSTM with BERT embedding yields a greater accuracy of 90.84% than the EMLO or GloVe embedding model to perform biochemical named entity identification tasks (Liu et al. 2021a). BiLSTM with domain-specific embedding defined for clinical de-identification on COVID-19 Italian data gains a micro F1-score of 94.48% (Catelli et al. 2020). The localization of software bugs using GloVe and the POS tagging methodology achieved a maximum average precision of 30.70% (Liu et al. 2019). A single neural network model to jointly learn the task of POS and semantic annotation is proposed to enhance the performance of existing rule-based systems for the Welsh language. The proposed approach achieves an accuracy of 99.23% for multitask taggers and improves out-of-vocabulary coverage for the Welsh language using fastText pre-trained embedding (Ezeani et al. 2019). The discontinuous nature of the text is handled using a GAN2vec technique. The suggested method produces real-valued vectors like the Word2Vec paradigm. The discontinuous nature of the text is handled using a GAN2vec technique. The experimental GAN2vec evaluation on the dataset of Chinese poetry yields a BLUE score of 66.08% (Budhkar et al. 2019). An ensemble approach is suggested to classify brief text sequences from the texts of various Arabic-speaking nations. The results of the experiments demonstrate that the performance of the proposed ensemble model is comparable to the Prediction by Partial Matching (PPM) character language model. It obtains an F1-score of 63.4% on the Arabic Dialect Corpus dataset (Lippincott et al. 2019). A sparse self-attention LSTM (SSALSTM) approach is proposed to learn sentiment lexicons from Twitter. The method employs a self-attention approach to determine the sentiment polarity associated with each word, demonstrating that the sparse characters are semantically and emotionally equivalent. The suggested SSALSTM approach effectively determines sentiment polarity and is helpful for named entity recognition. The sentiment-aware word embedding is used for evaluation on the SemEval dataset, which shows that the SSALSTM approach achieves an accuracy of

84.32% to generate the sentiment lexicon (Deng et al. 2019). To recognize a software flaw on large datasets, BiGRU with Dec2Vec yields an F1-score of 96.11%, whereas fastText performs better on short datasets (Jeon and Kim 2021). Drug name extraction and recognition from the text for clinical application are performed using BiLSTM, CNN with CRF, and Sence2Vec embedding and achieve an F1-score of 80.30% (Suárez-Paniagua et al. 2019). The CNN model and Word2Vec embedding create an efficient recommender system for e-commerce applications based on user preferences with an RMSE of 0.863 (Khan et al. 2021). For a word-level NER test in a language mix of English and Hindi, a multichannel neural network model consisting of BiLSTM and Word2Vec embedding gets an F1-score of 83.90% (Shekhar et al. 2019). A hierarchical attention network for reviewing toys and games products requires extracting meaning at the word and sentence level and obtains an accuracy of 85.13% (Yang et al. 2021a). An attention distribution directed information transmission network gets the lowest mean square error of 1.031% (Sun et al. 2020a). Deep learning models are applied to collect relevant characteristics from product reviews on musical instruments, and for the item recommendation job, the model obtains a mean absolute error of 9.04% (Dau et al. 2021). The Word2Vec model recognizes an entity from Chinese news articles and performs public opinion orientation analysis with an accuracy of 87.23% for the product assessment and recommendation task (Wang et al. 2019). A deep learning model such as CNN with Skip-Gram embedding achieves a 94% accuracy for question categorization and entity identification on a Turkish question dataset (Kapil and Ekbal 2020). The review of NER techniques and recommendation system in terms of data source, application area, datasets, and performance evaluation are illustrated in Table 10 of Annexure A.

5.5 Topic modelling

The technique of providing an overview of the themes mentioned in documents is known as topic modeling. For topic modeling and recommendation tasks, the semantic similarity of word vectors is employed to extract keywords. Word2Vec effectively expresses the relationship between job and worker, improving the system's overall performance (Pan et al. 2019b). An ontology-based word embedding is utilized to extract key geoscience terms and gets an F1-score of 40.7% (Qiu et al. 2019). A CNN with Word2Vec is used for bug localization to the associated bug file and yields an accuracy of 81.00% (Xiao et al. 2018). In topic modeling, the Lead2Trend embedding achieves an accuracy of 80% compared to the Skip-Gram model of Word2Vec embedding (Dridi et al. 2019). A multimodal word representation model achieves an accuracy of 78.23%, utilizing syntactic and phonetic information (Zhu et al. 2020a). The feelings and views connected with text in Arabic subjects are utilized for efficient sentiment analysis and topic modeling (Nassif et al. 2021). The learning of bilingual word embeddings (BWE) for the Arabic to English (Ar-En) language pair is investigated using the Bilingual Bag-of-Words without Alignment (Bil-BOWA) model. This model considers different morphological segmentations and various training settings, including sentence length and embedding size. Experimental evaluation shows that increasing the size of word embedding enhances the learning process of Ar-En BWE (Alqaisi and O'Keefe 2019). It is suggested to use multilingual word embedding to represent the lexicon of many languages. The proposed BiLex is tested against English, French, and Spanish texts to pinpoint the precise fine-grained word alignment based on lexical meanings. The outcome demonstrates the BiLex application's effectiveness in obtaining the cross-lingual equivalents of words and sentences in other languages (Shi et al. 2019). As part of the

Multi-Arabic Dialect Applications and Resources (MADAR) shared challenge, LSTM with fastText predicts the Arabic dialect from a collection of Arabic tweets with an accuracy of 50.59% (Talafha et al. 2019). Urdu is a low-resource language that needs a framework for interpretable subject modeling. Pre-trained embedding models, like Word2Vec and BERT, perform well when applied to datasets of Urdu tweets, demonstrating their effectiveness in classifying the text into useful topics (Nasim 2020). For Chinese and English language datasets, a topic modeling based item recommendation approach using sense-based embedding obtains the smallest RMSE of 0.0697 (Xiao et al. 2019). Software vulnerability identification from a vast corpus using domain-specific word embedding achieves 82% accuracy in identifying admitted coding errors (Flisar and Podgorelec 2019). The subject evolution study of scientific literature utilizing Word2Vec and geographical correlation yields a better result, with an RMSE of 3.259 for the spatial lagging model (Hu et al. 2019). The embedding method extracts semantic similarity between terms at a low abstraction level, achieving a standard deviation of 0.5 and reducing the amount of feedback necessary for efficient processing (El-Assady et al. 2020). Word2API embedding maps the relationship between words and APIs and achieves an average mean precision of 43.6% to extract a topic based on relatedness (Li et al. 2018). The review of topic modeling in terms of data source, application area, datasets, and performance evaluation is illustrated in Table 11 of Annexure A.

5.6 Importance of word embedding

In a nutshell, word embedding is the representation of text as vectors. The use of vector representations of text can aid in the discovery of word similarities. With the advancement of embedding techniques, deep learning is currently being employed efficiently in NLP (Verma and Khandelwal 2019) (Wang et al. 2020). The Skip-Gram model of Word2Vec efficiently represents the CNN model's architecture for performing image classification tasks (Dharmaretnam et al. 2021), efficiently explores the semantic correlations in music (Chuan et al. 2020), and effectively utilizing computational resources and parallelizing the technique in shared and distributed memory environment (Ji et al. 2019). Pre-trained embedding models assign similar embedding vectors to Words with similar meanings. A unique embedding should be given to words because their definitions vary depending on their context. The results of an experimental evaluation of a word similarity test demonstrate that the global relationship between the individual words and sub-words effectively represents the word vector. The suggested method minimizes the pre-trained model size while retaining the word embedding standard (Ohashi et al. 2020). An alternative word model called a graph of words is suggested to address the shortcomings of the Bag of Words model. The word order and distance are taken into account by the graph-of-words model. The experiment demonstrates that the graph-of-word model performs well on various tasks, including text summarization, ad-hoc information retrieval, and document keyword extraction (Vazirgiannis 2017). A model utilizing Skip-Gram is presented to determine whether spelling changes impact the effectiveness of word embedding. The study of spelling variation focuses on words with the same meaning but various spellings. In contrast to the non-conventional form, which represents spelling variants, the conventional form represents without spelling variation. The results of the experiment indicate that the word embedding model partially encodes the patterns of spelling variation (Nguyen and Grieve 2020). In contrast to the skip-gram negative sampling (SGNS) technique, which uses both word and context vectors, the context-free (CF) algorithm employs a word

vector. The suggested CF method effectively distinguishes between positive and negative word similarity. It produces results comparable to those of the SGNS algorithm (Zobnin and Elisratova 2019). An isotropic iterative quantization (IIQ) method is suggested for compacting embedding feature vectors into binary ones to satisfy the required isotropic property of pointwise mutual information (PMI)-based approaches. This approach uses the iterative quantization technique, which is well-established for image retrieval (Liao et al. 2020). A method for obtaining vector representations of noun phrases is suggested. Each noun phrase's semantic meaning is assumed to be represented as a vector of the phrase's meaning. The bigram composition method is used to comprehend the semantic meaning of a word, which effectively teaches the importance of a phrase. A specific dimension is essential for improving the phrase's semantic characteristics. Experiment evaluation of proposed constraints on the WordNet dataset efficiently represents the grammatically informed and understandable conceptual phrase vectors (Kalouli et al. 2019). An approach combining principal component analysis and a post-processing algorithm is proposed to minimize the dimensionality of Word2Vec, GloVe, and fastText pre-trained embedding models. The suggested method creates efficient word embeddings in lower dimensions for the binary text classification problem. It achieves the highest Spearman rank correlation coefficient (91.6) compared to other baseline models (Raunak et al. 2019). The reduction of the dimension of word embedding without sacrificing accuracy is achieved using a distillation ensemble strategy, which uses an intelligent transformation of word embedding. The Word2Vec model is used to extract the features, and the LSTM and CNN models are used to train them. The experiment evaluation reveals that the distillation ensemble strategy achieves 93.48% accuracy (Shin et al. 2019). A self-supervised post-processing strategy is suggested to obtain pre-trained embedding for domain-specific tasks, which improves end-task performance by choosing from a menu of reconstructing transformations (MORTY). In a multi-task environment using GloVe embedding, the MORTY technique yields smaller but more consistent benefits and works particularly well with smaller corpora (Rethmeier and Plank 2019). The performance of pre-trained words embedding models such as Word2Vec (CBOW and Skip-Gram), fastText, and the BERT model on a Kannada language text classification task is evaluated. The experimentation evaluation reveals that the CBOW model gives more efficient results than the Skip-Gram model, and the fastText model outperforms the Word2Vec model on the News Classification dataset (Ebadulla et al. 2021). An iterative mimicking (IM) strategy is suggested to treat out-of-vocabulary (OOV) terms. The IM framework iteratively improves the word and character embedding model, assigning a vector to the input sequence for any OOV word. Evaluation of experimental results demonstrates that the suggested framework performs better on the word similarity task than the baseline strategy (Ha et al. 2020). The BiGRU with domain-specific embedding and fastText yields up to 64% micro-average precision for downstream tasks in the patent categorization (Risch et al. 2019). The fastText embedding strategy and the RMSProp optimizer extract relationships between word pairs from the Turkish corpus, with a 90.76% accuracy (Yildirim 2019). The Skip-Gram model shows the highest semantic clustering accuracy with a mean of 6.7 words out of 10 words utilizing Korean word embedding (Ihm et al. 2019), sequence-to-sequence auto encoder efficiently utilized to understand phonetic information using audio Word2Vec embedding (Chen et al. 2019). Gaussian LDA model provides adequate service discovery queries by acquiring meaningful information in the discovery process (Tian et al. 2019). Big corpus scaling is achieved using Word2Vec, a 7.5 times acceleration achieved on GPU without accuracy drop (Li et al. 2019a). The adaptive cross-contextual word embedding model achieves F1-score of 76.9%, considering word polysemy (Li et al. 2021). The LSTM with Word2Vec embedding model efficiently utilizes

the log information to predict the next alarm in process plants and achieves an accuracy of 81.40% (Cai et al. 2019). Mirror Vector Space (MVS) embedding is an ensemble of Concept-Net, Word2Vec, GloVe, and BERT. The MVS model enhances the performance and achieves an accuracy of 83.14% for the text classification task (Kim and Jeong 2021). Improved word vector (IWW) created by combining CNN with Word2Vec, GloVe, Pos2Vec, Lexicon2Vec, and Word-position2Vec improves sentiment analysis task performance and reaches 87% accuracy (Rezaeinia et al. 2019). BiLSTM with CRF and Law2Vec embedding technique for representing legal texts obtains an F1-score of 88% (Chalkidis and Kampas 2019). The Word2Vec embedding with BiLSTM model hyperparameters optimization approaches reaches a classification task accuracy of 93.8% (Yildiz and Tezgider 2021). The meaning of polysemy words is efficiently extracted utilising sentence BERT and improves the overall textual similarity task performance (Wang and Kuo 2020). The examination of pooling procedures in conjunction with basic correlation coefficients produces the best results on subsequent semantic textual similarity problems. It demonstrates the value of applying statistical correlation coefficients to groups of word vectors as a strategy for computing similarity (Zhelezniak et al. 2019). The LDA topic model and Word2Vec are utilized to determine how similar the two terms are. Based on their similarity, the terms' semantic graph is created. By grouping the terms into various communities, each of which serves as a concept, the community detection algorithms are utilised to automatically extract concepts from text (Qiu et al. 2020a). The performance of biometric-based surveillance systems for monitoring user activity is improved using GloVe embedding with the BiLSTM model (Toor et al. 2019). The review of the importance of word embedding in terms of data source, application area, datasets, and performance evaluation is illustrated in Table 12 of Annexure A.

5.7 Deep learning environment

Artificial neural networks gave rise to deep learning technology, which is now a hot issue in computing and is used extensively in a wide range of fields, including cyber security, healthcare, visual identification, and many more. Nevertheless, the dynamic nature and fluctuations of real-world problems and data make it difficult to create an acceptable DL model. Additionally, the absence of fundamental knowledge transforms DL techniques into passive black boxes that limit standard-level advancement. This section gives a concise overview of deep learning techniques and includes a taxonomy that takes important application domains into account.

Deep learning is becoming an increasingly important component of security systems. In the field of computer security, the paper covers the appropriate approaches and the standards for comparing and assessing methods. The performance of deep learning architectures such as MLP, CNN, and LSTM is compared between 4 to 6 layers of different types. Additionally, the study suggests adopting and implementing intrusion detection systems and vulnerability identification techniques in computer security (Warnecke et al. 2020). A dynamic prototype network based on sample adaptation for few-shot malware detection was presented to formalize the identification of unknown malware. The method makes it possible to detect malware by enabling dynamic feature extraction based on sample adaptation and using a metric-based method to determine the distance between the query sample and the prototype. The suggested method performs better than the current few-shot malware detection algorithms (Chai et al. 2022). A deep reinforcement learning-based data poisoning attack approach is developed to aid

hostile personnel in endangering TruthFinder while remaining undetected. The workers experiment with various attack methods and refine their poisoning techniques to maximize their attack strategy and limit information extraction (Li et al. 2020a).

A system called DeepAutoD is suggested to use a deep convolutional neural network, which learns feature information for malicious code identification while removing the influence of reinforcement. The system increases the effectiveness of mobile communication and the security of networked computers (Lu et al. 2022). A unique ensemble deep learning-based web attack detection system is suggested to protect IoT network environments from web attacks. The three distinct deep learning models, such as the MRN, LSTM, and CNN, first identify web attacks independently before coming together as an ensemble to create an ensemble classifier that will ultimately determine the outcome. The feature vector is formed using TF-IDF, word2vec, and FastText. The experimental results on the HTTP CSIC dataset demonstrate that the proposed ensemble system can accurately identify online attacks with low false positive and negative rates (Luo et al. 2021).

For spatiotemporal data mining applications, deep learning models like CNN and RNN have shown amazing success (Wang et al. 2022). Deep learning models like CNN are used for feature extraction from spatial–temporal data, while GRU is used to improve query trajectory prediction accuracy. The investigations on the Porto dataset demonstrate that the suggested model achieves a mean absolute percentage error of 0.070% while approximating the properties of each segment of trajectory data at the time level (Qiu et al. 2020b). Identifying information that discriminates based on gender depends heavily on the meaningful classification of text from digital media. Word embedding is done using the ELMo and GloVe models, and sentence embedding is done using a BERT model. The experimentation shows that the suggested deep learning models effectively complete multi-label classification (Parikh et al. 2019).

Knowledge graphs, particularly domain knowledge graphs, are already playing significant roles in the field of knowledge engineering and serving as the foundation for intelligent Internet applications that are knowledge-driven. A Graph Convolutional Network (GCN) (Kipf and Welling 2017) is a multilayer neural network that specifically focuses on a graph and generates embedding vectors of nodes based on the characteristics of their neighborhood to accomplish state-of-the-art categorization.

GCN is suggested as a method for classifying text. The Text-GCN learns the embedding for both words and documents after initializing with a one-hot representation of each. The experimental outcome demonstrates Text-GCN's tolerance to minimal training data in text classification. Text-GCN can effectively use the limited labeled documents and collect information on global word co-occurrence (Yao et al. 2019). For text categorization, the graph-of-docs paradigm is proposed to represent numerous documents as a single graph. The suggested method recognizes a term's significance across the board in a document collection and encourages the inclusion of relationship edges across documents. Experimental results demonstrate that the suggested model outperforms the baseline models with an accuracy score of 97.5% (Giarelis et al. 2020).

Graph-based NLP combines the structural information in text and the representation learning ability of deep neural networks. Graph-based NLP approaches are extensively used in text clustering and multitask learning (Wu et al. 2022). Deep neural networks are suggested to produce compositional word embedding and sentence processing. The model multiplies matrices to create unitary matrices for big units that encode lexical data. These lexicons depict the embedding without diluting the information or considering the context (Bernardy and Lappin 2022).

Table 5 The most prominent word embedding models published from 2013 to 2020

Embedding approach	Researchers	Year	Organization	References	Citations
Word2Vec	Tomas Mikolov et al	2013	Google Inc	Mikolov et al. (2013a)	27,809
				Mikolov et al. (2013b)	32,592
GloVe	Jeffrey Pennington et al	2014	Stanford University	Jeffrey Pennington, Richard Socher (2014)	26,851
fastText	Armand Joulin et al	2016	Facebook AI Research Lab	Joulin et al. (2017)	3766
ELMo	Matthew E. Peters et al	2017	Allen Institute of AI	Peters et al. (2018)	9495
GPT	Alec Radford et al	2018	OpenAI	Radford et al. (2018)	3152
BERT	Jacob Devlin et al	2018	Google AI Lab	Devlin et al. (2019)	35,802
GPT2	Alec Radford et al	2018	OpenAI	Radford et al. (2019)	2987
GPT3	Tom B. Brown et al	2020	OpenAI	Brown et al. (2020)	3287

GloVe: global vectors for word representation, *ELMo*: embeddings from language models, *GPT*: generative pre-trained transformer, *BERT*: bidirectional encoder representations from transformers

6 Remarks and critical discussion

The selection of appropriate word embedding methods and deep learning models in text analytics is essential. This research aims to look at the steps different word embedding methods take and the behavior of various deep learning models in terms of text analytics task performance. In this part, the study's practical implications are examined. The advancement in the deep learning model approaches directly affects the growth of NLP techniques. The in-depth analysis of methods for analyzing unstructured text includes text classification, sentiment analysis, NER and recommendation system, biomedical text mining, and topic modeling, as shown in Fig. 3. Each of these strategies is employed in a variety of contexts.

6.1 The model architecture used for word embedding

Complex deep neural network models are becoming easier to train as technology advances on hardware and software fronts. As a result, researchers have begun integrating the characteristics of numerous deep neural networks and adding some innovative features to their design. Section 1 discusses the architectural constraints used in developing deep learning models. Section 2 discusses the development of word embedding methods for efficiently and accurately representing the word's meaning. The most prominent word embedding models discussed in section 2 are summarized in Table 5, and their citation counts.

It is observed from Table 5 the paper that proposed the Word2Vec embedding model has the highest citations among all other models. The Word2Vec model assigns probabilities to terms that perform well in word similarity tests. In contrast, the GloVe is a count-based model that combines the local context window approach and global matrix

factorization approaches. The GloVe model was proposed in 2014 and had a considerable number of citations representing their utilization by the researchers. The current review reflects the same information about the Word2Vec and GloVe models as shown in Fig. 18, indicating that the researchers have explored the performance of both models to perform a specific task in almost all domains. Each language consists of specific rules and patterns that require the base model to be modified for better results. The models learn static word embeddings, with each word's representation determined after training. The performance of the embedding model is enhanced to handle out-of-vocabulary words and the proposed fastText model. The fastText is a Word2Vec extension that recognizes words as character n-grams. It generates an efficient and effective vector representation of infrequent words.

Embedding models are further enhanced to handle polysemy words and represent the word's contextual meaning for a different language to perform more domain-specific related tasks. A polysemy word's meaning might change depending on the situation. Each word's vector representation can be altered in a contextualized word embedding approach depending on the input contexts, sentences, or documents. Domain-specific word embedding, on the other hand, is an effective strategy for task analysis for specific domain activities in research. The DSWE has grown as a more valuable solution than general word embedding since it concentrates on one particular aim of text analytics, as shown in Fig. 18.

BERT contextual embedding model has the most citations of all the recently published models in citation counts. The current review on embedding models for text analytics tasks shows that the researchers deeply explore the BERT model compared to ELMo and GPT models. Recently proposed, a variant of the GPT model was also utilized to perform domain-specific operations and is expected to achieve more citation and exploration among researchers. The description, benefits, and drawbacks of various word representation models are discussed in Table 14 of Annexure B. As per the current review, several model designs and methodologies have emerged to perform text analytics tasks. The remaining section summarizes, contrasts, and compares numerous word embedding and deep learning models and presents a detailed understanding of how to use these models to achieve efficient results on text analytics tasks.

6.2 Comparative analysis of word embedding models for text analytics tasks

The performance of word embedding techniques and deep learning models for various text analytics tasks observed from the current review is shown in Fig. 19. The study shows that the domain-specific word embedding performance is higher than the generalized embedding approach for performing domain-specific tasks related to text analytics. Specifically for the text classification task, the CBOW model of Word2Vec and domain-specific embedding performance is similar in the current review. The GloVe, fastText, and BERT embedding models show considerable performance and are limited to a few applications. The researchers utilize the ELMo and GPT models for text classification tasks in minimal circumstances, as per the current review.

Domain-specific word embedding is the preferred choice of the researchers to perform a task related to sentiment analysis. The researchers focus on character, word, or sentence levels to identify sentiment associated with the text. The performance of domain-specific embedding, which focuses on specific granules of text for evaluation, is higher than the generalized embedding approach, as shown in Fig. 19(a). The CBOW and BERT model also performed efficiently, considering specific evaluation features to identify sentiments. The researchers determined that the GloVe and fastText models also performed well for a

limited number of situations. In contrast, the performance of the ELMo and GPT model is not competitive compared to the BERT model for sentiment analysis tasks as per the current review.

Generalized word embedding models fail to capture the ontologies information available in domain-specific structured resources. The subword information from unlabeled biomedical text is combined with MeSH vocabulary to form a BioWordVec domain-specific word embedding, which creates an essential foundation for biomedical NLP. As per the current review, the researchers use domain-specific embedding as an efficient approach for biomedical text mining classification, as shown in Fig. 19(a). The CBOW, ELMo, and BERT embedding models are also good choices for biomedical text mining following a generic approach. The researchers utilize the CBOW and domain-specific word embedding to perform the named entity recognition and recommendation tasks. The other embedding models, such as Skip-Gram, GloVe, fastText, and BERT, are also explored and give better results for a limited number of situations, as shown in Fig. 19(a). The researcher utilizes domain-specific embedding heavily for the topic modeling task compared to Skip-Gram and ELMo embedding models.

It is observed from the review that CBOW and domain-specific word embedding models are used frequently by researchers. It performs better in analyzing word embedding models' impact on domain-specific text analytics. At the same time, the other models, such as Skip-Gram, GloVe, fastText, and BERT, are also explored for the possibility of a better outcome in a few instances.

6.3 Comparative analysis of deep learning models for text analytics tasks

The performance of deep learning models in various application areas is shown in Fig. 19(b). It is found from the current review that the researchers heavily recommend the CNN model to perform text classification tasks. The LSTM model is another alternative to efficiently perform text classification tasks, whereas there are few instances where the GRU or hybrid model achieves better performance. The LSTM model is strongly recommended for sentiment analysis tasks, and the CNN model can be another alternative for the same. Researchers discovered that both CNN and LSTM could be used for text classification tasks in the biomedical domain. The LSTM model is strongly recommended for named entity recognition and recommendation system tasks, as shown in Fig. 19(b), based on the model's performance. The other deep learning models, such as GRU, CNN, and hybrid models, prove their effectiveness in a few cases. The CNN and GRU models can be utilized for topic modeling tasks. It is observed from the current review that analyzing the impact of the word embedding model on the text analytics domain needs a powerful deep learning model. The LSTM model is preferred for analyzing the performance of the embedding model compared to the CNN and GRU models. Apart from the LSTM model, the CNN model can also be explored to perform the analysis.

6.4 Selection criteria for word embedding and deep learning models to perform text analytics tasks

Text analytics uses machine learning, deep learning, and NLP to extract meaning from vast amounts of text. Businesses may use this information to boost revenue, customer satisfaction, innovation, and public safety. This study explores the effectiveness of

Table 6 A reference for selecting a suitable word embedding approach and deep learning model for text analytics tasks

Application area	Embedding model							Deep learning model			
	Word2Vec		Glove	BERT	fastText	ELMo	DSWE	CNN	LSTM	Hybrid	GRU
	CBOW	Skip-Gram									
TC	✓		✓	✓	✓		✓	✓	✓		
SA	✓			✓			✓		✓	✓	
MTC	✓			✓		✓	✓	✓	✓	✓	
NER & RS	✓						✓	✓	✓	✓	
TM		✓	✓				✓	✓			✓
IWE	✓	✓					✓	✓	✓		

☑ : First preference, ✓ : Second preference

TC: text classification, SA: sentiment analysis, MTC: medical text classification, NER & RS: named entity recognition and recommendation system, TM: topic modeling, IWE: impact of word embedding, DSWE: domain-specific word embedding

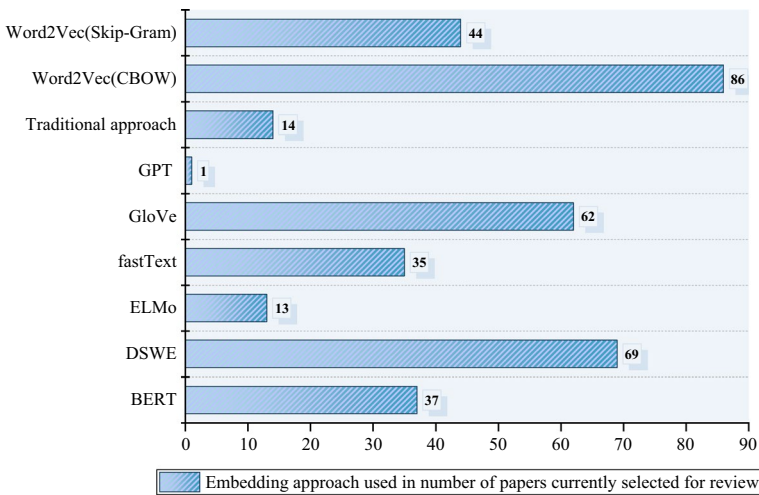


Fig. 18 Overview of the embedding approach used by the researchers

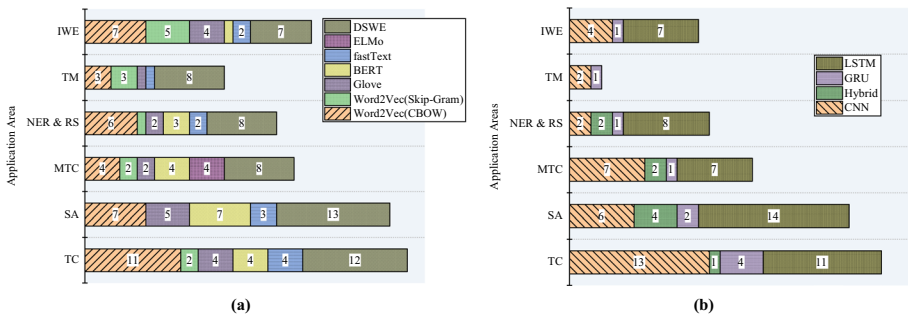


Fig. 19 Based on the current review, (a) performance of word embedding models and (b) performance of deep learning models for various text analytics tasks

utilizing word embedding techniques in a deep learning environment for text analytics tasks. The review reveals three main types of word embeddings: conventional representation, distributional representation, and contextual representation model. Deep learning models such as CNN, GRU, LSTM, and a hybrid approach are utilized by most researchers to accomplish text analytics tasks. The selection of word embedding and deep learning models for better outcomes is a vital step. It requires thorough knowledge of various types of embedding and deep learning models to accomplish the designated task in a specified time. A reference selection criteria for selecting a suitable word embedding and deep learning model for text classification tasks is illustrated in Table 6. It is revealed from the current review that domain-specific word embedding achieves the first preference as the most suitable embedding for the majority of application areas related to text analytics.

The CBOW model also achieves the first preference for performing text classification tasks, whereas GloVe, fastText, and BERT models achieve the second preference, as shown in Table 6. The CBOW and BERT model achieves the second preference for performing the sentiment analysis task. The CBOW, BERT, and ELMo models achieve second preference for performing biomedical text mining tasks. The CBOW model is the second choice for performing operations on the NER and recommendation system. The Skip-Gram and GloVe model achieves the second preference to perform topic modeling-related tasks. The domain-specific word embedding and CBOW embedding models are recommended as the first preferences, whereas the Skip-Gram model is recommended as a second preference to analyze the impact of the word embedding model on text analytics tasks.

Various deep learning models have been proposed and utilized to perform text analytics tasks. It is revealed from the current review that the CNN model achieves the first preference and the LSTM model attains the second preference to perform text classification tasks. Similarly, the LSTM model reaches the first preference for sentiment analysis tasks, named entity recognition and recommendation system tasks, and the hybrid approach is the second preference. The CNN and the LSTM model achieve the first preference for biomedical domain text classification tasks, and the hybrid approach achieves the second preference. The CNN and the GRU model attain the first preference for topic modeling tasks. As per the current review, for analyzing the impact of word embedding, the LSTM model achieves the first preference, and the CNN model achieves the second preference.

In the current review, comparing the performance of various word embedding and deep learning models for text analytics tasks reveals specific word embedding and deep learning models as the preferred choice to perform particular tasks. In conclusion, using the domain-specific word embedding and LSTM model can improve the overall performance of text analytics tasks.

7 Conclusion and future directions

7.1 Concluding remarks

In recent years, there has been an increase in interest in using word embedding and deep learning for analysis and prediction, and the research community has proposed various

approaches. This paper studies a systematic literature review to capture the state-of-the-art word embedding and deep learning models for text analytics tasks and discusses the key findings.

Three different electronic data sources were used to find and classify relevant articles about the influence and use of the word embeddings model on text analytics in a deep learning context. The relevant literature is categorized based on criteria to review the key applications of text analytics and word embedding techniques. Techniques for analyzing unstructured text include text classification, sentiment analysis, NER, recommendation systems, biomedical text mining, and topic modeling.

Deep learning models utilize multiple computing layers to learn hierarchical representations of data. Several model designs and methodologies have emerged for text analytics. This paper reviews the performance of various word embedding methodologies proposed by the researchers and the deep learning models employed to get better results. The review contains a summary of prominent datasets, tools, and APIs available and a list of notable publications. A reference for selecting a suitable word embedding approach and deep learning model for text analytics tasks is presented in Section 6. The comparative analysis is presented in both tabular and graphical forms.

According to the current review, domain-specific word embedding is the first preference for performing text analytics tasks. The CBOW model can be the first preference for performing operations like text classification tasks or analyzing the impact of word embedding. The CBOW model and the BERT model attain the second preference for performing the operations related to text analytics. The review shows that the researchers preferred CNN and LSTM models compared to the GRU and the hybrid approach to perform text analytics tasks. It can be concluded from the findings of this study that domain-specific word embedding and the LSTM model can be used to improve overall text analytics task performance.

7.2 Future directions

The selection of appropriate word embedding models plays an important role in the success of NLP applications. It is difficult to predict what kind of semantic or syntactic information is captured inherently in a contextualized word embedding. Extraneous tasks are the only way to evaluate contextualized word embeddings. It would be crucial to identify whether the goal of context-dependent representation has been achieved and assess the scope of this possible achievement. The expression of each embedding strongly depends on individual tasks for sentence representations. The essential basic components of the sentence required by various tasks are at different levels. It is necessary to understand how to learn sentence representations and even higher levels of text representation for various languages in the future.

Moreover, even though the present word vector model has generated significant results in various NLP tasks, these approaches have some limitations. For example, the model parameters are excessively huge, the lengthy training process, and existing neural network-based systems are incomprehensible. As a result, figuring out how to cut the cost of neural network training while improving the model interpretability is another area of research.

Sizes of the corpus should be considered when evaluating the embedding. Analyze the outcomes of reducing the embedding dimension and the steps that must be followed for a particular task in a given domain.

Pretrained embedding models have a large number of word vectors and need more storage space. On a system with limited resources, this expense represents a deployment constraint. Examine the best ways to increase isotropy and decrease dimension in pretrained embedding models. Investigate approaches for learning multilingual lexicons in a single embedding space, enhance ways for learning multilingual word embedding, and employ semantic information to transmit knowledge in a range of cross-lingual NLP tasks.

Contextualized word embeddings have achieved outstanding results in significant NLP tasks. Further research is required to develop a reliable contextual language model for the text analytics problem using a combination strategy leveraging the contextual word embedding model and multitask learning approach. Contextual embeddings and other sorts of spelling variation can be investigated in future studies. Investigate various classifiers and feature representations to capture the interaction between two embeddings for diagnostic classifiers. Explore how to get the correlation between text, audio, and video using enhanced deep canonical correlation analysis. These distinctive features are collected to provide multimodal embedding for the optimum downstream task. Extend the performance of the transformer-based language model to generate representation, reducing the dependency that requires human-labeled training data and efficiently extending for performing other downstream tasks.

8 Appendix A

Text analytics techniques include text classification, sentiment analysis, biomedical text mining, named entity recognition, recommendation system, and topic modeling. In terms of data source, application area, datasets, and performance evaluation, Tables 7, 8, 9, 10, 11, and 12 illustrate the approaches-wise reviews of word embedding and deep learning models employed.

Table 7 Review of text classification

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
1.	Craja et al. (2020)	Annual report analysis for fraud detection	EDGAR database	LR, RF, SVM, XGB, ANN, HAN	Word2Vec	HAN achieves an accuracy of 84.57%
2.	Alharthi et al. (2021)	Arabic text low-quality content classification	Twitter dataset	CNN, LSTM	Word2Vec, AraVec	LSTM achieves an accuracy of 98%
3.	Kozlowski et al. (2020)	French social media tweet analysis for crisis management	French dataset	SVM, CNN	fastText, BERT, French FlauBERT	FlauBERT achieves a micro F1-score of 85.4%
4.	Zuheros et al. (2019)	Social networking site tweet analysis for the use of polysemic words	Social media texts, both English and Spanish data	XGBoost, HAN, LSTM	GloVe	LSTM + GloVe achieves an F1-score of 97.90%
5.	Liao and Ni (2021)	Semantic similarity between the word in the Chinese language	Manually collected datasets from students	SVM, LR, RF, CNN, LSTM	Word2Vec (CBOW, Skip-Gram)	LSTM + CBOW achieves an accuracy of 90.5%
6.	Ochodek et al. (2020)	Estimation of software development	Projects data of Poznan University of Technology Company	CNN	DSWE	DSWE model achieves an accuracy of 45.33%
7.	El-Alami et al. (2021)	Arabic text categorization	OSAC datasets (corpus including BBC, CNN, and OSAC)	SVM, MLP, CNN, LSTM, BiLSTM	ULMFIT, EMLo, AraBERT	AraBERT achieves an accuracy of 99%
8.	Elmagar et al. (2020)	Arabic text classification	SANAD, NADIA	GRU, LSTM, CNN, BiGRU, BiLSTM, HAN	Word2Vec	GRU model achieves an accuracy of 96.94%
9.	Shaikh et al. (2021)	Bloom's learning outcomes classification	Sukkur IBA university dataset, Najran University, Saudi Arabia dataset	SVM, NB, LR, RF, RNN, LSTM	Word2Vec, fastText, DSWE, GloVe	LSTM + DSWE achieves an accuracy of 87%
10.	Zhu et al. (2020b)	Character embedding for Chinese short text classification	THUCNews dataset, Toutiao dataset, Invoice dataset	RNN, LSTM, HAN	Chinese character embedding (AFC)	LSTM + AFC achieves an accuracy of 84.2%

Table 7 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
11.	Roman et al. (2021)	Citation Intent Classification	Citation Context Dataset, Sci-Cite dataset	HDBSCAN	GloVe, BERT	Kmeans clustering + BERT achieves a precision of 89%
12.	Dinter et al. (2021)	Citation screening to improve the systematic literature review process	20 publicly available datasets	CNN	GloVe	CNN model achieves an accuracy of 88%
13.	Hammar et al. (2020)	Classification of Instagram posts from the fashion domain	Corpora of Instagram posts, Word-Sim353, SimLex-999, FashionSim	CNN	fastText, Word2Vec, GloVe	CNN + fastText achieves an F1-score of 61.00%
14.	Spinde et al. (2021)	Classification of news articles to detect bias-inducing words	News articles	LDA, NB, SVM, KNN, DT, RF, XGBoost, MLP	TF-IDF, LIWC	Achieves F1-score of 43%
15.	Zulqarnain et al. (2019)	Classify meaningful information into various categories	Google snippets dataset, TREC dataset	GRU, RNN, LSTM	GloVe	GRU + GloVe achieves an accuracy of 84.8%
16.	Almuzaini and Azmi (2020)	Classifying Arabic documents	Arabic news texts corpus, Saudi press agency corpus	CNN, LSTM, GRU, BiLSTM, BiGRU	Word2Vec	GRU + Skip-Gram model achieves F1-score of 97.76%
17.	Almuhareb et al. (2019)	Classifying Arabic documents	Arabic Treebank dataset, ATB elitics segmentation schema	BiLSTM	Word2Vec	LSTM + Word2Vec achieves an F1-score of 98.03% for word segmentation
18.	Kastrati et al. (2019)	Classifying educational content for various search and retrieval applications	MOOC platform, Coursera dataset	LDA, SVM, DT, NB, XGBoost, CNN	BoW, Word2Vec, GloVe, fastText	CNN + fastText achieves an F1-score of 91.55%

Table 7 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
19.	Shin et al. (2020)	Detecting cyber security intelligence in Twitter	Curated data, OSINT data, background knowledge	CNN, LSTM	DSWE	CNN + DSWE model achieves F1-score – 93.4%
20.	Hajek et al. (2020)	Fake consumer review detection	Hotel, restaurant, doctor, and Amazon datasets	DFNN, CNN	Word2Vec(Skip-Gram)	DFNN + CBOW achieves the highest accuracy of 89.56%
21.	Choudhary et al. (2021)	Fake news classification	George McIntire dataset, Kaggle, FakeNewsNet repository	CNN	BERT, GloVe, ELMo	BERT + CNN achieves an accuracy of 97.45%
22.	Pan et al. (2019a)	Improve text classification by transforming knowledge from one domain to another	Netease and Cnews are two public Chinese text classification datasets, English text datasets Yahoo dataset	SVM, LSTM	TF-IDF, BOW, Word2Vec	LSTM + Word2Vec achieves an accuracy of 90.07%
23.	Jin et al. (2020)	Korean historical documents analysis	Korean historical documents	Dynamic word embedding approach	BERT	NER task achieves an F1-score of 68%
24.	Fesseha et al. (2021)	Low-Resource Languages: Tigrinya	Tigrinya news datasets	CNN fastText	Word2Vec(CBOW, Skip-Gram)	CNN + CBOW achieves an accuracy of 93.41%
25.	Shahzad et al. (2019)	Matching corresponding activity pairs	PMMC-2015 dataset, University admissions, Birth registration, Asset management datasets	Syntactic and semantic similarity measure approach	Word2Vec, GloVe, fastText	fastText achieves F1-score of 91.00%
26.	Greiner-Petter et al. (2020)	Mathematical information retrieval	STEM documents, DLMF	MLP	Word2Vec, DSWE	DSWE achieves better performance
27.	Zheng et al. (2020)	Measuring the soft power of social entities	Chinese news articles	Soft power measurement framework	Word2Vec	The probability-based approach is efficient
28.	Xiong et al. (2021)	News clicks prediction based on timeliness and attractiveness	Dataset, Toutiao news dataset	BiGRU	Word2Vec	BiGRU + Word2Vec achieves an accuracy of 84.59%

Table 7 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
29.	Alrajhi and ELAffendi (2019)	Part-of-speech tagging for the Arabic language	Quranic Arabic corpus data set	RNN, LSTM	Word2Vec	LSTM + Word2Vec tagger achieves 99.55% for tagging morphemes
30.	Dogru et al. (2021)	Text classification	Turkish Text Classification 3600 (TTC-3600) dataset, BBC-News dataset	CNN	Doc2Vec	CNN with Doc2Vec model achieves an accuracy of 94.17%
31.	Al-Ramahi and Alsmadi (2021)	Question-Answer classification	Quora website, Dataset of Wikipedia comments	Meta and word-level analysis	Word2Vec, GloVe, fast-Text, BERT, TFIDF	Classification using BERT achieves an accuracy of 95%
32.	Roy et al. (2020)	SMS text classification	UCI repository, SMS corpus	NB, RF, GB, SGD, LSTM, CNN	GloVe	CNN + GloVe achieves an accuracy of 99.44%
33.	Jang et al. (2020)	Text Classification	Internet movie review database	Bi-LSTM + CNN	Word2Vec(Skip-Gram)	CNN + BiLSTM + Word2Vec achieves an accuracy of 90.2%
34.	de Mendonça and da Cruz Júnior (2020)	Text classification base on contextual information	Dataset from justice prosecutor office Brazilian public ministry	SVC, KNN, GBM, DT, MLP, XGBoost, CNN	Word2Vec	CNN achieves an accuracy of 82.91%
35.	Kim and Hong (2021)	Transportation-related text classification	Bostons public dataset	CNN	TF-IDF, Word2Vec	CNN achieves an accuracy of 90%
36.	Hasni and Faiz (2021)	Tweets analysis for geolocation	Tweets from the UK and the USA for the last two weeks of March 2021	BiLSTM	Word2Vec, fastText, Char2Vec	BiLSTM+ fastText achieves an accuracy of 56.20%
37.	Malla and Alphonse (2021)	Twitter tweet analysis for the disease information collection	COVID-19 labeled English dataset from Twitter	Majority voting based ensemble deep learning model	RoBERT, BERTweet, CT-BERT	RoBERT achieves an accuracy of 90.30%
38.	Phat and Anh (2020)	Vietnamese text classification	Vietnamese news articles	LSTM, CNN, SVM, NB	Word2Vec	LSTM + Word2Vec achieves an F1-score of 95.74%

Table 7 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
39.	Grzeża et al. (2020)	Social networking site tweets analysis for identification of alcohol-related tweets	Datasets DSI-Q1, Q2, Q3	SVM, XGBoost, CNN, BiLSTM	DSWE(Drink2Vec), BERT	CNN + Drink2Vec achieves an F1-score of 94.45%

SANAD Single-label Arabic news articles datasets, *NADiA* News articles datasets in Arabic with multi-labels, *HAN* Hierarchical attention network, *HDBSCAN* Hierarchical Density-Based Spatial Clustering of Applications with Noise, *LDA* Logistic regression, linear discriminant analysis, *QDA* Quadratic discriminant analysis, *NB* Naïve Bayes, *SVM* Support vector machine, *KNN* k-nearest neighbor, *DT* Decision tree, *RF* Random forest, *XGBoost MLP* Multilayer perceptron, *LWC* Linguistic Inquiry and Word Count features, *NER* Named entity recognition, *PMMC* Process model matching contest dataset, *DLMF* Digital Library of Mathematical Functions, *GB* Gradient Boosting, *SGC* Stochastic Gradient Descent, *HAN* Hierarchical attention network, *DFFN* Deep feed-forward neural network.

Table 8 Review of sentiment analysis

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
1.	Kapil and Ekbal (2020)	Hate and offensive text analysis	Hate & Offensive twitter dataset, Racist & Sexist Twitter dataset, Aggression Facebook and Twitter dataset, OLID Dataset, Harassment dataset	CNN, LSTM, GRU	Word2Vec	Macro F1-score—89.30%
2.	Alwehaibi et al. (2021)	Arabic short sentiment analysis	AraSenTi dataset	CNN, LSTM	Arabic fastText	The LSTM + CNN model achieves an accuracy of 96.7%
3.	Liu and Shen (2020)	Aspect-based sentiment analysis	SemEval2014 restaurant and laptop datasets, four Chinese datasets, Tweet dataset	GANN, GTR, SVM, LSTM, BiLSTM	Word2vec, GloVe	GANN achieves an accuracy of 89.17% on the Phone category
4.	Alamoudi and Alghamdi (2021)	Aspect-level sentiment classification	Yelp, SemEval-2014, META-SHARE website dataset	LR, NB, CNN	GloVe BERT ALBERT	CNN + GloVe achieves an accuracy of 83.04%
5.	Rida-e-fatima et al. (2019)	Identify the correlation between aspects and opinions	SemEval Challenge 2014 (Restaurant and Laptop), SemEval Challenge 2015 (Restaurant)	LSTM, Refined Dual Attention Model (RDAM) and Bi-directional RDAM (B-RDAM)	Word2Vec	BRDAM + Word2Vec achieves an F1-score of 87.21%
6.	Giorgi et al. (2021)	Deep Metric Learning	SemEval dataset, OpenWebText, Stanford Natural Language Interface (SNLI) dataset	Universal sentence embedding	Transformer-small and Transformer-base	DeCLUTR-base model achieves an accuracy of 88.82%
7.	Sun et al. (2020b)	Multimodal embedding for sentiment analysis and emotion recognition	CMU-MOSI, CMU-MOSEI, IEMOCAP	LSTM, CNN	BERT	The proposed ICCN network achieves an accuracy of 83.07%

Table 8 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
8.	Mulki et al. (2019)	Sentiment analysis tasks	Arabic Twitter Dataset, Tunisian Sentiment Analysis Corpus (TSAC)	LSTM	Word2Vec, Doc2Vec	The proposed n-gram embedding approach achieves an accuracy of 88.2%
9.	Zhang et al. (2020)	Identify the relation between words	Movie reviews dataset, the Reuters newswire 8 and 52 categories dataset, and the cardiovascular diseases dataset	GCN	Glove	GCN with GloVe achieves the highest accuracy of 98.04% on the R8 dataset
10.	Kilimci (2020)	Bitcoin price estimation	English Twitter dataset	RNN, LSTM, CNN	Word2Vec, GloVe, fastText	LSTM + fastText achieves an accuracy of 89.13%
11.	Wang et al. (2021a)	Comparable entity identification	Chinese Wikipedia data	Identifying comparable entities approach	Word2Vec	The proposed model achieves a precision of 52.37%
12.	Dadkhah et al. (2021)	Efficient categorization of social media text	LJAR dataset, Scrappy dataset, news-based dataset	DT, RF, KNN, AdaBoost, KNN, NB, SVM, MNB, LSTM	ULMFIT	The unsupervised learning model can be more effective and feasible for solving
13.	Vijayvergia and Kumar (2021)	Emotion detection	Twitter dataset	LSTM, CNN	GloVe	LSTM + CNN + GloVe achieves an accuracy of 86.16%
14.	Mohamed et al. (2020)	Enhanced ensemble sentiment classifier framework	UCI, ProsCons, Opinosis, UMICH dataset	SVM	Word2Vec, GloVe	The hybrid framework of Word2Vec + GloVe + BOW with SVM classifier achieves an accuracy of 92.88%
15.	Alatawi et al. (2021)	Hate speech detection	Stormfront dataset, Twitter dataset	Logistic Regression (LR), BiLSTM	DSWE, Word2Vec, BERT	The BiLSTM + BERT achieves an F1-score of 80%
16.	González et al. (2020)	Irony detection in Twitter	English language (SemEval 2018), Spanish language (IronSVA shared task)	Deep Averaging Networks (DAN), LSTM	DSWE	DSWE performance is efficient

Table 8 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
17.	Arzeni and Reforgiato Recupero (2020)	Mimicked and polarized word embeddings for human-robot interaction	ESWC 2018 Challenge	Bi-LSTM	Word2Vec, SentiWords, DSWE	LSTM + DSWE achieves an F1-score of 96.55%
18.	Beddier et al. (2021)	Social networking site tweets analysis	AskFm corpus, Formspring dataset, Olid, and Wikipedia toxic comments dataset	CNN, LSTM	fastText	LSTM achieves an F1-score of 97.2%
19.	Ren et al. (2021)	Personality detection	Myers-Briggs, Type Indicator (MBTI) datasets	GRU, LSTM, CNN	BERT, GloVe	CNN + BERT achieves an accuracy of 80.35%
20.	Yi et al. (2021)	Profanity Detection	SNS posts dataset, Naver movie review dataset, Twitter dataset	CNN, LSTM, GRU	fastText	LSTM achieves an accuracy of 96.15%
21.	Wang et al. (2021c)	Refined Global Word Embeddings	SemEval, SST1, SST2, IMDB, Amazon, Yelp-2014	CNN, LSTM, BiGRU	Word2Vec, GloVe	BiGRU + RGWE model achieves F1-score of 91.3%
22.	Birjiali et al. (2021) and Agüero-Torales et al. (2021)	Review paper on sentiment analysis	Sentiment140, a Large movie review dataset	Overview of approaches	Word2Vec, GloVe	Overview of sentiment analysis and its related approaches
23.	Fouad et al. (2020)	Sentiment analysis on Arabic Twitter data	Arabic sentiment tweets dataset, AraSenti dataset	CNN	ArWordVec(DSWE)	CNN + ArWordVec(Word2Vec-SG) achieves an accuracy of 73.86%
24.	Onan (2021)	Sentiment analysis on product reviews obtained from Twitter	Twitter product review corpus	CNN, LSTM	Word2Vec, fastText, GloVe, LDA2Vec, and DOC2Vec	CNN + LSTM + GloVe gives higher performance
25.	Naderalvojud and Sezer (2020)	sentiment-aware word embeddings	SemEval-2013 and SST	CNN, LSTM, BiLSTM, LR	Word2Vec(CBOW, Skip-Gram), fastText, SAWE	BiLSTM + SAWE achieves an accuracy of 87.00%

Table 8 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
26.	Hao et al. (2020)	Word polarity and occurrence information	Amazon website book, electronics product reviews, IMDb, Yelp	DANN, HATN	GloVe	Cross-domain sentiment analysis task achieve an accuracy of 83.50%
27.	Ayu and Khotimah (2019)	Survey of customer satisfaction reviews on hotel aspects	Hotel review datasets	LSTM	GloVe	LSTM + GloVe achieves an accuracy of 94.6%
28.	Liu et al. (2021b)	Task classification	Product review data from the Amazon website	CDSAW model	Word2Vec (CBOW), CDSAW	CDSAW achieves an accuracy of 92.8%
29.	Sharma et al. (2021)	To identify indeterminacy and neutrality in the data	SemEval 2017 Task-4 dataset	BiLSTM, MPNet, stacked ensemble models	GloVe, BERT, ALBERT, RoBERT	A stacked ensemble of pre-trained language models (BERT, ALBERT, RoBERT), achieved an accuracy of 71.6%
30.	Shin et al. (2020)	Toxicity detection within online textual comments	Toxicity dataset	DT, RF, MLP, Dense, CNN, LSTM, BiLSTM	Word2Vec (Skip-Gram), BERT, DSWE	LSTM + DSWE achieve an accuracy of 95.7%
31.	Kilimci and Duvar (2020)	Twitter and financial news impact on the Turkish stock exchange market	Twitter, Bigpara, Public Disclosure platform dataset, Mynet Finans dataset	CNN, LSTM	Word2Vec, GloVe, fast-Text, BERT	LSTM + BERT achieves an accuracy of 84.32%
32.	Harb et al. (2020)	Twitter tweets emotion analysis	Tweets datasets from Github	CNN, Bi-LSTM	BERT	CNN with BERT achieves an accuracy of 76%
33.	Song et al. (2019)	Aspect-based sentiment analysis for the Korean language	Wikipedia Korean version, Customer reviews, Korean news articles dataset	LSTM	Word2Vec, Sentiment, and aspect lexicon embedding	The LSTM + SALE model achieves an accuracy of 92.91%
34.	El-Demerdash et al. (2022)	Personality detection using linguistic and psycholinguistic features	myPersonality dataset, Essays dataset	Fusion model	ELMo, ULMFET, BERT	Fusion model + BERT achieve an accuracy of 72.10%

Table 8 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
35.	Smetanin and Komarov (2021)	Russian language dataset sentiment analysis	SentRuEval-2015, 2016, RuTweetCorp, RuSentiment, LINIS Crowd, Kaggle Russian News Dataset, and RuReviews	Universal sentence encoder model	BERT, RuBERT	RuBERT achieves an F1-score of 77.44%

GANN gated alternate neural network, *GTR* gate truncation RNN, *BERT* bidirectional encoder representations from transformer, *OLiD* offensive language identification dataset, *ULMFiT* universal language model finetuning, *RGWE* refined global word embeddings, *SST* Stanford Sentiment Treebank, *SAWE* sentiment-aware word embeddings, *CDSAWA* cross-domain sentiment-aware word embeddings, *DANN* domain adversarial neural network, *HATN* hierarchical attention transfer network, *SALE* sentiment and aspect lexicon embedding.

Table 9 Review of biomedical text mining

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
1.	Wang et al. (2021b)	EHR analysis for visual prognosis	Electronic Health Records	Ophthalmology-specific word embedding approach	DSWE, Word2Vec, GloVe, BioWordVec	DSWE + BioWordVec achieves an accuracy of 99.5%
2.	Farris et al. (2021)	Disease diagnosis	Alfibi dataset of 263,867 questions	LR, RF, SGD, MLP	TF-IDF, Doc2Vec	MLP model achieves an accuracy of 84.9%
3.	Pandey et al. (2021)	Medical imaging and diagnosis	Cancer histology dataset	Deep learning model overview	Domain-specific word embedding	Overview of deep learning technologies
4.	Fan et al. (2020)	Adverse drug detection	WebMD and Drugs.com	Lexicon based approach	BERT	BERT + Sentence embedding achieves an accuracy of 94%
5.	Racharak (2021)	Description logic (DL) ontology	Medical diagnosis dataset	Data-driven method	Word2Vec, GloVe, fast-Text, BioWordVec	BioWordVec achieves higher performance
6.	Akhtyamova et al. (2020)	Biomedical entity extraction in Spanish clinical narratives	PharmacNER biomedical data	Entity extraction model	Flair, BERT	DSWE achieve an F1-score of 90.84%
7.	Koutsomitropoulos and Andriopoulos (2021)	Biomedical literature classification	PubMed repository	Deep- and shallow network approaches	Doc2Vec, ELMo, BERT	ELMo classifiers achieve an F1-score of 77%
8.	Ning and Bai (2021)	Biomedical NER	JNLPBA 2004 shared task data set	BiLSTM	GloVe	BiLSTM + GloVe model achieved F1-score of 75.62%
9.	Li et al. (2019b)	Biomedical word sense disambiguation	MSH Word sense disambiguation dataset, NLM WSD dataset	Bi-LSTM SVM	Word2Vec	BiLSTM achieve a macro average of 96.71%
10.	Guo et al. (2020)	Chromatin analysis	GM12878, K562, MCF-7, HeLa-S3, HI-hESC datasets	CNN, BiGRU	Word2Vec, GloVe	DeepANF is better achieves an accuracy of 92.14%
11.	Magna et al. (2020)	Classification of Cancer Diagnosis	EHR from a hospital in Chile, MIMIC III dataset	NB, SVM, Tree CART, RF, KNN, LSTM, BiLSTM	Word2Vec, BERT	BiLSTM + Word2Vec achieves an F1-score of 98.00%

Table 9 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
12.	Jiang et al. (2020b)	Classification of cancerous cells and normal cells	PubMed journals record	LR, SVM, CNN, LSTM, DECBAB-LSTM	ELMo, fastText, Word2Vec, GloVe, TF-IDF	DECBAB-LSTM achieves an accuracy of 86.3%
13.	Amin et al. (2020)	Diseases detection from Tweets on online social networking sites	Twitter corpus related to disease	LR, NB, SVM, ANN, DNN, LSTM	Word2Vec (Skip-Gram and CBOW)	LSTM + CBOW achieves an accuracy of 94%
14.	Yang et al. (2021b)	DNA sequencing	independent dataset	CNN	Word2Vec (Skip-Gram model)	iEnhancer-GAN achieves an accuracy of 95.10%
15.	Catelli et al. (2021)	EHR-NER	i2b2/UT Health 2014 de-identification corpus	BiLSTM, CRF	Flair embedding, GloVe, and BERT embedding	BiLSTM + CRF + BERT achieves an F1-score of 98.32%
16.	Khanal (2020)	Enhancer for identifying eukaryotic gene expression	Functional genomics datasets	CNN	Word2Vec (CBOW)	CNN + Word2Vec achieves an accuracy of 77.50%
17.	Kalyan and Sangeetha (2021)	Health information classification from tweets on social networking sites	CADEC-MCN dataset, TwADR-L datasets	LR, CNN, GRU	BERT, biomedical BERT, clinical BERT	The BERT model achieves an accuracy of 89.95%
18.	Yang et al. (2019)	Information extraction from electronic medical records using a contextual word embedding	I2B2/VA 2010 dataset	LSTM-CRF, BiLSTM	ELMo (Clinical)	BiLSTM + CRF + ELMo achieves an F1-score of 88.78%
19.	Fu and Yang (2019)	predicting type III secreted effectors	cross-species dataset	SVM, CNN	Word2Vec (Skip-Gram), WEDeepT3,	CNN + WEDeepT3 achieves an accuracy of 81.20%
20.	Moradi et al. (2020)	Summarization of bio-medical articles	PubMed Central open-access dataset	Graph based approach	Word2Vec (Skip-Gram and CBOW), GloVe, BioBERT, DSWE	GloVe model efficiently summarizes the text

Table 9 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
21.	Wu et al. (2019)	Therapeutic peptides for disease treatments	independent anticancer peptide dataset, virulent protein dataset, Hajjisharifi-Chen (HC) dataset	CNN	Word2Vec(Skip-Gram)	CNN + Word2Vec achieves an accuracy of 90.20%
22.	Goodrum et al. (2020)	EHR categorization into clinical and non-clinical categories	EHR record dataset	MNB, LR, RF	ClinicalBERT	ClinicalBERT achieves an accuracy of 97.3%
23.	Akkasi and Moens (2021)	EHR analysis for Cause and effect relationship identification	Hahn-Powell's dataset, BioCause dataset	SVM, LSTM, CNN	ELMo, BioBERT	CNN + Attention-based BiLSTM models achieves F1-score of 52%
24.	Blanco et al. (2020)	EHR classification based on diseases	Basque public health system dataset	BiGRU	fastText and ELMo	ELMo + BiGRU achieves an accuracy of 63.16%
25.	Pattisapu et al. (2019)	Text categorization based on Consultant and Patient persona	Gold and distant dataset	SVM, CNN, LSTM, HAN	ELMo, BERT	HAN achieves an F1 score of 29.30% for patient persona
26.	Yusuf et al. (2021)	Prediction of protein functions	G-protein coupled receptor hierarchical level dataset, cluster of orthologous groups (COG) database, phages of orthologous groups dataset	CNN	Word2vec	CNN + Word2Vec achieves 90.31% of MCC on the COG dataset

DSWE domain-specific word embeddings, *HAN* hierarchical attention networks, *WEDepT3* word embedding and deep learning for predicting T3SEs, *MCC* Mathew's correlation coefficients, *i2b2* informatics for integrating biology and the bedside, *n2c2* National NLP Clinical Challenges

Table 10 Review of named entity recognition and recommendation system

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
1.	Suárez-Paniagua et al. (2019)	Drug name extraction and recognition	DDI corpus, eHealth-KD corpus in English and Spanish language	BiLSTM, CNN	Word2Vec, GloVe, Sense2Vec	BiLSTM + CNN + CRF achieves an F1-score of 80.30%
2.	Liu et al. (2021a)	Extract useful information	CHEMDNER corpus	BiLSTM	BERT, GloVe, ELMo	BiLSTM + BERT achieves an accuracy of 90.84%
3.	Li et al. (2020b)	Chineses EHR classification	CCKS-CNER 2017 dataset	LSTM	BERT	LSTM + CRF + BERT achieves an accuracy of 91.60% better performance
4.	Wen et al. (2020)	Chinese language EHR classification	Chinese database from the internet	CNN, BiLSTM	Word2Vec, GloVe	CNN + BiLSTM achieves an F1-score of 74.39%
5.	Catelli et al. (2020)	Italian language EHR classification	English i2b2 2014 de-identification corpus, the Italian SIRM COVID-19 de-identification corpus	Bi-LSTM	BERT, MultiBPEmb, and Flair Multilingual Fast embeddings	MultiBPEmb + Flair multi-fast achieves a micro F1-score of 94.48%
6.	Chuang et al. (2021)	Automatic speech recognition and translation system	LibriSpeech corpus, Augmented LibriSpeech, Fisher Spanish corpora	LSTM, BiLSTM, CNN	Word2Vec, fastText	Word2Vec model efficiently maps speech signals to semantic space
7.	Zhang et al. (2021)	Chinese word representation	SogouCA data, Wikipedia dump, Fudan dataset	LSTM	Word2Vec, GloVe, BERT, CWE	LSTM + CWE achieves an F1-score of 95.53% for the NER task
8.	Shekhar et al. (2019)	English-Hindi mixed languages text identification	Dataset ICON 2016, Forum for IR Evaluation 2014 shared task on transliterated search, MSIR 2015, 2016	BiLSTM	Word2Vec, Character and word embedding	BiLSTM + Word2Vec achieves an F1-score of 83.9% for the NER task
9.	Khan et al. (2020)	NER in Urdu	IJCNLP-2008 dataset, UNER news dataset	LSTM, ANN	Text2Vec	LSTM + Text2Vec achieves highest F1-score of 81.10%

Table 10 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
10.	Wang et al. (2019)	Public opinion orientation analysis for text in the Chinese language	Business news corpus in Chinese language, Chinese Opinion Analysis Evaluation 2011 corpus	Document orientation analysis approach	Word2Vec (CBOW)	The proposed model achieves an accuracy of 87.23%
11.	Liu et al. (2019)	software bug localization by obtaining semantic similarity between bug reports and code file	Project dataset: Eclipse, SWT, AsepectJ, Zxing	Information retrieval approach	Word2Vec(Skip-Gram), GloVe	The GloVe+ POS tagging model achieves average precision of 30.7%
12.	Ezeani et al. (2019)	Embedding for the Welsh language	Welsh Wikipedia articles	NN approach	fastText	NN + fast-Text+ POS + SEM achieves an accuracy of 99.23% for multi-task taggers
13.	Budhkar et al. (2019)	Discrete data generalization	Chinese Poetry dataset, Coco Dataset	GAN	Word2Vec	GAN2vec achieves a BLUE score of 66.08%
14.	Lippincott et al. (2019)	Dialect identification	Arabic Dialect Corpus dataset, MADAR dataset	CNN, LSTM, Ensemble	Word2Vec, PPM	The ensemble approach achieves an F1-score of 63.4%
15.	Deng et al. (2019)	Named entity recognition	SemEval 2013–2016 dataset	LSTM	Domain-specific word embedding	The SSALSTM approach achieves an accuracy of 84.32%
16.	Morales-Garzón et al. (2021)	Word embedding to understand ingredients relations to adopt food recipes	Food.com dataset	Unsupervised approach	Word2Vec(CBOW), fast-Text, GloVe	Word embedding with fuzzy metrics achieves 95% confidence in selecting appropriate food recipes
17.	Yılmaz and Toklu (2020)	Question classification task on Turkish question dataset	Turkish question dataset	CNN, LSTM, SVM	Word2Vec (CBOW and Skip-Gram)	CNN + Skip-Gram achieves an accuracy of 94%

Table 10 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
18.	Khan et al. (2021)	Efficient recommendation system based on user preferences	Amazon Instant Videos, Apps for Android, Yelp dataset	CNN	Word2Vec	CNN + Word2Vec achieves an RMSE of 0.863
19.	Yang et al. (2021a)	Product review analysis based on word sentiment	Amazon Toy and Games, Kindle_Store dataset, Yelp-2017	BiLSTM	Word2Vec, GloVe	BiLSTM + Word2Vec achieves an accuracy of 85.13%
20.	Sun et al. (2020a)	Recommendation of product to the user based on previous user experience	Amazon product dataset, Yelp19 datasets	attention distribution guide information transfer network (ADGITTN)	GloVe	ADGITTN achieves the lowest mean square error of 1.031%
21.	Dau et al. (2021)	User product reviews analysis	Amazon, Yelp datasets	Adaptive Deep learning-based method for Recommendation System (ADRS)	Word2Vec(CBOW)	ADRS achieves a mean absolute error of 9.04%
22.	Jeon and Kim (2021)	Source code analysis for software vulnerability	National vulnerability dataset, Software assurance reference dataset	LSTM, BiLSTM, GRU, BiGRU	Word2Vec, Doc2Vec, GloVe, fastText	BiGRU achieves F1-score of 96.11%,

DSWE domain-specific word embeddings, *CWE* Chinese words embedding, *POS* part of speech tagging

Table 11 Review of topic modelling

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
1.	Xiao et al. (2018)	Automatic bug localization to corresponding bug file	AspectJ, Eclipse, JDT, SWT, Tomcat projects	CNN, DeepLoc	Word2Vec	DeepLoc + Word2Vec achieves an accuracy of 81.00%
2.	Qiu et al. (2019)	Extraction algorithm using enhanced word embedding for keyphrase extraction	Chinese academic database	Naive Bayesian model	Word2Vec, GloVe, Doc2Vec, OWE	OWE achieves an F1-score of 40.7%
3.	Pan et al. (2019b)	Key phrase extraction to recommend task allocation	Tianpeng web dataset, text8 corpora	Semantic tag similar matrix	Word2Vec	Task recommendation performance is improved with Word2Vec
4.	Dridi et al. (2019)	Leap2Trend approach to rank keywords for recent trend analysis	Google Trends hits data, Google Scholar citations data	Similarity based approach	Word2Vec(Skip-Gram)	Leap2Trend approach achieves an accuracy of 80%
5.	Zhu et al. (2020a)	Multimodal word representation model to understand syntactic and phonetic information	ESLI dataset, Word-Sim-353, WS-240, 296, SemEval-2012, IMDB, Yelp reviews datasets	Multimodal model	Word2Vec, GloVe	The multimodal word representation model achieves an accuracy of 78.23%
6.	Xiao et al. (2019)	Recommendations based on user preferences and sense-based word embedding approach on short Chinese text messages	Social network dataset in the Chinese language, English dataset from Wikipedia, Hownet database	Time-aware probabilistic model	Sense-based word embedding	The proposed model is efficiently recommended based on the combined outcome of sense-based embedding, feature selection using topic modeling
7.	Flisar and Podgorelec (2019)	Software flawed identification	Source code comments extracted from open source java projects from GitHub	NB, SVM	Word2Vec, DSWE	DSWE achieves an accuracy of 82%
8.	Hu et al. (2019)	Topic evolution analysis based on the semantic keyword mapping	SCIE and SSCI datasets from 1985 to 2016	Spatial autocorrelation analysis	Word2Vec(Skip-Gram)	Word2Vec is capable of representing topics efficiently

Table 11 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
9.	El-Assady et al. (2020)	Understanding the semantics of words and readjusting the relations based on requirements	US Presidential debate corpus	Targeted refinement	Word2Vec	Topic modeling performance is improved using Word2Vec
10.	Li et al. (2018)	Word2API embedding to map the relation between words and corresponding API	source code files from GitHub, Java-tagged questions in Stack Overflow	Acquisition and alignment approach	Word2API	The Word2API approach achieves average mean precision of 43.6%
11.	Qiu et al. (2020a)	Domain concept extraction	Chinese Football Association Super League (CSL) competition dataset	Semantic-based method	Word2Vec	Semantic graph-based concept extraction is effective
12.	Alqaisi and O'Keefe (2019)	Machine translation of Arabic to English (Ar-En) language	Arabic to English (Ar-En) dataset, Web Inventory of Transcribed and Translated Talks (WIT3) dataset	Semantic-based method	Word2Vec	Bil-BOWA model efficiently learns better Ar-En BWEs
13.	Shi et al. (2019)	word-level and sentence-level translation tasks	Monolingual lexical definitions in English, Spanish language	Semantic-based method	Word2Vec(Skip-Gram)	BilLex effectively portrays the task of multilingual translation
14.	Talafha et al. (2019)	Arabic Tweet Dialect Identification	Arabic Twitter Dataset	SVM, LSTM	Word2Vec(CBOW, Skip-Gram), fastText	Linear SVC outperforms with an F1-score of 71.84%
15.	Nassif et al. (2021)	People's emotions and opinions analysis	Arabic language dataset	RNN and CNN	Word2Vec	Review paper on deep learning for sentiment analysis: for the Arabic language

OEWE Ontology and enhanced word embedding-based

Table 12 Review of the importance of word embedding

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
1.	Dharmaretnam et al. (2021)	CNN representation	ImageNet, CIFAR-100	VGG16, Inception-v3, ResNet50, CNN	Word2Vec	Efficiently represented the word
2.	Risch et al. (2019)	Downstream tasks in the patent domain	WIPO- α data set, USPTO, USPTO-2 M,	BiGRU	Word2Vec	MAP—64%
3.	Ji et al. (2019)	Parallelizing the algorithm	Small text8 dataset from Wikipedia, One Billion Words benchmark dataset	Mini batching and negative sample sharing	fastText	Word2Vec helps to scale multi-core architectures
4.	Chuan et al. (2020)	Semantic relationships in music with Word2Vec	MIDI dataset	Slices based on spatial proximity	Word2Vec	Word2Vec captures meaningful tonal and harmonic relationships
5.	Yildirim (2019)	Turkish corpus relation extraction	Turkish dataset	Optimized projection algorithm	Word2Vec	Accuracy—90.75%
6.	Wang and Kuo (2020)	Multiple layers of the BERT model	STS-dataset 2012–2016	SBERT-WK model	Word2Vec, fastText	SBERT-WK efficiently represents sentences
7.	Yildiz and Tezgider (2021)	Hyperparameters	Turkish texts dataset	Grid search and the random search	BERT	F1-score—89.7%, Accuracy—93.8%
8.	Chalkidis and Kampas (2019)	Legal analytics	Board of Veterans Appeals (BVA) cases and World Intellectual Property Organization	LSTM, BiLSTM CNN	Word2Vec, Law2Vec	BiLSTM+CRF+Law2Vec achieves F1-score of 88%
9.	Rezaeinia et al. (2019)	Sentiment analysis	Movie reviews, Stanford sentiment treebank, TREC, CR from Amazon, RT movie review,	CNN	Law2Vec, Word2Vec, GloVe, fastText	CNN + IWV achieves an accuracy of 87%
10.	Kim and Jeong (2021)	Mirroring vector space approach	American online dictionary Disaster Tweets dataset	Mirror Vector Space model	Word2Vec, GloVe, Pos2Vec, Lexicon2Vec, Word-position2Vec	The proposed model achieves an accuracy of 83.14%

Table 12 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
11.	Cai et al. (2019)	Prediction of next alarm in process plants	Data from Central Heating and Cooling Plant at the University of California, Davis campus	LSTM	Word2Vec, GloVe, BERT	LSTM+ Word2Vec achieves an accuracy of 81.40%
12.	Li et al. (2019a)	Scale Word2Vec on a GPU cluster	Word-Sim353 (WS), MEN dataset, and Text8 corpus	High-level Chainer deep learning framework	Word2Vec	The proposed framework with Word2Vec achieves higher results at the subword level
13.	Chen et al. (2019)	Phonetic information extraction	Sequence-to-sequence autoencoder (SSA)	RNN	Word2Vec	Achieved higher accuracy using SSA with Word2Vec
14.	Li et al. (2021)	Topic modeling, text classification, word similarity	Wikipedia and IMDB	Probabilistic approach	AudioWord2Vec	Precision-82.8%, F1-score76.9%
15.	Liao et al. (2020)	Word Similarity	IMDB dataset	Isotropic Iterative Quantization (IIQ), CNN	GloVe	The proposed IIQ approach achieves 76.43% accuracy
16.	Kalouli et al. (2019)	Synonymy detection	WordNet dataset	LSTM	GloVe	Semantic similarity between vectors of nouns improves synonymy detection
17.	Raunak et al. (2019)	Dimensionality reduction	Sentence classification datasets	Post-processing algorithm	Word2Vec (Skip-Gram model), GloVe, fast-Text	The proposed approach achieves a Spearman rank correlation coefficient of 91.6
18.	Shin et al. (2019)	Dimensionality reduction	TREC dataset, Amazon Review dataset, SST dataset	CNN, LSTM	Word2Vec	The distillation ensemble strategy achieves an accuracy of 93.48% on TREC dataset

Table 12 (continued)

Sr. No	References	Application area	Name of dataset	Model architecture	Embedding method	Performance
19.	Rethmeier and Plank (2019)	Task-specific word embedding	WikiText dataset	Post-processing algorithm	GloVe, fastText	GloVe + Post-Processing approach yields better embedding with small corpora
20.	Tian et al. (2019)	Extract API information	AWSDL-TC3 dataset, corpus from Wikipedia	Gaussian Latent Dirichlet Allocation (GLDA)	GloVe, Word2Vec, ELMo, BERT, Adaptive Cross-contextual Word Embedding model	Gaussian LDA model performs efficiently for web service discovery
21.	Wang et al. (2020), Verma and Khandelwal (2019)	Deep learning environment	Discuss the steps followed in each embedding	RNN, CNN	Word2Vec, GloVe	Discuss the effectiveness of deep learning models
22.	Ihm et al. (2019)	Word extraction for Korean language	Assault Incident News Articles and Knowledge Encyclopedia Dictionary	Backward mapping and skipping method	Word2Vec, GloVe, fastText, ELMo, OpenAI-GPT, BERT	The proposed approach achieves effective performance

Table 13 List of publishers/journals

Abbreviation	Name of publishers/journals
ACM Tran	ACM Transactions on Audio, Speech, and Language Processing
AIKDE	AI, Knowledge and Data Engineering
AIL	Artificial Intelligence and Law
AIM	Artificial Intelligence in Medicine
ASCJ	Applied Soft Computing Journal
CCPE	Concurrency Computat Pract Exper
CEE	Computers and Electrical Engineering
CMPB	Computer Methods and Programs in Biomedicine
CS	Computer and Security
DEIDH	Data-Enabled Intelligence for Digital Health
DSE	Data Science and Engineering
DSS	Decision Support Systems
DTA	Data Technologies and Applications
EAAI	Engineering Applications of Artificial Intelligence
EIJ	Egyptian Informatics Journal
ESA	Expert Systems with Applications
ETRI Journal	<i>Electronics and Telecommunications Research Institute</i>
FGCS	Future Generation Computer Systems
HWCMC	Hindawi Wireless Communications and Mobile Computing
IEEE	Institute of Electrical and Electronics Engineers
IEEE Tran. PDS	IEEE Transactions on Parallel and Distributed Systems
IJCDS	International Journal of Computing and Digital Systems
IJCIA	International Journal of Computational Intelligence and Applications
IJIES	International Journal of Intelligent Engineering and Systems
IJISAE	International Journal of Intelligent Systems and Applications in Engineering
IJITEE	International Journal of Innovative Technology and Exploring Engineering
IJIV	International Journal on Informatics Visualization
IJKIES	International Journal of Knowledge-based and Intelligent Engineering Systems
IJMI	International Journal of Medical Informatics
IJMLC	International Journal of ML and Cybernetics
IJMS	International Journal of Molecular Sciences
IM	Information & Management
IMU	Informatics in Medicine Unlocked
INASS	The Intelligent Networks And System Society
IPM	Information Processing and Management
IS	Information Systems
ISAT	ISA Transactions
IST	Information and Software Technology
JBA	Journal of Business Analytics
JB1	Journal of Biomedical Informatics
JCMSE	Journal of Computational Methods in Sciences and Engineering
JDS	Journal of Decision Systems
JKSU-CIS	Journal of King Saud University—Computer and Information Sciences
JSS	The Journal of Systems and Software
JWS	John Wiley & Sons, Ltd
KBS	Knowledge-Based Systems
KDSD	Knowledge Discovery for Software Development
MIS	Mobile Information Systems

Table 13 (continued)

Abbreviation	Name of publishers/journals
MM	Microprocessors and Microsystems
MTA	Multimedia Tools and Applications
NCA	Neural Computing and Applications
OSNM	Online Social Networks and Media
PRL	Pattern Recognition Letters
SNAM	Social Network Analysis and Mining
TJEECS	Turkish Journal of Electrical Engineering & Computer Sciences
TKDE	Transactions on Knowledge and Data Engineering
TSE	Transactions on Software Engineering
TVCG	Transactions on Visualization and Computer Graphics
AAAI	The Thirty-Third AAAI Conference on Artificial Intelligence
ACL	Association for Computational Linguistics
ACL & IJCNLP	59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing
ANLPW & ACL	Proceedings of the Fourth Arabic Natural Language Processing Workshop—Association for Computational Linguistics
CAIDA	1st International Conference on Artificial Intelligence and Data Analytics
CMCL	Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics
COLING	Proceedings of the 28th International Conference on Computational Linguistics
EMNLP & IJCNLP	(EMNLP & IJCNLP—ACL) Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing—Association for Computational Linguistics
EuroSSP	IEEE European Symposium on Security and Privacy
ICIDT	International Conference on Intelligent Decision Technologies
ICLR	International Conference on Learning Representations
ICNLSP—ACL	Proceedings of the 4th International Conference on Natural Language and Speech Processing, Association for Computational Linguistics
IF	Elsevier—Information Fusion
IJCAI	Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence
IJCAI-DCT	Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Doctoral Consortium Track
IOT	IEEE Internet of Things Journal
IW3C2	International World Wide Web Conference Committee
Repl4NLP—ACL	Proceedings of the 4th Workshop on Representation Learning for NLP, Association for Computational Linguistics
SNS	Springer Nature Singapore
TASLP	IEEE/ACM Transactions on Audio, Speech, and Language Processing
TCSS	IEEE Transactions on Computational Social Systems
TII	IEEE Transactions on Industrial Informatics
TKDE	IEEE Transactions on Knowledge and Data Engineering
TNSE	IEEE Transactions on Network Science and Engineering

SANAD Single-label Arabic news articles datasets; *NADiA* News articles datasets in Arabic with multi-labels; *HAN* Hierarchical attention network; *HDBSCAN* Hierarchical Density-Based Spatial Clustering of Applications with Noise; *LDA* Logistic regression, linear discriminant analysis; *QDA* Quadratic discriminant analysis; *NB* Naïve Bayes; *SVM* Support vector machine; *KNN* k-nearest neighbor; *DT* Decision tree, *RF* Random forest; *XGBoost MLP* Multilayer perceptron; *LIWC* Linguistic Inquiry and Word Count features; *NER* Named entity recognition; *PMMC* Process model matching contest dataset; *DLMF* Digital Library of Mathematical Functions; *GB* Gradient Boosting; *SGC* Stochastic Gradient Descent; *HAN* Hierarchical attention network; *DFFNN* Deep feed-forward neural network

9 Annexure B

See Table 14.

Table 14 The description, benefits, and drawbacks of various word representation models

	Embedding approach	Description	Benefits	Drawbacks
Conventional model	BOW	Based on the frequency of words assigns weight to each word	Easily represents the word in the corresponding vector form	Sparsity and ignoring word orders Frequent words have more power Inefficient to handle the out-of-vocabulary word
	n-gram	Divides the sentences into n tokens (word level and character level)	Importance to the sequence of the word It can be used as a spell checker Capable of predicting the next word	Sparsity problem Inefficient to handle the out-of-vocabulary word
	TF-IDF	Using document frequency and inverse document frequency calculates a more accurate vector representation of a word	Keep relevant word scores, and reduce the score for frequent words Easy to get document similarity	Only consider the terms Unable to capture the semantic relationship between words Unable to capture document topic
Distributed model	VSM	Multidimensional vectors are used to represent the corpus documents. Each term has a weight assigned to it that signifies its importance in each document	Especially good at estimating document similarity and, as a result, document clustering	The terms positioning in the text, word order, or co-occurrence across the corpus are not considered Large, sparse vectors are complicated to work within large corpora with an extensive vocabulary
	Word2Vec	By employing dense representation, the Word2Vec technique can construct word embedding. It is a prediction model that assigns probabilities to terms that perform well in word similarity tests	It captures the syntactic and semantic information about the text Word2Vec is a predictive model	It is unable to extract out-of-vocabulary words from a corpus It is unable to extract the polysemy word from the text
	GloVe	GloVe is an unsupervised learning technique that generates word vector representations. GloVe is a count-based model that combines the local context window approach and global matrix factorization	In the vector space, it captures sub-linear interactions It captures the word's syntactic and semantic meanings A large corpus of data is used in training	It is unable to extract out-of-vocabulary words from a corpus It is unable to extract the polysemy word from the text The construction of the global word to word co-occurrence matrix is a computationally intensive task for large corpora and needs more storage memory

Table 14 (continued)

	Embedding approach	Description	Benefits	Drawbacks
Contextual model	fastText	fastText is a Word2Vec extension that recognizes words as character n-grams. It generates an efficient and effective vector representation of infrequent words	Handle out-of-vocabulary terms effectively with character n-grams	Fails to capture the essence of the word polysemy Compared to GloVe and Word2Vec, it is more computationally intensive
	ELMo	The ELMo embedding is character-based and context-dependent. Depending on the context, a word might have different meanings	It uses the BiLSTM model, a feature-based method that includes a feature of pre-trained representation Able to adequately represent the meaning of the polysemy word	Fails to catch both the left and right contexts of words at the same time
	GPT	GPT extracts feature using a transformer, a one-way language model. The language model uses a multi-layer transformer decoder with a self-attention mechanism to anticipate the current word through the first N words	Employs task-specific parameters that have been trained on downstream tasks and applies a fine-tuning approach Uses a one-way language model	GPT-2 necessitates a lot of processing, and there's a danger it will provide erroneous data because it has been trained on multiple sources
	BERT	BERT creates dense vector representations for natural language by combining a deep, pre-trained neural network with the transformer architecture. The BERT model is fine-tuned by first initializing it with pre-trained parameters, then fine-tuning all the parameters with labeled data from downstream operations	Parallelization is achieved via the transformer architecture Effectively captures the context of words by simultaneously evaluating the left and right sides of words Includes a multi-layer bidirectional transformer encoder Employs masked language modeling to optimize and combine position embedding with static word embeddings	BERT training and fine-tuning processes necessitate a lot of computing power, making it expensive

References

- Agüero-Torales MM, Abreu Salas JI, López-Herrera AG (2021) Deep learning and multilingual sentiment analysis on social media data: An overview. *Appl Soft Comput* 107:107373. <https://doi.org/10.1016/j.asoc.2021.107373>
- Akhtyamova L, Martínez P, Verspoor K, Cardiff J (2020) Testing contextualized word embeddings to improve NER in Spanish clinical case narratives. *IEEE Access* 8:164717–164726. <https://doi.org/10.1109/ACCESS.2020.3018688>
- Akkasi A, Moens MF (2021) Causal relationship extraction from biomedical text using deep neural models: a comprehensive survey. *J Biomed Inform* 119:103820. <https://doi.org/10.1016/j.jbi.2021.103820>
- Al-Ramahi M, Alsmadi I (2021) Classifying insincere questions on Question Answering (QA) websites: meta-textual features and word embedding. *J Bus Anal* 4:55–66. <https://doi.org/10.1080/2573234X.2021.1895681>
- Alamoudi ES, Alghamdi NS (2021) Sentiment classification and aspect-based sentiment analysis on yelp reviews using deep learning and word embeddings. *J Decis Syst* 30:259–281. <https://doi.org/10.1080/12460125.2020.1864106>
- Alatawi HS, Alhothali AM, Moria KM (2021) Detecting white supremacist hate speech using domain specific word embedding with deep learning and BERT. *IEEE Access* 9:106363–106374. <https://doi.org/10.1109/ACCESS.2021.3100435>
- Radford A, Narasimhan K, Salimans T, Sutskever I (2018) Improving language understanding by generative pre-training
- Alharthi R, Alhothali A, Moria K (2021) A real-time deep-learning approach for filtering Arabic low-quality content and accounts on Twitter. *Inf Syst* 99:101740. <https://doi.org/10.1016/j.is.2021.101740>
- Almuhareb A, Alsanie W, Al-thubaity A (2019) Arabic word segmentation with long short-term memory neural networks and word embedding. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2019.2893460>
- Almuzaini HA, Azmi AM (2020) Impact of stemming and word embedding on deep learning-based Arabic text categorization. *IEEE Access* 8:127913–127928. <https://doi.org/10.1109/ACCESS.2020.3009217>
- Alqaisi T, O’Keefe S (2019) En-Ar bilingual word embeddings withoutword alignment: Factors Effects. In: *Proc Fourth Arab Nat Lang Process Work - Assoc Comput Linguist ANLPW-ACL-2019*, pp 97–107. <https://doi.org/10.18653/v1/w19-4611>
- Alrajhi K, ELAffendi MA (2019) Automatic Arabic part-of-speech tagging: deep learning neural LSTM versus Word2Vec. *Int J Comput Digit Syst* 8:308–315. <https://doi.org/10.12785/ijcids/080310>
- Alwehaibi A, Bikdash M, Albogmi M, Roy K (2021) A study of the performance of embedding methods for Arabic short-text sentiment analysis using deep learning approaches. *J King Saud Univ*. <https://doi.org/10.1016/j.jksuci.2021.07.011>
- Amin S, Irfan Uddin M, Ali Zeb M et al (2020) Detecting dengue/flu infections based on tweets using LSTM and word embedding. *IEEE Access* 8:189054–189068. <https://doi.org/10.1109/ACCESS.2020.3031174>
- Atzeni M, Reforgiato Recupero D (2020) Multi-domain sentiment analysis with mimicked and polarized word embeddings for human–robot interaction. *Futur Gener Comput Syst* 110:984–999. <https://doi.org/10.1016/j.future.2019.10.012>
- Ayu D, Khotimah K (2019) Sentiment analysis of hotel aspect using probabilistic latent semantic analysis word embedding and LSTM. *Int J Intell Eng Syst*. <https://doi.org/10.22266/ijies2019.0831.26>
- Beddiar DR, Jahan MS, Oussalah M (2021) Data expansion using back translation and paraphrasing for hate speech detection. *Online Soc Networks Media* 24:153. <https://doi.org/10.1016/j.osnem.2021.100153>
- Bengio Y, Ducharme R, Vincent P et al (2003) A neural probabilistic language model. *J Mach Learn Res* 3:1137–1155. <https://doi.org/10.1162/153244303322533223>
- Bernardy JP, Lappin S (2022) A neural model for compositional word embeddings and sentence processing. In: *Proc Work Cogn Model Comput Linguist C*, pp 12–22. <https://doi.org/10.18653/v1/2022.cmcl-1.2>
- Birjali M, Kasri M, Beni-Hssane A (2021) A comprehensive survey on sentiment analysis: approaches, challenges and trends. *Knowl-Based Syst* 226:107134. <https://doi.org/10.1016/j.knosys.2021.107134>
- Blanco A, Perez-de-Viñaspre O, Pérez A, Casillas A (2020) Boosting ICD multi-label classification of health records with contextual embeddings and label-granularity. *Comput Methods Programs Biomed*. <https://doi.org/10.1016/j.cmpb.2019.105264>
- Brown TB, Mann B, Ryder N et al (2020) Language models are few-shot learners. *Adv Neural Inf Process Syst*. <https://doi.org/10.48550/arXiv.2005.14165>

- Budhkar A, Vishnubhotla K, Hossain S, Rudzicz F (2019) Generative adversarial networks for text using word2vec intermediaries. In: Proc 4th Work Represent Learn NLP, Assoc Comput Linguist Repl4NLP-ACL-2019, pp 15–26. <https://doi.org/10.18653/v1/W19-4303>
- Cai S, Palazoglu A, Zhang L, Hu J (2019) Process alarm prediction using deep learning and word embedding methods. ISA Trans 85:274–283. <https://doi.org/10.1016/j.isatra.2018.10.032>
- Campbell JC, Hindle A, Stroulia E (2015) Latent dirichlet allocation: extracting topics from software engineering data. Art Sci Anal Softw Data 3:139–159. <https://doi.org/10.1016/B978-0-12-411519-4.00006-9>
- Catelli R, Casola V, De Pietro G et al (2021) Combining contextualized word representation and sub-document level analysis through Bi-LSTM+CRF architecture for clinical de-identification. Knowl Based Syst 213:106649. <https://doi.org/10.1016/j.knosys.2020.106649>
- Catelli R, Gargiulo F, Casola V et al (2020) Crosslingual named entity recognition for clinical de-identification applied to a COVID-19 Italian data set. Appl Soft Comput J 97:106779. <https://doi.org/10.1016/j.asoc.2020.106779>
- Chai Y, Du L, Qiu J et al (2022) Dynamic prototype network based on sample adaptation for few-shot malware detection. IEEE Trans Knowl Data Eng. <https://doi.org/10.1109/TKDE.2022.3142820>
- Chalkidis I, Kampas D (2019) Deep learning in law: early adaptation and legal word embeddings trained on large corpora. Artif Intell Law 27:171–198. <https://doi.org/10.1007/s10506-018-9238-9>
- Chen YC, Huang SF, Lee HY et al (2019) Audio Word2vec: sequence-to-sequence autoencoding for unsupervised learning of audio segmentation and representation. IEEE/ACM Trans Audio Speech Lang Process 27:1481–1493. <https://doi.org/10.1109/TASLP.2019.2922832>
- Cheng L, Kim N, Liu H (2022) Debiasing word embeddings with nonlinear geometry. In: Proc 29th Int Conf Comput Linguist COLING, pp 1286–1298. <https://doi.org/10.48550/arXiv.2208.13899>
- Choudhary M, Chouhan SS, Pilli ES, Vipparthi SK (2021) BerConvoNet: a deep learning framework for fake news classification. Appl Soft Comput 110:10614. <https://doi.org/10.1016/j.asoc.2021.107614>
- Chuan CH, Agres K, Herremans D (2020) From context to concept: exploring semantic relationships in music with word2vec. Neural Comput Appl 32:1023–1036. <https://doi.org/10.1007/s00521-018-3923-1>
- Chuang SP, Liu AH, Sung TW, Lee HY (2021) Improving automatic speech recognition and speech translation via word embedding prediction. IEEE/ACM Trans Audio Speech Lang Process 29:93–105. <https://doi.org/10.1109/TASLP.2020.3037543>
- Craja P, Kim A, Lessmann S (2020) Deep learning for detecting financial statement fraud. Decis Support Syst. <https://doi.org/10.1016/j.dss.2020.113421>
- Dau A, Salim N, Idris R (2021) An adaptive deep learning method for item recommendation system. Knowl Based Syst 213:106681. <https://doi.org/10.1016/j.knosys.2020.106681>
- Dadkhah S, Shoehleh F, Yadollahi MM et al (2021) A real-time hostile activities analyses and detection system. Appl Soft Comput 104:107175. <https://doi.org/10.1016/j.asoc.2021.107175>
- de Mendonça LRC, da Cruz Júnior G (2020) Deep neural annealing model for the semantic representation of documents. Eng Appl Artif Intell 96:103982. <https://doi.org/10.1016/j.engappai.2020.103982>
- Deng D, Jing L, Yu J, Sun S (2019) Sparse self-attention LSTM for sentiment lexicon construction. IEEE/ACM Trans Audio Speech Lang Process 27:1777–1790. <https://doi.org/10.1109/TASLP.2019.2933326>
- Dessi D, Recupero DR, Sack H (2021) An assessment of deep learning models and word embeddings for toxicity detection within online textual comments. Electron. <https://doi.org/10.3390/electronics10070779>
- Devlin J, Chang MW, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL HLT Conf North Am Chapter Assoc Comput Linguist Hum Lang Technol, vol 1, pp 4171–4186. <https://doi.org/10.18653/v1/N19-1423>
- Dhar A, Mukherjee H, Sekhar N, Kaushik D (2020) Text categorization: past and present. Springer, Amsterdam
- Dharmaretnam D, Foster C, Fyshe A (2021) Words as a window: using word embeddings to explore the learned representations of convolutional neural networks. Neural Netw 137:63–74. <https://doi.org/10.1016/j.neunet.2020.12.009>
- Döbrössy B, Makrai M, Tarján B, Szaszák G (2019) Investigating sub-word embedding strategies for the morphologically rich and free phrase-order Hungarian. In: Proc 4th Work Represent Learn NLP, Assoc Comput Linguist Repl4NLP-ACL-2019, pp 187–193. <https://doi.org/10.18653/v1/w19-4321>
- Dogru HB, Tilki S, Jamil A, Ali Hameed A (2021) Deep learning-based classification of news texts using Doc2Vec model. In: 1st Int Conf Artif Intell Data Anal CAIDA-2021, pp 91–96. <https://doi.org/10.1109/CAIDA51941.2021.9425290>

- Dridi A, Gaber MM, Muhammad Atif Azad R, Bhogal J (2019) Leap2Trend: a temporal word embedding approach for instant detection of emerging scientific trends. *IEEE Access* 7:176414–176428. <https://doi.org/10.1109/ACCESS.2019.2957440>
- Du C, Sun H, Wang J, et al (2019) Investigating capsule network and semantic feature on hyperplanes for text classification. In: *Proc 2019—Conf Empir Methods Nat Lang Process 9th Int Jt Conf Nat Lang Process (EMNLP-IJCNLP-ACL)*, Assoc Comput Linguist, pp 456–465. <https://doi.org/10.18653/v1/d19-1043>
- Ebadulla D, Raman R, Shetty HK, Mamatha HR (2021) A comparative study on language models for the Kannada language. In: *Proc 4th Int Conf Nat Lang Speech Process Assoc Comput Linguist ICNLSP-ACL-2021*, pp 280–284
- Ekaterina Vylomova NH (2021) Semantic changes in harm-related concepts in English. *Language Science Press, Berlin*
- El-Alami F, zahra, Ouatik El Alaoui S, En Nahnahi N, (2021) Contextual semantic embeddings based on fine-tuned AraBERT model for Arabic text multi-class categorization. *J King Saud Univ*. <https://doi.org/10.1016/j.jksuci.2021.02.005>
- El-Assady M, Kehlbeck R, Collins C et al (2020) Semantic concept spaces: guided topic model refinement using word-embedding projections. *IEEE Trans Vis Comput Graph* 26:1001–1011. <https://doi.org/10.1109/TVCG.2019.2934654>
- El-Demerdash K, El-Khoribi RA, Ismail Shoman MA, Abdou S (2022) Deep learning based fusion strategies for personality prediction. *Egypt Inform J* 23:47–53. <https://doi.org/10.1016/j.eij.2021.05.004>
- Elnagar A, Al-Debsi R, Einea O (2020) Arabic text classification using deep learning models. *Inf Process Manag* 57:102121. <https://doi.org/10.1016/j.ipm.2019.102121>
- Elsafoury F, Wilson SR, Katsigiannis S, Ramzan N (2022) SOS: systematic offensive stereotyping bias in word embeddings. In: *Proc 29th Int Conf Comput Linguist COLING 1263–1274*
- Erk K (2012) Vector space models of word meaning and phrase meaning: a survey. *Linguist Lang Compass* 6:635–653. <https://doi.org/10.1002/Inco.362>
- Ezeani I, Piao S, Neale S, et al (2019) Leveraging pre-trained embeddings for Welsh taggers. In: *Proc 4th Work Represent Learn NLP, Assoc Comput Linguist RepL4NLP-ACL-2019*, pp 270–280. <https://doi.org/10.18653/v1/W19-4332>
- Fan B, Fan W, Smith C, Garner H, “Skip”, (2020) Adverse drug event detection and extraction from open data: a deep learning approach. *Inf Process Manag* 57:102131. <https://doi.org/10.1016/j.ipm.2019.102131>
- Faris H, Habib M, Faris M et al (2021) An intelligent multimodal medical diagnosis system based on patients’ medical questions and structured symptoms for telemedicine. *Inform Med Unlocked* 23:100513. <https://doi.org/10.1016/j.imu.2021.100513>
- Fesseha A, Xiong S, Emiru ED et al (2021) Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya. *Informatics* 12:1–17. <https://doi.org/10.3390/info12020052>
- Firth JR (1957) *Studies in linguistic analysis*. Blackwell, Oxford
- Flisar J, Podgorelec V (2019) Identification of self-admitted technical debt using enhanced feature selection based on word embedding. *IEEE Access* 7:106475–106494. <https://doi.org/10.1109/ACCESS.2019.2933318>
- Flor M, Hao J (2021) Text mining and automated scoring. *Comput Psychom New Methodol New Gener Digit Learn Assess*. https://doi.org/10.1007/978-3-030-74394-9_14
- Fouad MM, Mahany A, Aljohani N et al (2020) ArWordVec: efficient word embedding models for Arabic tweets. *Soft Comput* 24:8061–8068. <https://doi.org/10.1007/s00500-019-04153-6>
- Fu X, Yang Y (2019) WEDeepT3: predicting type III secreted effectors based on word embedding and deep learning. *Quant Biol* 7:293–301. <https://doi.org/10.1007/s40484-019-0184-7>
- Giarelis N, Kanakaris N, Karacapilidis N (2020) On a novel representation of multiple textual documents in a single graph. *Smart Innov Syst Technol* 193:105–115. https://doi.org/10.1007/978-981-15-5925-9_9/TABLES/1
- Giesen J, Kahlmeyer P, Nussbaum F, Zarriß S (2022) Leveraging the Wikipedia Graph for Evaluating Word Embeddings. *Proc Thirty-First Int Jt Conf Artif Intell IJCAI-22* 4136–4142. <https://doi.org/10.24963/ijcai.2022/574>
- Giorgi J, Nitski O, Wang B, Bader G (2021) DeCLUTR: deep contrastive learning for unsupervised textual representations. In: *Proc 59th Annu Meet Assoc Comput Linguist 11th Int Jt Conf Nat Lang Process ACL-IJCNLP*, pp 879–895. <https://doi.org/10.18653/v1/2021.acl-long.72>
- González JÁ, Hurtado LF, Pla F (2020) Transformer based contextualization of pre-trained word embeddings for irony detection in Twitter. *Inf Process Manag* 57:102262. <https://doi.org/10.1016/j.ipm.2020.102262>

- Goodrum H, Roberts K, Bernstam EV (2020) Automatic classification of scanned electronic health record documents. *Int J Med Inform* 144:104302. <https://doi.org/10.1016/j.ijmedinf.2020.104302>
- Greiner-Petter A, Youssef A, Ruas T et al (2020) Math-word embedding in math search and semantic extraction. *Scientometrics* 125:3017–3046. <https://doi.org/10.1007/s11192-020-03502-9>
- Grishman R, Sundheim BM (1996) Message Understanding Conference—6: A Brief History. In: *The 16th International Conference on Computational Linguistics. COLING 1996*, pp 466–471
- Grzeça M, Becker K, Galante R (2020) Drink2Vec: Improving the classification of alcohol-related tweets using distributional semantics and external contextual enrichment. *Inf Process Manag* 57:102369. <https://doi.org/10.1016/j.ipm.2020.102369>
- Guo Y, Zhou D, Nie R et al (2020) DeepANF: a deep attentive neural framework with distributed representation for chromatin accessibility prediction. *Neurocomputing* 379:305–318. <https://doi.org/10.1016/j.neucom.2019.10.091>
- Ha P, Zhang S, Djuric N, Vucetic S (2020) Improving word embeddings through iterative refinement of word- and character-level models. In: *Proc 28th Int Conf Comput Linguist COLING*, pp 1204–1213. <https://doi.org/10.18653/v1/2020.coling-main.104>
- Hajek P, Barushka A, Munk M (2020) Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. *Neural Comput Appl* 32:17259–17274. <https://doi.org/10.1007/s00521-020-04757-2>
- Hammar K, Jaradat S, Dokoohaki N, Matskin M (2020) Deep text classification of Instagram data using word embeddings and weak supervision. In: *Web Intelligence*, vol 18, pp 53–67. <https://doi.org/10.3233/WEB-200428>
- Hao Y, Mu T, Hong R et al (2020) Cross-domain sentiment encoding through stochastic word embedding. *IEEE Trans Knowl Data Eng* 32:1909–1922. <https://doi.org/10.1109/TKDE.2019.2913379>
- Harb JGD, Ebeling R, Becker K (2020) A framework to analyze the emotional reactions to mass violent events on Twitter and influential factors. *Inf Process Manag* 57:2372. <https://doi.org/10.1016/j.ipm.2020.102372>
- Harris ZS (1954) *Distributional structure*. WORD, Rutledge, Taylor Fr Gr 10:146–162. <https://doi.org/10.1080/00437956.1954.11659520>
- Hasni S, Faiz S (2021) Word embeddings and deep learning for location prediction: tracking Coronavirus from British and American tweets. *Soc Netw Anal Min*. <https://doi.org/10.1007/s13278-021-00777-5>
- Hu K, Luo Q, Qi K et al (2019) Understanding the topic evolution of scientific literatures like an evolving city: using Google Word2Vec model and spatial autocorrelation analysis. *Inf Process Manag* 56:1185–1203. <https://doi.org/10.1016/j.ipm.2019.02.014>
- Ihm S, Lee J, Park Y (2019) Skip-gram-KR : Korean word embedding for semantic clustering. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2019.2905252>
- Jang B, Kim M, Harerimana G et al (2020) Bi-LSTM model to increase accuracy in text classification: combining word2vec CNN and attention mechanism. *Appl Sci*. <https://doi.org/10.3390/app10175841>
- Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: *Proc 2014 Conf Empir Methods Nat Lang Process Assoc Comput Linguist EMNLP-ACL*, pp 1532–1543. <https://doi.org/10.3115/v1/D14-1162>
- Jeon S, Kim HK (2021) AutoVAES: an automated vulnerability analysis system with a deep learning approach. *Comput Secur* 106:102308. <https://doi.org/10.1016/j.cose.2021.102308>
- Ji S, Satish N, Li S, Dubey PK (2019) Parallelizing word2vec in shared and distributed memory. *IEEE Trans Parallel Distrib Syst* 30:2090–2100. <https://doi.org/10.1109/TPDS.2019.2904058>
- Jiang L, Sun X, Mercaldo F, Santone A (2020) DECAB-LSTM: deep contextualized attentional bidirectional LSTM for cancer hallmark classification. *Knowl-Based Syst* 210:106486. <https://doi.org/10.1016/j.knosys.2020.106486>
- Jiang L, Sun X, Mercaldo F, Santone A (2020) DECAB-LSTM: deep contextualized attentional bidirectional LSTM for cancer hallmark classification. *Knowl Based Syst* 210:6486. <https://doi.org/10.1016/j.knosys.2020.106486>
- Jiao Q, Zhang S (2021) A brief survey of word embedding and its recent development. In: *IAEAC 2021—IEEE 5th Adv Inf Technol Electron Autom Control Conf 2021*, pp 1697–1701. <https://doi.org/10.1109/IAEAC50856.2021.9390956>
- Jin K, Wi J, Kang K, Kim Y (2020) Korean historical documents analysis with improved dynamic word embedding. *Appl Sci* 10:1–12. <https://doi.org/10.3390/app10217939>
- Joulin A, Grave E, Bojanowski P, Mikolov T (2017) Bag of tricks for efficient text classification. In: *15th Conf Eur Chapter Assoc Comput Linguist EACL 2017 - Proc Conf*, vol 2, pp 427–431. <https://doi.org/10.18653/v1/e17-2068>

- Kalouli AL, De Paiva V, Crouch R (2019) Composing noun phrase vector representations. *Proc 4th Work Represent Learn NLP, Assoc Comput Linguist Repl4NLP-ACL-2019* 84–95. <https://doi.org/10.18653/v1/w19-4311>
- Kalyan KS, Sangeetha S (2021) BertMCN: mapping colloquial phrases to standard medical concepts using BERT and highway network. *Artif Intell Med* 112:102008. <https://doi.org/10.1016/j.artmed.2021.102008>
- Kapil P, Ekbal A (2020) A deep neural network based multi-task learning approach to hate speech detection. *Knowl-Based Syst* 210:106458. <https://doi.org/10.1016/j.knosys.2020.106458>
- Kastrati Z, Imran AS, Kurti A (2019) Integrating word embeddings and document topics with deep learning in a video classification framework. *Pattern Recogn Lett* 128:85–92. <https://doi.org/10.1016/j.patrec.2019.08.019>
- Khan W, Daud A, Alotaibi F et al (2020) Deep recurrent neural networks with word embeddings for Urdu named entity recognition. *ETRI J* 42:90–100. <https://doi.org/10.4218/etrij.2018-0553>
- Khan Z, Hussain MI, Iltaf N et al (2021) Contextual recommender system for E-commerce applications. *Appl Soft Comput* 109:107552. <https://doi.org/10.1016/j.asoc.2021.107552>
- Khanal J (2020) Identifying enhancers and their strength by the integration of word embedding and convolution neural network. *IEEE Access* 8:58369–58376. <https://doi.org/10.1109/ACCESS.2020.2982666>
- Kilimci ZH (2020) Sentiment analysis based direction prediction in bitcoin using deep learning algorithms and word embedding models. *Int J Intell Syst Appl Eng* 8:60–65. <https://doi.org/10.18201/ijisae.2020261585>
- Kilimci ZH, Duvar R (2020) An efficient word embedding and deep learning based model to forecast the direction of stock exchange market using twitter and financial news sites: a case of istanbul stock exchange (BIST 100). *IEEE Access* 8:188186–188198. <https://doi.org/10.1109/ACCESS.2020.3029860>
- Kim J, Jeong OR (2021) Mirroring vector space embedding for new words. *IEEE Access* 9:99954–99967. <https://doi.org/10.1109/ACCESS.2021.3096238>
- Kim N, Hong S (2021) Automatic classification of citizen requests for transportation using deep learning: case study from Boston city. *Inf Process Manag* 58:102410. <https://doi.org/10.1016/j.ipm.2020.102410>
- Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. In: *5th Int Conf Learn Represent ICLR 2017—Conf Track Proc*, pp 1–14. <https://doi.org/10.48550/arXiv.1609.02907>
- Kitchenham B (2004) Procedures for performing systematic reviews, version 1.0. *Empir Softw Eng* 33:1–26
- Koutsomitropoulos DA, Andriopoulos AD (2021) Thesaurus-based word embeddings for automated biomedical literature classification. *Neural Comput Appl*. <https://doi.org/10.1007/s00521-021-06053-z>
- Kozłowski D, Lannelongue E, Sautemont F et al (2020) A three-level classification of French tweets in ecological crises. *Inf Process Manag* 57:2284. <https://doi.org/10.1016/j.ipm.2020.102284>
- Kumar N, Suman RR, Kumar S (2021) Text classification and topic modelling of web extracted data. In: *2021 2nd Glob Conf Adv Technol GCAT 2021*, pp 2–9. <https://doi.org/10.1109/GCAT52182.2021.9587459>
- Lavanya PM, Sasikala E (2021) Deep learning techniques on text classification using Natural language processing (NLP) in social healthcare network: a comprehensive survey. In: *2021 3rd Int Conf Signal Process Commun ICPSC 2021*, pp 603–609. <https://doi.org/10.1109/ICSPC51351.2021.9451752>
- Li B, Drozdz A, Guo Y et al (2019a) Scaling Word2Vec on Big Corpus. *Data Sci Eng* 4:157–175. <https://doi.org/10.1007/s41019-019-0096-6>
- Li M, Sun Y, Lu H et al (2020a) Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems. *IEEE Internet Things J* 7:6266–6278. <https://doi.org/10.1109/JIOT.2019.2962914>
- Li S, Pan R, Luo H et al (2021) Adaptive cross-contextual word embedding for word polysemy with unsupervised topic modeling. *Knowl Based Syst* 218:106827. <https://doi.org/10.1016/j.knosys.2021.106827>
- Li X, Jiang H, Kamei Y, Chen X (2018) Bridging semantic gaps between natural languages and APIs with word embedding. *IEEE Trans Softw Eng* 46:1081–1097. <https://doi.org/10.1109/TSE.2018.2876006>
- Li X, Zhang H, Zhou XH (2020) Chinese clinical named entity recognition with variant neural structures based on BERT methods. *J Biomed Inform* 107:103422. <https://doi.org/10.1016/j.jbi.2020.103422>
- Li Y, Yang T (2018) Word embedding for understanding natural language: a survey. *Big Data Appl*. https://doi.org/10.1007/978-3-319-53817-4_4
- Li Z, Yang F, Luo Y (2019b) Context embedding based on Bi-LSTM in semi-supervised biomedical word sense disambiguation. *IEEE Access* 7:72928–72935. <https://doi.org/10.1109/ACCESS.2019.2912584>

- Liao S, Chen J, Wang Y, et al (2020) Embedding compression with isotropic iterative quantization. In: Assoc Adv Artif Intell (AAAI 2020)—34th AAAI Conf Artif Intell, pp 8336–8343. <https://doi.org/10.1609/aaai.v34i05.6350>
- Liao Z, Ni J (2021) Construction of Chinese synonymous nouns discrimination and query system based on the semantic relation of embedded system and LSTM. *Microprocess Microsyst* 82:103848. <https://doi.org/10.1016/j.micpro.2021.103848>
- Lippincott T, Shapiro P, Duh K, McNamee P (2019) JHU system description for the MADAR Arabic dialect identification shared task. In: Proc Fourth Arab Nat Lang Process Work Assoc Comput Linguist ANLP-ACL-2019, pp 264–268. <https://doi.org/10.18653/v1/w19-4634>
- Liu G, Lu Y, Shi K et al (2019) Mapping bug reports to relevant source code files based on the vector space model and word embedding. *IEEE Access* 7:78870–78881. <https://doi.org/10.1109/ACCESS.2019.2922686>
- Liu J, Gao L, Guo S et al (2021) A hybrid deep-learning approach for complex biochemical named entity recognition. *Knowl Based Syst* 221:106958. <https://doi.org/10.1016/j.knsys.2021.106958>
- Liu J, Zheng S, Xu G, Lin M (2021b) Cross-domain sentiment aware word embeddings for review sentiment analysis. *Int J Mach Learn Cybern* 12:343–354. <https://doi.org/10.1007/s13042-020-01175-7>
- Liu N, Shen B (2020) Aspect-based sentiment analysis with gated alternate neural network. *Knowl Based Syst* 188:105010. <https://doi.org/10.1016/j.knsys.2019.105010>
- Lu H, Jin C, Helu X et al (2022) DeepAutoD: research on distributed machine learning oriented scalable mobile communication security unpacking system. *IEEE Trans Netw Sci Eng* 9:2052–2065. <https://doi.org/10.1109/TNSE.2021.3100750>
- Luo C, Tan Z, Min G et al (2021) A novel web attack detection system for internet of things via ensemble classification. *IEEE Trans Ind Inform* 17:5810–5818. <https://doi.org/10.1109/TII.2020.3038761>
- Magna AAR, Allende-Cid H, Taramasco C et al (2020) Application of machine learning and word embeddings in the classification of cancer diagnosis using patient anamnesis. *IEEE Access* 8:106198–106213. <https://doi.org/10.1109/ACCESS.2020.3000075>
- Malla SJ, Alphonse PJA (2021) COVID-19 outbreak: an ensemble pre-trained deep learning model for detecting informative tweets. *Appl Soft Comput* 107:107495. <https://doi.org/10.1016/j.asoc.2021.107495>
- Mikolov T, Chen K, Corrado G, Dean J (2013a) Efficient estimation of word representations in vector space. In: 1st Int Conf Learn Represent ICLR 2013a - Work Track Proc, pp 1–12. <https://doi.org/10.48550/arXiv.1301.3781>
- Mikolov T, Sutskever Ilya, Chen K et al (2013) Distributed representations of words and phrases and their compositionality. *Adv Neural Inf Process Syst*. <https://doi.org/10.48550/arXiv.1310.4546>
- Mohamed EH, Moussa MES, Haggag MH (2020) An enhanced sentiment analysis framework based on pre-trained word embedding. *Int J Comput Intell Appl*. <https://doi.org/10.1142/S1469026820500315>
- Moradi M, Dashti M, Samwald M (2020) Summarization of biomedical articles using domain-specific word embeddings and graph ranking. *J Biomed Inform* 107:103452. <https://doi.org/10.1016/j.jbi.2020.103452>
- Morales-Garzón A, Gomez-Romero J, Martin-Bautista MJ (2021) A word embedding-based method for unsupervised adaptation of cooking recipes. *IEEE Access* 9:27389–27404. <https://doi.org/10.1109/ACCESS.2021.3058559>
- Moreo A, Esuli A, Sebastiani F (2021) Word-class embeddings for multiclass text classification. Springer, New York
- Mulki H, Haddad H, Gridach M, Babaoğlu I (2019) Syntax-ignorant N-gram embeddings for sentiment analysis of Arabic dialects. In: Proc Fourth Arab Nat Lang Process Work Assoc Comput Linguist ANLP-ACL-2019, pp 30–39. <https://doi.org/10.18653/v1/w19-4604>
- Phat NH, Anh NTM (2020) Vietnamese text classification algorithm using long short term memory and Word2Vec. *Artif Intell Knowl Data Eng* 19:1255–1279. <https://doi.org/10.15622/ia.2020.19.6.5>
- Naderalvojud B, Sezer EA (2020) Sentiment aware word embeddings using refinement and senti-contextualized learning approach. *Neurocomputing* 405:149–160. <https://doi.org/10.1016/j.neucom.2020.03.094>
- Nasar Z, Jaffry SW, Malik MK (2021) Named entity recognition and relation extraction: state-of-the-art. *ACM Comput Surv*. <https://doi.org/10.1145/3445965>
- Nasim Z (2020) On building an interpretable topic modeling approach for the Urdu language. In: Proc Twenty-Ninth Int Jt Conf Artif Intell Dr Consort Track, IJCAI-DCT-2020 5200–5201. <https://doi.org/10.24963/ijcai.2020/740>

- Nassif AB, Elnagar A, Shahin I, Henno S (2021) Deep learning for Arabic subjective sentiment analysis: challenges and research opportunities. *Appl Soft Comput* 98:106836. <https://doi.org/10.1016/j.asoc.2020.106836>
- Nguyen D, Grieve J (2020) Do word embeddings capture spelling variation? In: *Proc 28th Int Conf Comput Linguist COLING* pp 870–881. <https://doi.org/10.18653/v1/2020.coling-main.75>
- Ning G, Bai Y (2021) Biomedical named entity recognition based on Glove-BLSTM-CRF model. *J Comput Methods Sci Eng* 21:125–133. <https://doi.org/10.3233/JCM-204419>
- Ochodek M, Koczyńska S, Staron M (2020) Deep learning model for end-to-end approximation of COSMIC functional size based on use-case names. *Inf Softw Technol.* <https://doi.org/10.1016/j.infsof.2020.106310>
- Ohashi S, Isogawa M, Kajiwara T, Arase Y (2020) Tiny Word Embeddings Using Globally Informed Reconstruction. *Proc 28th Int Conf Comput Linguist COLING* 1199–1203. <https://doi.org/10.18653/v1/2020.coling-main.103>
- Okoli C, Schabram K (2010) A guide to conducting a systematic literature review of information systems research. *Work Pap Inf Syst.* <https://doi.org/10.2139/ssrn.1954824>
- Onan A (2021) Sentiment analysis on product reviews based on weighted word embeddings and deep neural networks. *Concurr Comput Pract Exp* 33:1–12. <https://doi.org/10.1002/cpe.5909>
- Pan C, Huang J, Gong J, Yuan X (2019a) Few-shot transfer learning for text classification with light-weight word embedding based models. *IEEE Access* 7:53296–53304. <https://doi.org/10.1109/ACCESS.2019.2911850>
- Pan Q, Dong H, Wang Y, et al (2019b) Recommendation of crowdsourcing tasks based on Word2vec semantic tags. *Algorithm Optim Wirel Mob Appl Smart Cities.* <https://doi.org/10.1155/2019/2121850>
- Pandey B, Kumar Pandey D, Pratap Mishra B, Rhmann W (2021) A comprehensive survey of deep learning in the field of medical imaging and medical natural language processing: challenges and research directions. *J King Saud Univ.* <https://doi.org/10.1016/j.jksuci.2021.01.007>
- Pariqh P, Abburi H, Badjatiya P, et al (2019) Multi-label categorization of accounts of sexism using a neural framework. In: *Proc 2019 - Conf Empir Methods Nat Lang Process 9th Int Jt Conf Nat Lang Process Assoc Comput Linguist EMNLP-IJCNLP-ACL* 1642–1652. <https://doi.org/10.18653/v1/d19-1174>
- Pattisapu N, Gupta M, Kumaraguru P, Varma V (2019) A distant supervision based approach to medical persona classification. *J Biomed Inform* 94:3205. <https://doi.org/10.1016/j.jbi.2019.103205>
- Pennington J, Socher R, Manning CD (2014) GloVe: global vectors for word representation. <https://nlp.stanford.edu/projects/glove/>. Accessed 10 Jun 2021
- Peters ME, Neumann M, Iyyer M, et al (2018) Deep contextualized word representations. In: *NAACL HLT 2018 - 2018 Conf North Am Chapter Assoc Comput Linguist Hum Lang Technol - Proc Conf* 1:2227–2237. <https://doi.org/10.18653/v1/n18-1202>
- Qiu J, Chai Y, Tian Z et al (2020a) Automatic concept extraction based on semantic graphs from big data in smart city. *IEEE Trans Comput Soc Syst* 7:225–233. <https://doi.org/10.1109/TCSS.2019.2946181>
- Qiu J, Du L, Zhang D et al (2020b) Nei-TTE: intelligent traffic time estimation based on fine-grained time derivation of road segments for smart city. *IEEE Trans Ind Inform* 16:2659–2666. <https://doi.org/10.1109/TII.2019.2943906>
- Qiu Q, Xie Z, Wu L, Li W (2019) Geoscience keyphrase extraction algorithm using enhanced word embedding. *Expert Syst Appl* 125:157–169. <https://doi.org/10.1016/j.eswa.2019.02.001>
- Racharak T (2021) On approximation of concept similarity measure in description logic ELH with pre-trained word embedding. *IEEE Access* 9:61429–61443. <https://doi.org/10.1109/ACCESS.2021.3073730>
- Radford A, Wu J, Child R, et al (2019) Language models are unsupervised multitask learners. 1:OpenAI blog
- Raunak V, Gupta V, Metz F (2019) Effective Dimensionality Reduction for Word Embeddings. N: *Proc 4th Work Represent Learn NLP, Assoc Comput Linguist Repl4NLP-ACL-2019* 235–243. <https://doi.org/10.18653/v1/W19-4328>
- Ren Z, Shen Q, Diao X, Xu H (2021) A sentiment-aware deep learning approach for personality detection from text. *Inf Process Manag* 58:2532. <https://doi.org/10.1016/j.ipm.2021.102532>
- Rethmeier N, Plank B (2019) MoRTy: unsupervised learning of task-specialized word embeddings by autoencoding. In: *Proc 4th Work Represent Learn NLP, Assoc Comput Linguist Repl4NLP-ACL-2019* 49–54. <https://doi.org/10.18653/v1/w19-4307>
- Rezaeinia SM, Rahmani R, Ghodsi A, Veisi H (2019) Sentiment analysis based on improved pre-trained word embeddings. *Expert Syst Appl* 117:139–147. <https://doi.org/10.1016/j.eswa.2018.08.044>

- Rida-e-fatima S, Javed A, Banjar A et al (2019) A multi-layer dual attention deep learning model with refined word embeddings for aspect-based sentiment analysis. *IEEE Access* 7:114795–114807. <https://doi.org/10.1109/ACCESS.2019.2927281>
- Risch J, Krestel R, Risch J, Krestel R (2019). Domain-Specific Word Embeddings for Patent Classification. <https://doi.org/10.1108/DTA-01-2019-0002>
- Roman M, Shahid A, Khan S et al (2021) Citation intent classification using word embedding. *IEEE Access* 9:9982–9995. <https://doi.org/10.1109/ACCESS.2021.3050547>
- Roy PK, Singh JP, Banerjee S (2020) Deep learning to filter SMS Spam. *Futur Gener Comput Syst* 102:524–533. <https://doi.org/10.1016/j.future.2019.09.001>
- Salton G, Wong A, Yang CS (1975) A vector space model for automatic indexing. *Commun ACM* 18:613–620. <https://doi.org/10.1145/361219.361220>
- Scott D, Richard H, Susan T et al (1990) Indexing by latent semantic analysis. *J Am Soc Inf Sci* 41:391–407. <https://doi.org/10.1002/1097-4571>
- See A (2019) Natural language processing with deep learning: natural language generation. 2022:1–39
- Shahzad K, Kanwal S, Malik K et al (2019) A word-embedding-based approach for accurate identification of corresponding activities. *Comput Electr Eng* 78:218–229. <https://doi.org/10.1016/j.compeleceng.2019.07.011>
- Shaikh S, Daudpotta SM, Imran AS (2021) Bloom’s learning outcomes’ automatic classification using LSTM and pretrained word embeddings. *IEEE Access* 9:117887–117909. <https://doi.org/10.1109/access.2021.3106443>
- Sharma M, Kandasamy I, Vasantha WB (2021) Comparison of neutrosophic approach to various deep learning models for sentiment analysis. *Knowledge-Based Syst* 223:107058. <https://doi.org/10.1016/j.knosys.2021.107058>
- Shekhar S, Sharma DK, Sufyan Beg MM (2019) An effective cybernated word embedding system for analysis and language identification in code-mixed social media text. *Int J Knowl-Based Intell Eng Syst* 23(3):167–79. <https://doi.org/10.3233/KES-190409>
- Shi W, Chen M, Tian Y, Chang KW (2019) Learning bilingual word embeddings using lexical definitions. In: *Proc 4th Work Represent Learn NLP, Assoc Comput Linguist Repl4NLP-ACL-2019* 142–147. <https://doi.org/10.18653/v1/w19-4316>
- Shin B, Yang H, Choi JD (2019) The pupil has become the master: teacher-student model-based word embedding distillation with ensemble learning. In: *Proc Twenty-Eighth Int Jt Conf Artif Intell IJCAI-2019* 2019-Augus:3439–3445. <https://doi.org/10.24963/ijcai.2019/477>
- Shin HS, Kwon HY, Ryu SJ (2020) A new text classification model based on contrastive word embedding for detecting cybersecurity intelligence in twitter. *Electron* 9:1–21. <https://doi.org/10.3390/electronic9091527>
- Smetanin S, Komarov M (2021) Deep transfer learning baselines for sentiment analysis in Russian. *Inf Process Manag* 58:2484. <https://doi.org/10.1016/j.ipm.2020.102484>
- Song M, Park H, Shin Shik K (2019) Attention-based long short-term memory network using sentiment lexicon embedding for aspect-level sentiment analysis in Korean. *Inf Process Manag* 56:637–653. <https://doi.org/10.1016/j.ipm.2018.12.005>
- Spinde T, Rudnitskaia L, Mitrović J et al (2021) Automated identification of bias inducing words in news articles using linguistic and context-oriented features. *Inf Process Manag* 58:102505. <https://doi.org/10.1016/j.ipm.2021.102505>
- Suárez-Paniagua V, Rivera Zavala RM, Segura-Bedmar I, Martínez P (2019) A two-stage deep learning approach for extracting entities and relationships from medical texts. *J Biomed Inform* 99:3285. <https://doi.org/10.1016/j.jbi.2019.103285>
- Sun G, Li Y, Yu H, Chang V (2020) Attention distribution guided information transfer networks for recommendation in practice. *Appl Soft Comput J*. <https://doi.org/10.1016/j.asoc.2020.106772>
- Sun Z, Sarma PK, Sethares WA, Liang Y (2020b) Learning relationships between text, audio, and video via deep canonical correlation for multimodal language analysis. *Assoc Adv Artif Intell (AAAI 2020b)*—34th AAAI Conf Artif Intell 8992–8999. <https://doi.org/10.1609/aaai.v34i05.6431>
- Talafha B, Farhan W, Altakrouri A, Al-Natsheh HT (2019) Mawdoo3 AI at MADAR Shared Task: Arabic Tweet Dialect Identification. *Proc Fourth Arab Nat Lang Process Work Assoc Comput Linguist ANLP-ACL-2019* 239–243. <https://doi.org/10.18653/v1/w19-4629>
- TensorFlow Hub BERT. https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4. Accessed 14 Mar 2022
- Tian G, Zhao S, Wang J et al (2019) Semantic sparse service discovery using word embedding and Gaussian LDA. *IEEE Access* 7:88231–88242. <https://doi.org/10.1109/ACCESS.2019.2926559>
- Toor AS, Wechsler H, Nappi M (2019) Biometric surveillance using visual question answering. *Pattern Recogn Lett* 126:111–118. <https://doi.org/10.1016/j.patrec.2018.02.013>

- Torregrossa F, Allesiardo R, Claveau V et al (2021) A survey on training and evaluation of word embeddings. *Int J Data Sci Anal* 11:85–103. <https://doi.org/10.1007/s41060-021-00242-8>
- Dinter VR, Catal C, Tekinerdogan B (2021) A multi-channel convolutional neural network approach to automate the citation screening process. *Appl Soft Comput* 112:7765. <https://doi.org/10.1016/j.asoc.2021.107765>
- Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. *Adv Neural Inf Process Syst*. <https://doi.org/10.48550/arXiv.1706.03762>
- Vazirgiannis M (2017) Graph of words: boosting text mining with graphs. *Int World Wide Web Conf Commun*. <https://doi.org/10.1145/3041021.3055362>
- Verma P, Khandelwal B (2019) Word embeddings and its application in deep learning. *Int J Innov Technol Explor Eng* 8:337–341. <https://doi.org/10.35940/ijitee.K1343.0981119>
- Vijayvergia A, Kumar K (2021) Selective shallow models strength integration for emotion detection using GloVe and LSTM. *Multimed Tools Appl* 80:28349–28363. <https://doi.org/10.1007/s11042-021-10997-8>
- Wang B, Kuo CCJ (2020) SBERT-WK: a sentence embedding method by dissecting BERT-based word models. *IEEE/ACM Trans Audio Speech Lang Process* 28:2146–2157. <https://doi.org/10.1109/TASLP.2020.3008390>
- Wang L, Zhang J, Chen G, Qiao D (2021) Identifying comparable entities with indirectly associative relations and word embeddings from web search logs. *Decis Support Syst* 141:113465. <https://doi.org/10.1016/j.dss.2020.113465>
- Wang P, Luo Y, Chen Z et al (2019) Orientation analysis for Chinese news based on word embedding and syntax rules. *IEEE Access* 7:159888–159898. <https://doi.org/10.1109/ACCESS.2019.2950900>
- Wang S, Cao J, Yu PS (2022) Deep learning for spatio-temporal data mining: a survey. *IEEE Trans Knowl Data Eng* 34:3681–3700. <https://doi.org/10.1109/TKDE.2020.3025580>
- Wang S, Tseng B, Hernandez-Boussard T (2021) Development and evaluation of novel ophthalmology domain-specific neural word embeddings to predict visual prognosis. *Int J Med Inform* 150:104464. <https://doi.org/10.1016/j.ijmedinf.2021.104464>
- Wang S, Zhou W, Jiang C (2020) A survey of word embeddings based on deep learning. *Computing* 102:717–740. <https://doi.org/10.1007/s00607-019-00768-7>
- Wang Y, Huang G, Li J et al (2021c) Refined global word embeddings based on sentiment concept for sentiment analysis. *IEEE Access* 9:37075–37085. <https://doi.org/10.1109/ACCESS.2021.3062654>
- Warnecke A, Arp D, Wressnegger C, Rieck K (2020) Evaluating explanation methods for deep learning in security. In: *Proc—5th IEEE Eur Symp Secur Privacy-2020* 158–174. <https://doi.org/10.1109/EuroSP48549.2020.00018>
- Wen G, Chen H, Li H et al (2020) Cross domains adversarial learning for Chinese named entity recognition for online medical consultation. *J Biomed Inform* 112:3608. <https://doi.org/10.1016/j.jbi.2020.103608>
- Wu C, Gao R, Zhang Y, De Marinis Y (2019) PTPD: predicting therapeutic peptides by deep learning and word2vec. *BMC Bioinform* 20:1–8. <https://doi.org/10.1186/s12859-019-3006-z>
- Wu L, Cui P, Pei J, Zhao L (2022) *Graph neural networks: foundations, frontiers, and applications*. Springer, Singapore
- Xiao Y, Fan Z, Tan C et al (2019) Sense-based topic word embedding model for item recommendation. *IEEE Access* 7:44748–44760. <https://doi.org/10.1109/ACCESS.2019.2909578>
- Xiao Y, Keung J, Bennin KE, Mi Q (2018) Improving bug localization with word embedding and enhanced convolutional neural networks. *Inf Softw Technol*. <https://doi.org/10.1016/j.infsof.2018.08.002>
- Xiong J, Yu L, Zhang D, Leng Y (2021) DNCP: an attention-based deep learning approach enhanced with attractiveness and timeliness of News for online news click prediction. *Inf Manag*. <https://doi.org/10.1016/j.im.2021.103428>
- Xu D, Tian Z, Lai R et al (2020) Deep learning based emotion analysis of microblog texts. *Inf Fusion* 64:1–11. <https://doi.org/10.1016/j.inffus.2020.06.002>
- Yang C, Zhou W, Wang Z, et al (2021a) Accurate and Explainable Recommendation via Hierarchical Attention Network Oriented Towards Crowd Intelligence. *Knowledge-Based Syst* 213:106687. <https://doi.org/10.1016/j.knosys.2020.106687>
- Yang J, Liu Y, Qian M, et al (2019) Information extraction from electronic medical records using multitask recurrent neural network with contextual word embedding. *Appl Sci* 9:. <https://doi.org/10.3390/app9183658>
- Yang R, Wu F, Zhang C, Zhang L (2021b) iEnhancer-GAN: A Deep Learning Framework in Combination with Word Embedding and Sequence Generative Adversarial Net to Identify Enhancers and Their Strength. *Int J Mol Sci* 22:. <https://doi.org/10.3390/ijms22073589>

- Yao L, Mao C, Luo Y (2019) Graph Convolutional Networks for Text Classification. Thirty-Third AAAI Conf Artif Intell 19. <https://doi.org/10.1609/aaai.v33i01.33017370>
- Yi MH, Lim MJ, Ko H, Shin JH (2021) Method of Profanity Detection Using Word Embedding and LSTM. *Mob Inf Syst* 2021:1. <https://doi.org/10.1155/2021/6654029>
- Yildirim S (2019) Improving word embeddings projection for Turkish hypernym extraction. 4418–4428. <https://doi.org/10.3906/elk-1903-65>
- Yildiz B, Tezgider M (2021) Improving word embedding quality with innovative automated approaches to hyperparameters. *Concurr Comput Pract Exp* 33:1–10. <https://doi.org/10.1002/cpe.6091>
- Yilmaz S, Toklu S (2020) A deep learning analysis on question classification task using Word2vec representations. *Neural Comput Appl* 32:2909–2928. <https://doi.org/10.1007/s00521-020-04725-w>
- Young T, Hazarika D, Poria S, Cambria E (2018) Recent trends in deep learning based natural language processing. *IEEE Comput Intell Mag* 13:55–75. <https://doi.org/10.1109/MCI.2018.2840738>
- Yusuf SM, Zhang F, Zeng M, Li M (2021) DeepPPF: a deep learning framework for predicting protein family. *Neurocomputing* 428:19–29. <https://doi.org/10.1016/j.neucom.2020.11.062>
- Zhang Y, Liu Y, Zhu J, Wu X (2021) FSPRM: a feature subsequence based probability representation model for Chinese word embedding. *IEEE/ACM Trans Audio Speech Lang Process* 29:1702–1716. <https://doi.org/10.1109/TASLP.2021.3073868>
- Zhang Y, Yu X, Cui Z et al (2020) Every document owns its structure: inductive text classification via graph neural networks. In: 58th Annu Meet Assoc Comput Linguist, pp 334–339. <https://doi.org/10.18653/v1/2020.acl-main.31>
- Zhao H, Phung D, Huynh V, et al (2021) Topic Modelling Meets Deep Neural Networks: A Survey. 4713–4720. <https://doi.org/10.24963/ijcai.2021/638>
- Zhelezniak V, Shen A, Busbridge D, et al (2019) Correlations between Word Vector Sets. *Proc 2019 - Conf Empir Methods Nat Lang Process 9th Int Jt Conf Nat Lang Process Assoc Comput Linguist EMNLP-IJCNLP-ACL* 77–87. <https://doi.org/10.18653/v1/d19-1008>
- Zheng C, Fan H, Shi Y (2020) A Domain expertise and word-embedding geometric projection based semantic mining framework for measuring the soft power of social entities. *IEEE Access* 8:204597–204611. <https://doi.org/10.1109/ACCESS.2020.3037462>
- Zhu W, Liu S, Liu C et al (2020a) Learning multimodal word representations by explicitly embedding syntactic and phonetic information. *IEEE Access* 8:223306–223315. <https://doi.org/10.1109/ACCESS.2020.3042183>
- Zhu Y, Li Y, Yue Y et al (2020b) A hybrid classification method via character embedding in chinese short text with few words. *IEEE Access* 8:92120–92128. <https://doi.org/10.1109/ACCESS.2020.2994450>
- Zobnin A, Elistratova E (2019) Learning Word Embeddings without Context Vectors. *Proc 4th Work Represent Learn NLP, Assoc Comput Linguist Repl4NLP-ACL-2019* 244–249. <https://doi.org/10.18653/v1/w19-4329>
- Zuheros C, Tabik S, Valdivia A et al (2019) Deep recurrent neural network for geographical entities disambiguation on social media data. *Knowledge-Based Syst* 173:117–127. <https://doi.org/10.1016/j.knsys.2019.02.030>
- Zulqarnain M, Ghazali R, Ghouse MG, Mushtaq MF (2019) Efficient processing of GRU based on word embedding for text classification. *Int J Informatics Vis* 3:377–383. <https://doi.org/10.30630/joyv.3.4.289>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.