

RESEARCH ARTICLE

Enhanced Word-Unit Broad Learning System With Sememes

YUCHAO JIANG¹, JIADONG LU², TONGFEI LI², AND WEI LV²¹School of Computer Science, Zhuhai College of Science and Technology, Zhuhai 519041, China²School of Aliyun Big Data Application, Zhuhai College of Science and Technology, Zhuhai 519041, China

Corresponding author: Wei Lv (luwei@zcst.edu.cn)

This work was supported by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515310003, Guangdong Universities' key scientific research platforms and projects under Grant 2023ZDZX1049, and the Innovation and Entrepreneurship Training Programme of Zhuhai College of Science and Technology under Grant S202213684012, S202313684038 and S202313684039.

ABSTRACT High accuracy in text classification can be achieved by simultaneously learning multiple sources of information, such as sequence and word. In this study, we propose a novel learning framework for text classification, called Word-unit Broad Learning System (BLS). The Word-unit BLS utilises a flat neural network known as BLS and offers three key advantages. First, it provides higher accuracy and shorter training time compared to popular machine learning methods, allowing for the simultaneous learning of sequence information and word importance. Second, we incorporate a multi-layer perceptron with attention aggregation in the *feature-mapped* layer, along with position encoding, to capture the latent relationship between each word and the contextual information in a global context. Lastly, we introduce a novel approach to enhance word representation by employing sememes in the *enhancement node* layer, thereby improving the feature distribution of each word in the vector space. The effectiveness of the proposed framework was evaluated by conducting experiments on four datasets covering various types of text classifications. The results demonstrate that Word-unit BLS achieves 8.26% higher accuracy than Naive Bayes while requiring 1/33 of the training time. Furthermore, when compared with traditional BLS models, Word-unit BLS outperforms in learning the sequence information. The effectiveness of sememe enhancement in word representation is also demonstrated, particularly in the case of large-scale datasets.


INDEX TERMS Broad learning system, natural language processing, neural network, sememe, sequences, simultaneous learning, text classification.

I. INTRODUCTION

Text classification is crucial component of Natural Language Processing (NLP) and finds applications in various downstream tasks, including information retrieval [1], sentiment analysis [2], and spam detection [3]. The ability of a model to effectively extract relevant information from the context [4], [5], [6] is a key factor contributing to the success of text classification. Several techniques have been proposed to enhance the performance of word/text representation, such as Word2Vec [7], GloVe [8], FastText [9], and Transformer [5]. Additionally, researchers have integrated lexical knowledge

bases [10], [11], [12] into models for feature extraction in NLP tasks, thereby reducing feature dimensions and improving downstream task accuracy [13].

Currently, the state-of-the-art (SOTA) approach for text representation is the extraction of content information using Bidirectional Encoder Representation from Transformers (BERT) [14], a pretrained language model with a deep bidirectional transformer [5] architecture that captures relationships between phrases and words. It can be fine-tuned for specific tasks and has demonstrated high accuracy across a range of NLP tasks. However, applying BERT poses challenges due to the large number of parameters and the need for significant computing resources, especially for researchers with limited access to high-performance

The associate editor coordinating the review of this manuscript and approving it for publication was Francisco J. Garcia-Penalvo .

computing resources [15]. To address these issues, Sanh et al. developed a knowledge distillation approach that transfers the knowledge acquired by a large BERT model to a smaller model, ensuring comparable performance while reducing the computational cost and memory requirement [16]. Another approach is RoBERTa [17], which introduces a training procedure that iterates on BERT's pretraining. It incorporates dynamic masking, FULL-SENTENCES without next sentence prediction loss, large mini-batches, and a larger byte-level byte-pair encoding to reduce the memory requirement and achieve faster inference time. However, it is important to note that deep learning models, due to their large number of parameters and computational complexity, present challenges. Training and optimising these models require substantial amounts of data and computational resources [18].

Recently, Broad Learning System (BLS), a flattened feed-forward neural network, has gained considerable attention owing to its compact network structure. In the original BLS approach [19], inputs are converted into feature nodes within the *feature-mapped* layer, with weights and biases of these feature nodes randomly generated. The network structure is expanded in the *enhancement node* layer, and weights and biases for these feature nodes are also randomly generated. Ridge regression of the pseudoinverse is then employed to estimate the connecting weights from the *feature-mapped* layer and *enhancement node* layer to the output layer. BLS can effectively handle multiple classification or regression tasks with low generalisation errors and minimal computing resource consumption [20], [21]. However, BLS assumes independent inputs [22], utilising the entire data matrix X as input without considering the latent correlations between features. For example, in the input, "I love Zhuhai because of its pleasant climate," the phrase "I love Zhuhai" is contextually dependent on "its (Zhuhai's) pleasant climate." Thus, "I love Zhuhai" and "its (Zhuhai's) pleasant climate" are relevant rather than independent. Consequently, BLS does not perform well on tasks involving continuous input features [22], [23].

Several advancements have been proposed to enhance the feature selection schemes in BLS, e.g. Tree-based BLS [24], Multi-Attn BLS [25], and self-paced BLS [26]. There are two main strategies for improving the information extraction capability of the BLS: 1) data augmentation prior to transferring the input data as mapped features; 2) renovating the structure/learning scheme within the *feature-mapped* layer. In [22], two approaches were introduced to enhance sequential modelling performance in text classification tasks in the *feature-mapped* layer: recurrent broad learning system (R-BLS), which is an RNN (recurrent neural network)-like network, and a gated broad learning system (G-BLS), which is an LSTM (long short-term memory)-like network. These approaches replace the randomly generated feature nodes of the original BLS with different recurrent architectures. They optimise the parameters of the constructed recurrent structure and perform ridge regression in each iteration. In [23], variational attention BLS (VABLS) was

proposed. VABLS incorporates a variational form of the expectation-maximisation attention mechanism to represent the sequential information by adding an auxiliary mapped feature within the *feature-mapped* layer. VABLS introduces the attention mechanism into BLS after position encoding by using a non-iterative structure, providing advantages in terms of fitting, calculation, and flexibility. However, improving the performance of BLS remains challenging, particularly in NLP scenarios with limited data [22], [27], such as short text tasks and question answering tasks.

Considering this, we propose a framework for NLP tasks based on BLS at the **word-unit** level, *a.k.a* **Word-unit BLS**. This framework comprehensively reorganises the feature representation in a language context containing sequential information while retaining the efficiency of the *feature-mapped* layer and *enhancement node* layer to minimise computational resource requirements and ensure fast processing. We incorporate a multi-layer perceptron (MLP) with attention aggregation to extract the contextual information and introduce a novel enhancement scheme for learning the word importance using sememes derived from HowNet. The contributions of our study are as follows:

- 1) We propose a novel framework called Word-unit BLS that enhances the model's understanding of text by incorporating linguistic theory and improving the learning scheme of features in the *feature-mapped* layer and *enhancement node* layer of the original BLS. Hence, Word-unit BLS proves to be more effective for NLP tasks, as it boosts performance while minimizing computational resource requirements.
- 2) We constructed a model named MLP with attention aggregation, capable of effectively extracting both context and sequence information after position encoding. This model employs the attention mechanism to capture the underlying relationship between each word and the contextual information in a global context.
- 3) We incorporated HowNet into BLS to enhance the word representation performance of the Word-unit BLS framework. By incorporating sememes, we enhanced the feature distribution of each word in the vector space through the utilisation of an external knowledge base for the language. To the best of our knowledge, this is the first attempt at improving the learning strategy by generating the enhancement nodes of BLS.
- 4) We evaluated the performance of Word-unit BLS on four real-world datasets—*THUCNews*, *Chinese_news*, *ChnSentiCorp_h1l*, and *apple-twitter-sentiment*. We assessed the framework's effectiveness in text classification and sentiment analysis tasks, considering accuracy and training time as the primary performance metrics.

The rest of this paper is organised as follows. In Section I, we briefly introduce the techniques utilised in this work, including BLS, MLP [28], and HowNet [10]. Section I-B presents the Word-unit BLS framework for NLP tasks. Section II describes the dataset setup, experimental setup,

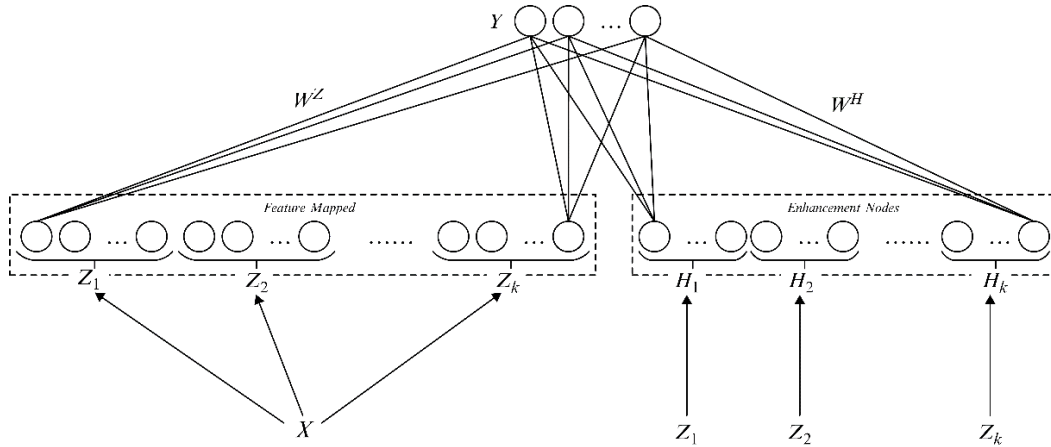


FIGURE 1. Structure of broad learning system (BLS).

and experimental results. It presents the results of the comparison experiments and ablation experiments. Section III discusses the key results of the study and its limitations. Finally, Section III-B presents the conclusions.

RELATED WORK

A. BROAD LEARNING SYSTEM

$$Z_p = f(X\alpha_p^z + \beta_p^z) \tag{1}$$

$$H_p = g(Z_p\alpha_p^h + \beta_p^h) \tag{2}$$

$$Y = [Z_1, \dots, Z_k | H_1, \dots, H_k]W \tag{3}$$

$$W = A^+Y \tag{4}$$

$$A^+ = (A^T A)^{-1} A^T, A = [Z_1, \dots, Z_k | H_1, \dots, H_k] \tag{5}$$

BLS, a flat feedforward neural network proposed in [19], extracts data features using randomly generated feature nodes and enhancement nodes. The weights of the model are obtained by utilising a pseudoinverse solution exclusively in the output layer [29], [30], [31]. This characteristic enhances the efficiency and effectiveness of the BLS. When the model is initially too simplistic to adequately fit the data or target function, BLS leverages the mathematical properties of the pseudoinverse solutions to incrementally increase the feature nodes, enhancement nodes, and even the input data. Incremental learning in BLS can be achieved with minimal computational effort, without recomputing the network output weights from scratch. By optimising the parameters of ridge regression, the model’s generalisation performance can be enhanced. In summary, BLS can effectively fit any function, given an appropriate width and parameter configuration [32]. An overview of the original BLS structure is shown in Fig. 1.

1) CONSTRUCTION OF FEATURE-MAPPED LAYER IN BLS

Consider N training data $\{x^i, y^i\}$, $i = 1$ to N , where $x^i \in \mathbb{R}^{1 \times D}$ represents the i^{th} training sample with the corresponding target output y^i . In the matrix form, $X = [x^i] \in \mathbb{R}^{N \times D}$

is the input matrix and $Y = [y^i] \in \mathbb{R}^{N \times m}$ is the output matrix, where D is the dimension of the input vector x^i and m is the number of class labels. For k features, Z_p , mapped from X , where $p = 1$ to k , each Z_p has l_z hidden nodes and can be represented as (1), where k and l_z are hyper parameters and f is an activation function, such as a sigmoid. $\alpha_p^z \in \mathbb{R}^{D \times l_z}$ and $\beta_p^z \in \mathbb{R}^{N \times l_z}$ are the random weights and biases under the standard normal cumulative distribution for the input X , respectively. Therefore, each Z_p is obtained with the dimension of $N \times l_z$.

2) CONSTRUCTION OF ENHANCEMENT NODE LAYER IN BLS

As in the case of the *feature-mapped* layer, the enhancement nodes H_p with l_h hidden nodes, $p = 1 \dots k$, are obtained using (2), where g is an activation function, which is the same as f . $\alpha_p^h \in \mathbb{R}^{l_z \times l_h}$ and $\beta_p^h \in \mathbb{R}^{N \times l_h}$ are randomly generated weights and biases for the mapped features Z_p . Hence, H_p is obtained with the dimension of $N \times l_h$. Then, the *output nodes* Y can be represented as a wide or broad structure in (3).

3) CALCULATION OF CONNECTING WEIGHT IN BLS

Following [19], the connecting weight for this BLS structure is shown in (4), which can be computed by applying ridge regression approximation and pseudoinverse, as shown in (5).

Moreover, three incremental learning algorithms of BLS are also applied to the feature nodes, enhancement nodes, and new incoming inputs, as in [19].

B. ATTENTION MECHANISM

$$attention_score_i = V^T \tanh(w_1 x_i + w_2 Q + b) \tag{6}$$

$$attention_weight_i = softmax(attention_score_i) \tag{7}$$

$$context_vector = \sum_{i=1}^n x_i \cdot attention_weight_i \tag{8}$$

The attention mechanism is an approach used to enhance the performance of models when processing sequential data in language modelling tasks. It enables the model to selectively focus on specific parts of the input by generating a context

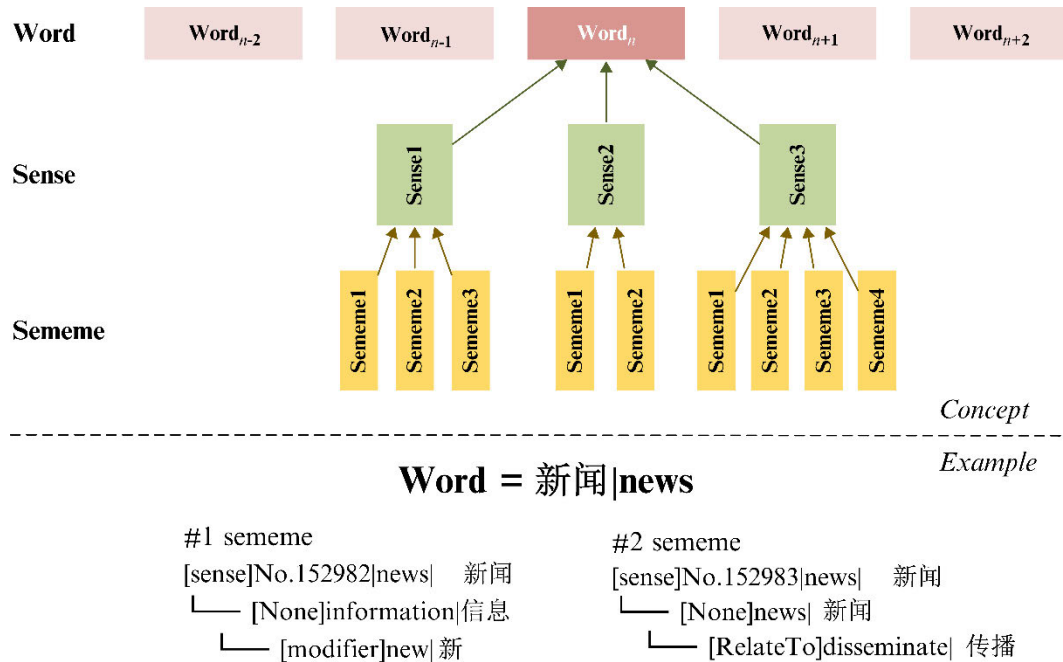


FIGURE 2. Structure of HowNet. Concept of HowNet (top) and example of HowNet (bottom).

vector. This context vector captures the information contained in an entire sentence by assigning different weights to the word vectors and summing them. The attention mechanism has been applied to various tasks, e.g. text classification [33], machine translation [34], and speech recognition [35]. Its implementation has consistently demonstrated improvements in the accuracy and efficiency of models handling sequential data, enabling them to better capture the relevant information within the underlying structure of the data.

Given a sentence $X = (x_1, x_2, x_3, \dots, x_n) \in \mathbb{R}^D$, where n represents the number of words in the entire sentence, each $x_i, i \in n$ is concatenated to a matrix form X ; D represents the dimension vector of the word, which is converted into a vector space through the use of word embedding techniques. Let $Q \in \mathbb{R}^D$ be a vector that represents a particular context of the sentence generated by a reliable model, where D represents the dimension of the vector, which is the same as that of the word. The attention score for each attribution x_i is calculated using the dot product with its weight by employing an activation function $\tanh(\cdot)$, as shown in (6), where w_1, w_2, V are learnable weight matrices and b is a learnable bias. The $\text{softmax}(\cdot)$ function is utilized to transform a vector of real values into a probability distribution that is contextually relevant. This distribution ensures that the values range between 0 and 1 and sum up to 1. In our approach, this function is applied to the attention scores, resulting in a probability distribution over the words as depicted in (7). Finally, the output/context vector is computed by taking the weighted sum of all n initially assigned vectors, as illustrated in (8).

The additive attention mechanism is typically used in conjunction with an MLP. There are several other popular

attention mechanisms widely used in NLP, e.g. dot product attention [36] and self-attention [5]. In [23], Hu et al. combines the expectation-maximisation attention network [37] with BLS for enhancing series feature extraction, which uses iteration, instead of the BP algorithm, to determine the attention values.

C. HOWNET

HowNet [10] is recognised as one of the most extensive lexical knowledge bases currently available. It annotates each concept in Chinese with one or more relevant sememes. HowNet consists of 2,000 sememes, which are used to annotate over 200,000 senses of English and Chinese words. This resource has been widely employed in various NLP tasks, e.g. word similarity computation [38] and sentiment analysis [39].

HowNet serves as a knowledge representation system based on ontology, providing a comprehensive understanding of the relationships between concepts. It incorporates an ontological view of the world through a hierarchical structure consisting of four top-level categories. At the core of HowNet lies a set of sememes, which are the most fundamental semantic units that cannot be further decomposed. These sememes are derived bottom-up from approximately 6,000 Chinese characters. The hierarchy of over 2,000 sememes ensures that all concepts can be expressed using combinations of existing Chinese characters. This bottom-up approach guarantees the stability and robustness of the sememe set, as demonstrated through successful verification of over 65,000 concepts. The structure and example of HowNet are shown in Fig. 2. HowNet was implemented in OpenHowNet API [40], which

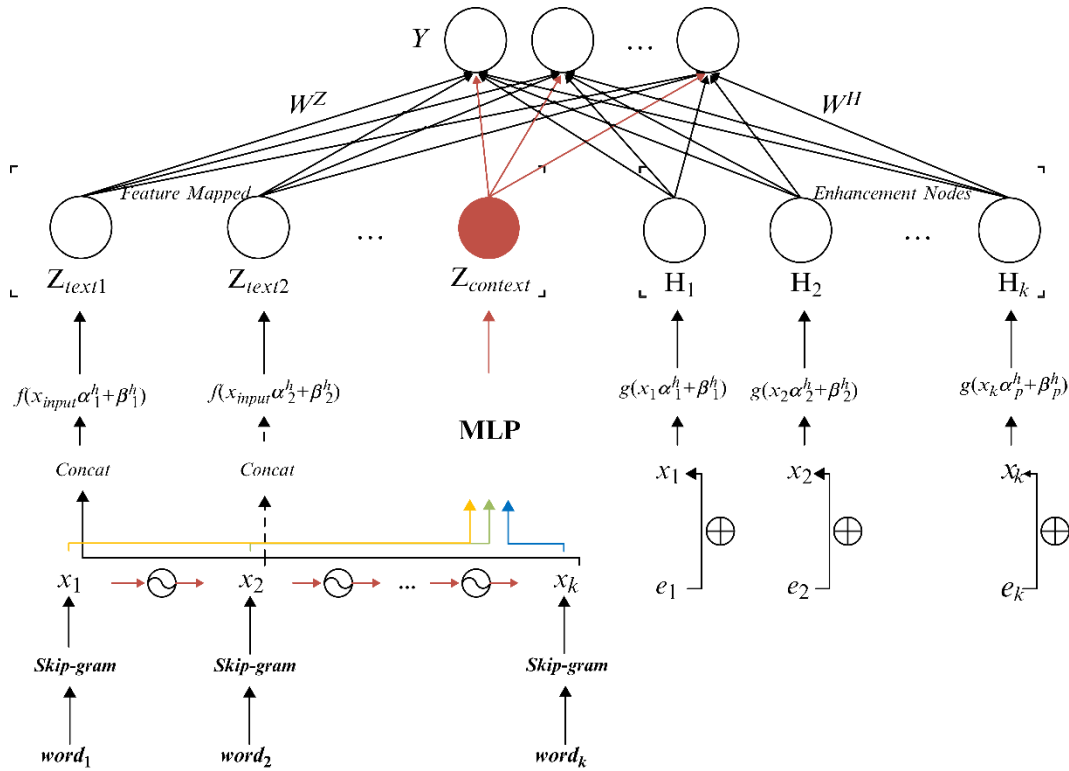


FIGURE 3. Enhanced Word-unit BLS with attention and sememes.

provides the core data and a convenient data access API for HowNet.

II. METHODOLOGY

In this section, 1) we propose a novel framework—Word-unit BLS—which enables simultaneous learning of both sequence information and word importance; 2) we incorporate the MLP model with attention aggregation to capture the latent relationship between each word and its contextual information in global views; 3) we propose a novel approach for improving the word representation by adopting sememes. An overview of the structure of the enhanced BLS with sememes is shown in Fig. 3.

A. WORD-UNIT BLS FRAMEWORK

Inspired by [22], we recognise that NLP tasks typically involve both sequence and word information. In this study, we considered that sentence sequence information encompasses the original text sequence as well as its contextual information. In the original BLS shown in Fig. 1, the feature-mapped layer is employed to extract the sentence features (a.k.a learning the sequence information), and the enhancement node layer to learn the word information (a.k.a learning the word importance).

1) LEARNING SEQUENCE INFORMATION

To enhance the learning of the sentence information, we reorganise the feature-mapped layer of the original BLS structure by utilising a contextual matrix in the MLP model with

attention aggregation instead of randomly generated matrices. The learning of the sequence information is then divided into two parts, text sequence and context sequence. These parts are represented as Z_{text} and $Z_{context}$, respectively, which are matrices that combine the feature vectors of the sequence information.

a: TEXT SEQUENCE

Consider K training data $\{x^i, y^i\}, i = 1 \dots K$, in the dataset X , where $x^i = (x_1^i, x_2^i, \dots, x_n^i)$ denotes the training data and n is the number of words, and $y^i = (y_1^i, y_2^i, \dots, y_m^i)$ denotes the target label, where m indicates the number of classes. In the original BLS, each mapped feature matrix Z_p is required to be independent of all other mapped features. Under this independence, each mapped feature Z_p can be learned using a random set of weights α_p^z and bias β_p^z over the entire input matrix X , whereas the sequence information of the text data cannot be learned [22]. We focus on the representation vector of each word in the stage of learning the text sequence. Therefore, each x^i , which contains N attributes, is split into N word elements, and each of them, $x_p^i \in x^i, p = 1 \dots N$, is converted into a feature vector under Skip-gram [22]. Then, we assign N words x_p^i to an entire feature vector $x^i \in \mathbb{R}^{N \times h_D}$, which contains the entire information of the sentence, where h_D is the dimension of the hidden nodes. The input matrix X is represented as

$$X = [x^1, x^2, \dots, x^k]. \tag{9}$$

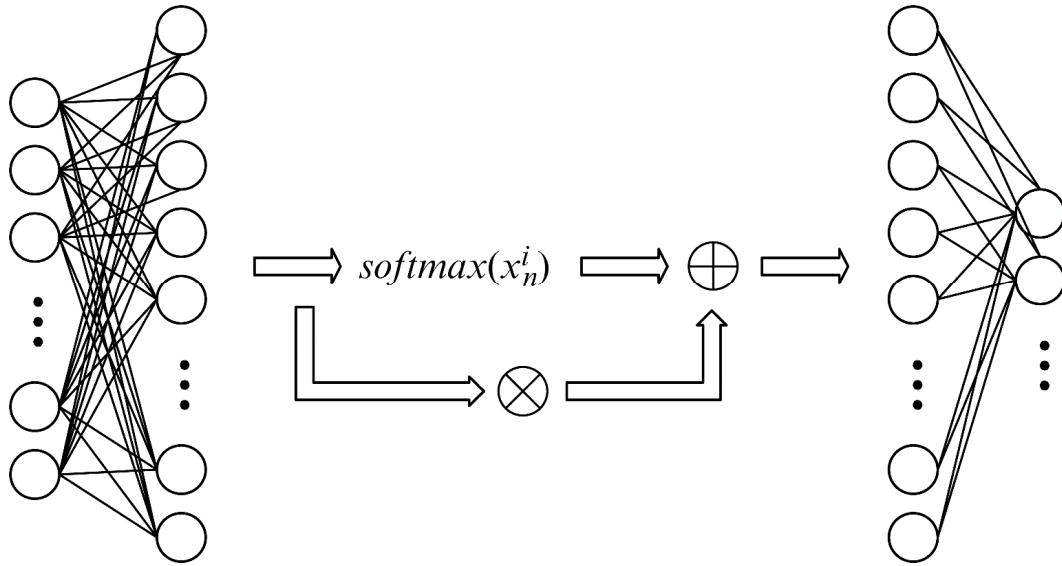


FIGURE 4. Structure of multi-layer perceptron with attention mechanism.

Moreover, the dimension of X is represented as $K \times N$. X_k is reflected in the Z_{textk} matrix inherited from the original BLS according to (10).

$$Z_{textk} = f(X_k \alpha_k^z + \beta_k^z), \quad (10)$$

where $f(\cdot)$ is the sigmoid activation function [19]. Both the weights α_k^h and bias β_k^h are randomly generated as well. In addition, the dimension of the text sequence matrix Z_{text} is the same as that of the input matrix X_{input} , which is $K \times N$.

b: CONTEXT SEQUENCE

In this section, we describe the improvement of the original BLS by incorporating a node $Z_{context}$ in the *feature-mapped* layer. This node is a matrix that captures the latent contextual information required for learning the context sequence to help the BLS to understand the relationship between the text and context information without a recurrent structure in the network. Therefore, we constructed a contextual sequence representation model with an attention mechanism to capture the global contextual information in the learning sequence information. The structure of the MLP with attention mechanism is shown in Fig. 4.

Moreover, inspired by the transformer model in [5], we employed position encoding to capture the relative position of each word in the sequence. This enables the model to differentiate between words that share the same embedding but occur at different positions within the sequence. Subsequently, the sentence is input into an MLP that incorporates an attention aggregation mechanism. The model assigns weights to each word in the sequence based on its relevance to the overall context. This capability allows the model to effectively capture the abstract dependencies and relationships among different parts of a sentence, thus generating an interpretable matrix instead of a random one. Importantly,

this matrix is non-iterative during the Word-unit BLS training process, which further reduces the computing resource consumption and enhances the procession of model training and construction.

Given the word embeddings $x_n^{(k)}$ as input, where k represents the k^{th} word and n represents the n^{th} sentence in the entire dataset, which are generated using Skip-gram, the embeddings are processed using position encoding, as shown in (11) and (12).

$$Position_encoding_{(position,2i)} = \sin\left(\frac{position}{10000^{\frac{2i}{d_{model}}}}\right), \quad (11)$$

$$Position_encoding_{(position,2i+1)} = \cos\left(\frac{position}{10000^{\frac{2i}{d_{model}}}}\right), \quad (12)$$

where i represents the i^{th} value in the embedding, and d_{model} represents the dimension for positional encoding, which is the dimension output from Skip-gram. After the position encoding step, the embeddings are fed into the MLP with attention mechanism. The input embeddings are first processed by a linear layer to reduce their dimensionality, followed by a non-linear activation function, as shown in (13).

$$a^l = \text{sigmoid}(w^l a^{l-1} + b^l), \quad (13)$$

where $\text{sigmoid}(\cdot)$ is the activation function, and a is the output of the l^{th} layer neural network. The attention mechanism then calculates the relevance between the target word and each of the other words in the sequence. The relevance scores are used to weigh the context words before summing them to form a context vector. This context vector is combined with the target word's embedding to form the final output

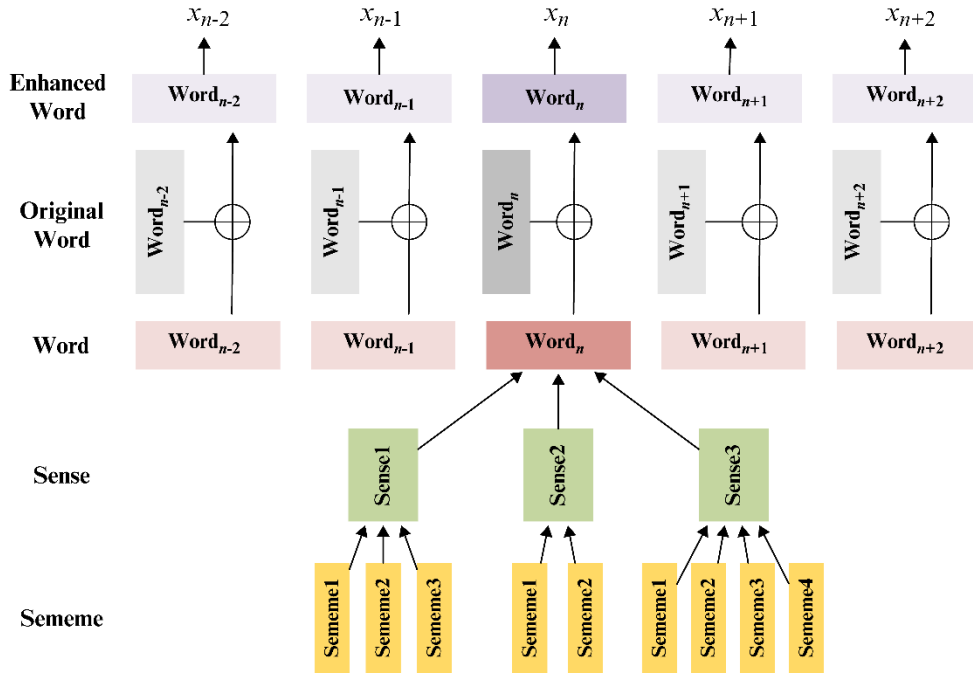


FIGURE 5. Structure of enhanced word importance with sememes.

of the attention mechanism, as shown in (6), (7), and (8) in Section I.B.

To train the MLP, we employed the *MSE* loss function, as shown in (14).

$$\mathcal{L} = \frac{1}{2n} \sum \|\hat{y} - y\|^2, \quad (14)$$

where \hat{y} represents the predicted output, and y represents the ground truth label. To optimise the initialised parameter, the BP algorithm was adopted, which calculates the gradients of the loss function (for calculating the difference between the estimated value and true value for a given instance of data¹) with respect to the model’s parameters, as shown in (15), (16), and (17).

$$\delta^l = \left((w^{l+1})^T \delta^l \right) \odot a^l, \quad (15)$$

$$\frac{d}{dw_j^l} \mathcal{L} = a^{l-1} \delta_j^l, \quad (16)$$

$$\frac{d}{db_j^l} \mathcal{L} = \delta_j^l, \quad (17)$$

where δ^l represents the error term of the l^{th} layer, $(w^{l+1})^T$ is the transpose of the weight of the $(l+1)^{\text{th}}$ layer, and \odot denotes the elementwise multiplication. The deactivation function of a^l is calculated as shown in (18).

$$a^l = \left(w^l a^{l-1} + b^l \right) \times \left(1 - \left(w^l a^{l-1} + b^l \right) \right) \quad (18)$$

¹https://en.wikipedia.org/wiki/Loss_function

The mapped feature node $Z_{context} \in \mathbb{R}^{N \times D}$ is constructed as

$$Z_{context} = [\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N]. \quad (19)$$

The pseudocode of the contextual representation model is presented below.

2) LEARNING WORD IMPORTANCE

In the original BLS, the enhancement nodes are computed based on the mapped features $Z_1 \dots Z_k$. However, as we learn the sequence information within the *feature-mapped* layer, we discover that word importance is also crucial in NLP tasks, as reported in our previous studies [41], [42], where we observed significant performance improvements in understanding short texts with sparsity and polysemy by utilising knowledge-based feature extension methods and feature fusion algorithms, respectively. Hence, in this study, we attempted to incorporate the sememes information from HowNet to improve the representation in word importance learning in BLS. We illustrate our approach in Fig. 5.

The Word-unit BLS takes the word embeddings $X_n \in \mathbb{R}^{N \times D}$ as the input. The matrix X_n is expressed as

$$X_k = [x_1^{(k)}, x_2^{(k)}, \dots, x_N^{(k)}], \quad (20)$$

where N is the number of sentences in the entire dataset, D is the dimension of the word embedding, k represents the k^{th} word in the sentence, and n represents the attribute x_k from the n^{th} sentence. We define the original word as the target word (*Word*, layer one) in HowNet, and the sememe enhancement algorithm looks up the sense and sememe embeddings for the

Multi-Layer Perceptron With Attention Aggregation

Input: The attribution x_n of matrix form X .

Output: The $Z_{context}$ matrix.

Multi-layer Perceptron with Attention Aggregation

1. initialise *hidden_dimension, learning_rate, n_epoch, h₁, b₁, h₂, b₂, attention_weight, context_vector*, and $Z_{context}$.

Position encoding

1. calculate position encoding using eq.11&eq.12.

Training

1. for $epoch = 1, epoch \leq n_epoch$ do
forward pass
2. calculate $h_1, attention_weight_n^k, attention_score_n^k, context_vector, h_2$ using eq.13, eq.6, eq.7, eq.8, and eq.13, respectively.
3. calculate *validated sequence information* utilising LSTM.
4. calculate loss between h_2 and *validated sequence information* using eq.15.
backward pass
5. calculate $d_{h_2}, d_{b_2}, d_{context_vector}, d_{attention_weight}, d_{h_1}, d_{b_1}$
parameter update
6. $h_1, b_1, h_2, b_2, attention_weight, context_vector$ update with gradient descent.
7. end for

Contextual information extraction

1. $h_1, b_1, h_2, b_2, attention_weight,$
contextual information \leftarrow train
 2. initialise *Multi-layer Perceptron with Attention Aggregation* $\leftarrow h_1, b_1, h_2, b_2, attention_weight,$
contextual information
 3. calculate \hat{y} using *Multi-layer Perceptron with Attention Aggregation*
 4. for $sentence = 1, sentence \leq num_sentence$ do
 5. $Z_{context} \leftarrow \hat{y}$
 6. end for
-

target word. The enhancement vector $e_k^{(n)}$, where the description of k and n are the same as that for $x_k^{(n)}$, is calculated using the average aggregation method in *Sense* units, which is a mediator in the aggregation process, as shown in (21). Furthermore, alternative aggregation strategies, *Attention over Context* and *Attention over Target*, were proposed in [43].

$$e_k^{(n)} = \frac{1}{Sense} \sum_{j \in Sense} \sum_{i \in Sememe} sememe_i^{(j)}, \quad (21)$$

where i represents the i^{th} sememe in the set of *Sememe* searched for the target word, j represents the j^{th} sense in the set of *Sense* searched for the target word. Finally, the enhanced word $x_k^{(n)}$ is formed by combining the original word vector $x_k^{(n)}$ and enhancement vector $e_k^{(n)}$, as shown in (22), where the sememe information is utilised to construct the latent

correlations between the words.

$$x_k^{(n)} = x_k^{(n)} \oplus e_k^{(n)} \quad (22)$$

The enhancement node $H_k \in \mathbb{R}^{K \times D}$ is concatenated with K numbers x_k . In Word-unit BLS, the importance of each word is captured by the weight W_h , which connects the enhancement nodes H_k to the output nodes Y . Each k^{th} word matrix in the entire dataset X_k can be converted into the *enhancement node* H_k using (23).

$$H_k = g \left(X_k \alpha_k^h + \beta_k^h \right), \quad (23)$$

where g is a tanh (\bullet) activation function, and $\alpha_k^h \in \mathbb{R}^{l_z \times l_h}$ and $\beta_k^h \in \mathbb{R}^{N \times l_h}$ are randomly generated weights and biases, as in the original BLS. Hence, H_k also has the dimension of $K \times N$.

The pseudocode of the sememe enhancement algorithm is presented below.

Sememe Enhancement

Input: The *words* and *word embedding* of each word.

Output: The *sememe enhancement* matrix.

Sememe enhancement

1. initialise *sense and sememe*.

Look up sense and sememe embeddings

1. for $word = 1, word \leq Sentence$ do
2. for $sense = 1, sense \leq word$ do
3. look up $sence_i^{word}$
4. $sense \leftarrow sence_i^{word}$
5. for $sememe = 1, sememe \leq sense$ do
6. look up $sememe_i^{sense}$
7. $sememe \leftarrow sememe_i^{sense}$
8. end for
9. end for
10. end for

Sememe enhancement

1. for $k = 1, k \leq maxSentence$ do
 2. calculate e_k using eq.21
 3. enhance x_k with sememe aggravation e_k using eq.23
 4. $H_k \leftarrow x_k$
 5. end for
-

3) CONNECTING WEIGHTS

Following [19], W^Z and W^H respectively represent the sequence information and word importance in natural language tasks. The connecting weight W is then the concatenation of W^Z and W^H , where $W = [W^Z | W^H]$. As in the original BLS, W^Z and W^H are not computed separately, but $W = [W^Z | W^H]$ is directly calculated as the pseudoinverse of $[Z_{text1}, Z_{text2}, \dots, Z_{textn} | Z_{context} | H_1, \dots, H_k]$ using ridge regression approximation as in [19]. This allows the sequence information and word importance to be learned simultaneously, resulting in a more effective determination of the final classification result. The connecting weight W is obtained as

shown in (24).

$$W = A^+Y, A^+ = \left(A^T A\right)^{-1} A^T, \quad (24)$$

where

$$A = [Z_{text1}, Z_{text2}, \dots, Z_{textn}, Z_{context} | H_1, H_2, \dots, H_k]. \quad (25)$$

The connecting weights can be calculated by substituting (24) into (25). The pseudocode for the general BLS scheme is illustrated below.

Word-unit BLS Framework

Input: The raw data of natural language.

Output: The connecting weight W .

learning sequence information

Representation of text sequence

1. initialise X and Z_{text}
2. randomly generate the weights α_p^z and bias β_p^z
3. for $k = 1, k \leq K$ do
4. $x^i \leftarrow \text{Skip_gram}(x_i^k)$
5. End
6. $Z_{text} \leftarrow eq.10$

Representation of context sequence

1. initialise $Z_{context}$
2. do *Multi-layer Perceptron with Attention Aggregation*

learning word importance

1. initialise X_p and H_p
2. randomly generate the weights α_p^z and bias β_p^z
3. do *Sememe Enhancement*
4. for $p = 1, p \leq P$ do
5. $H_p = g(X_p \alpha_p^h + \beta_p^h)$
6. End

Calculate connecting weight W

$$W = \left([Z_{text1}, Z_{textn}, Z_{context} | H_1, \dots]^T \cdot [Z_{text1}, Z_{textn}, Z_{context} | H_1, \dots]^T \right)^{-1} \cdot [Z_{text1}, Z_{textn}, Z_{context} | H_1, \dots]^T Y$$

III. EXPERIMENTS

This section outlines the experiments conducted in our study. We used three main datasets in the experiment, as explained later, and evaluated the performance of Word-unit BLS by comparing it with those of Naive Bayes, Support Vector Machine (SVM), and LSTM. Furthermore, we performed a sensitivity analysis and an ablation study to gain additional insights into the performance of the model.

A. EXPERIMENTAL SETUPS

To represent the words in the *zhwiki* (version. 20230301) dataset in the vector space, we followed these steps:

1) We utilized the regular expressions library (re) to remove punctuation. In the Chinese datasets (using OpenCC,² jieba³), we converted the text to Chinese Simplified using OpenCC and jieba. Additionally, words with a *mincount* < 3 were replaced with < unk >. 2) For the implementation of Skip-gram, we used gensim.⁴ We manually configured the *mincount* to 3, the training *epoch* to 300, the embedding *size* to 50, 100, and 200. The remaining parameters in Skip-gram were set to their default values.

We designed different MLP architectures to accommodate various embedding sizes. For each embedding dimension, we designed the network architecture as presented in Table 1. During the training process of the MLP, the stochastic gradient descent strategy was employed with 0.01 learning rate and 300 training epochs as usual [36]. Additionally, we generated the sequence (the values) validated using LSTM, which is one of the most popular models in sequence learning. In this study, the LSTM was configured with a single layer consisting of 200 hidden nodes. We used a learning rate of 0.01 and trained the model for 300 epochs, as described in [22].

TABLE 1. Multi-layer perceptron architecture design.

Network	Embedding Dimension	Architecture
Multi-Layer	50	[50, 16][16, Classes]
Perceptron	100	[100, 32][32, Classes]
	200	[200, 64][64, Classes]

The performance was evaluated on four datasets pertaining to the real world in text classification tasks by using the metrics of accuracy and training time. Except the comparison experiments involving the learning sequence information, which had an embedding dimension of 50, all other experimental results were obtained using an embedding size of 100. All the experiments were conducted using Python (version. 3.10.0) on a device equipped with an AMD Ryzen 7 5800H CPU with 16 GB RAM.

B. DATASETS

In the experiments, the following four real-world datasets were used to evaluate the performance of our model: *THUCNews*, *Chinese_news*, *ChnSentiCorp_h1l*, and *apple-twitter-sentiment*. The *THUCNews*⁵ and *Chinese_news*⁶ datasets were used to evaluate the performance of Word-unit BLS for different text lengths and data scales in the topic categorisation task, respectively. In addition, the *ChnSentiCorp_h1l* and *apple-twitter-sentiment* datasets were applied to evaluate the performance of our framework in the sentimental analysis task. Specifically, the *apple-twitter-sentiment* dataset was employed to compare the performance of learning the sequence information between the model in [22] and our

²<https://github.com/BYVoid/OpenCC>

³<https://github.com/fxsjy/jieba>

⁴<https://radimrehurek.com/gensim/models/word2vec.htm>

⁵Text length: Long; Data scale: Large

⁶Text length: Short; Data scale: Small

TABLE 2. Datasets.

Dataset	Number of Classes	Samples
<i>THUCNews</i>	10	65,000
<i>Chinese_news</i>	3	20,738
<i>ChnSentiCorp_hlt</i>	2	7,766
<i>apple-twitter-sentiment</i>	3	1,631

model. A brief overview of these datasets can be found in Table 2. To create training, validation, and testing sets, we allocated the data manually using a ratio of 7:1:2.

THUCNews is a large-scale Chinese news dataset comprising data collected from various websites and covering 10 distinct categories including #Sports, #Finance, #Real Estate, #Home Furnishing, #Education, #Technology, #Fashion, #Current Affairs, #Games, and #Entertainment. The dataset comprises 65,000 samples, with an average of 6,500 samples per category. For each category, 5,000 assigned to training, 500 to validation, and 1,000 to testing.

Chinese_news is a large-scale dataset comprising Chinese news sourced from China Central Television. It is divided into three categories: #long news with 11,534 samples, #domestic short news with 6,186 samples, and #international short news with 3,018 samples.

ChnSentiCorp_hlt is a small-scale dataset designed for sentiment analysis and focuses on hotel reviews. The dataset classifies the comments into two categories—#Positive with 5,322 samples and #Negative with 2,444 samples.

apple-twitter-sentiment is a small-scale dataset for sentiment analysis, comprising tweets related to Apple. It categorises the comments into three groups: #Positive with 686 samples, #Neutral with 801 samples, and #Negative with 143 samples.

C. EXPERIMENTAL RESULTS AND ANALYSIS

1) COMPARISON RESULTS

The accuracy (accuracy \pm standard deviation) and training time of the various approaches are shown for each dataset in Table 3. The results reveal that Word-unit BLS surpassed the baseline by margins of 8.26%, 4.25%, and 7.79% for the *THUCNews*, *Chinese_news*, and *ChnSentiCorp_hlt* datasets, respectively, in terms of accuracy. This superior performance is attributed to the combined use of sequence information learning and word importance learning. Firstly, the extraction of contextual information by an MLP, coupled with attention aggregation after position encoding, enhances the performance of sequence information learning in the *feature-mapped* layer. Then, the sememe enhancement proposed for learning the word importance at the *enhancement node* layer for word representation, facilitated by the incorporation of a linguistic lexicon, further boosts the effectiveness of the proposed approach.

Furthermore, Word-unit BLS demonstrated significantly shorter training time, specifically 1/33, 1/6, and 1/6 of the time required for *baseline* (Naive Bayes) with the

TABLE 3. Experimental results.

Dataset	Method	Accuracy (%)	Improvement (%)	Training Time (s)	Time Ratio
<i>THUCNews</i>	Naive Bayes	88.06 \pm 0.68	<i>baseline</i>	926.44	<i>baseline</i>
	SVM	88.74 \pm 2.76	0.68	4,012.82	\
	LSTM	90.06 \pm 1.37	2.00	4,217.29	\
	our	96.32\pm1.53	8.26*	28.74	1/33*
<i>Chinese_news</i>	Naive Bayes	88.41 \pm 0.73	<i>baseline</i>	70.72	<i>baseline</i>
	SVM	87.90 \pm 0.85	-0.51	484.75	\
	LSTM	89.13 \pm 1.94	0.72	820.28	\
	our	92.66\pm1.62	4.25	12.18	1/6
<i>ChnSentiCorp_hlt</i>	Naive Bayes	82.94 \pm 0.54	<i>baseline</i>	18.74	<i>baseline</i>
	SVM	86.12 \pm 0.82	3.18	237.10	\
	LSTM	86.81 \pm 1.51	3.87	400.63	\
	our	90.53\pm0.97	7.59	3.07	1/6

THUCNews, *Chinese_news*, and *ChnSentiCorp_hlt* datasets, respectively. The primary reason for this reduction in training time is the computation of the connecting weights using the pseudoinverse and ridge regression methods, as shown in (24) and (25) [19], [22], which is also performed for the original BLS. The position-add employed in sememe enhancement also contributes to the reduced computational demand, which maintains the same dimensionality without extending the training time. Furthermore, the incorporation of contextual information nodes based on an MLP does not appreciably increase the training duration. Therefore, Word-unit BLS exhibits superior performance when compared with the conventional machine learning methodologies in terms of both effectiveness and efficiency, particularly for large-scale datasets.

2) LEARNING SEQUENCE INFORMATION

To evaluate the performance of Word-unit BLS in learning the sequence information, we conducted comparative and sensitivity experiments on the *apple-twitter-sentiment* dataset. Firstly, we compared the performance of Word-unit BLS with those of R-BLS and G-BLS, which are considered the most efficient models for sequence information learning in NLP tasks based on BLS. In this experiment, we used an embedding dimension of 50 as in [22], and selected LSTM as the baseline model, which is considered the most efficient and popular method in sequence learning. The experimental results shown in Table 4 indicate that Word-unit

TABLE 4. Experimental results of learning sequence information.

Dataset	Method	Accuracy	Improvement
<i>apple-twitter-sentiment</i>	LSTM	66.81 \pm 0.73 [22]	\
	R-BLS	67.34 \pm 1.00 [22]	0.53
	G-BLS	67.76 \pm 0.67 [22]	0.95
	our	72.84\pm1.74	6.03

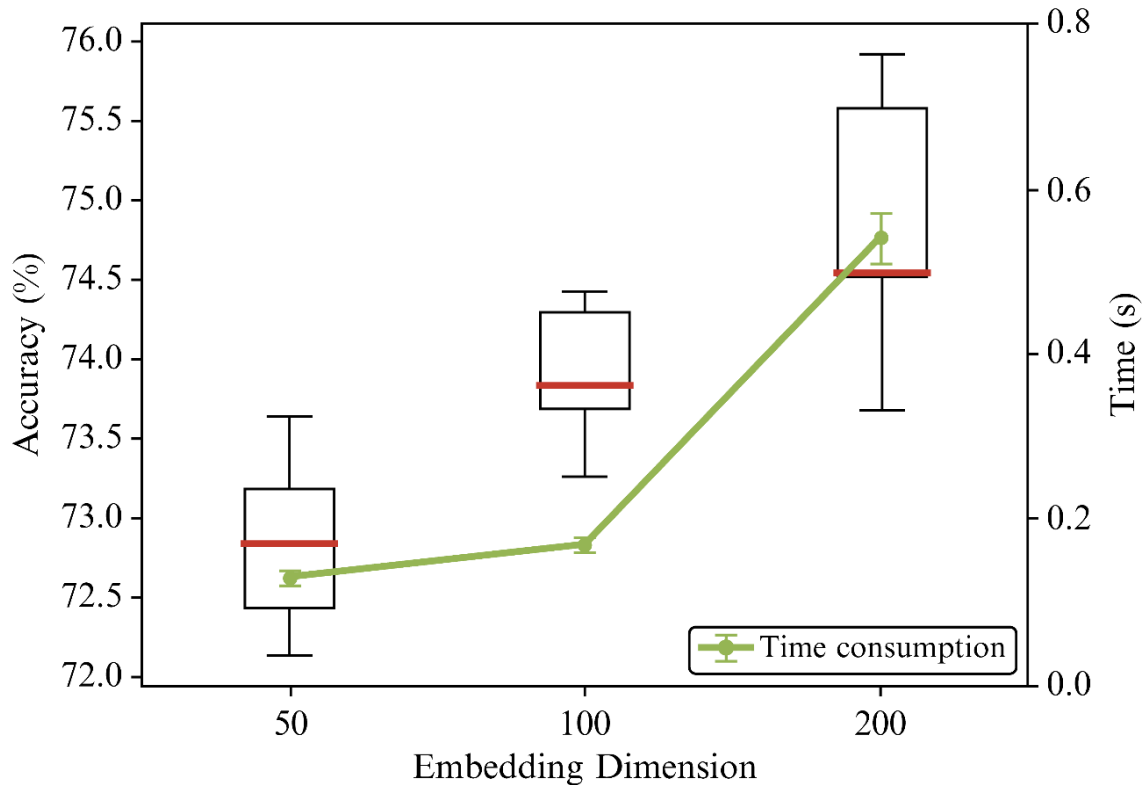


FIGURE 6. Accuracy and time consumption for different embedding dimensions.

BLS achieved an average accuracy of 72.84% on the *apple-twitter-sentiment* dataset, which is a 6.03% improvement over that of LSTM. Furthermore, Word-unit BLS outperformed R-BLS and G-BLS by 5.50% and 5.08%, respectively. These results suggest that Word-unit BLS has excellent performance in learning the sequence information, and the proposed method based on sequence extraction and contextual information is an effective BLS at the *feature-mapped* layer.

Moreover, we designed sensitivity experiments to evaluate the effect of varying the embedding dimension on the test accuracy and training time. Specifically, the embedding dimension was set to 50, 100, and 200. The experimental results are illustrated in Fig. 6. For the embedding dimensions of 50, 100, and 200, the Word-unit BLS model achieved average accuracies of 72.84%, 73.90%, and 74.84%, with training times of 0.13 s, 0.17 s, and 0.54 s, respectively. Increasing the embedding dimension improved the accuracy but extended the training time. Obviously, both accuracy and training time tended to increase with the embedding dimension. The growth in accuracy appeared to be linear, and the results displayed stability at the embedding dimension of 100 when compared with those at the other dimensions. Simultaneously, the training time exhibited only a minor increase when the embedding dimension transitioned from 50 to 100, followed by a significant increase thereafter.

3) LEARNING WORD IMPORTANCE

To demonstrate the influence of sememe enhancement on learning the word importance, we conducted an ablation study on three datasets—*THUCNews*, *Chinese_news*, and *ChnSentiCorp_htl*. The experimental results are shown in Fig. 7. The corresponding accuracies with sememe enhancement were 96.32%, 92.66%, and 90.53% in testing, which indicated improvements of 2.68%, 3.74%, and 1.27%, respectively, when compared with the accuracy obtained with the original model (non-sememe enhancement: 93.64%, 88.92%, 89.26%, respectively). It is clear that sememe enhancement significantly impacts text classification across all datasets, particularly those of a large scale. However, sememe enhancement yielded 1.27% improvement for *ChnSentiCorp_htl*. This result can be explained by information redundancy [13], which occurs due to the augmented sememe information in word representation. As a result, the improvement for a relatively smaller dataset was not significant.

Furthermore, we present the relationship between information gain and construction time of Word-unit BLS in Fig. 8, providing valuable insights for incorporating sememe enhancement in downstream tasks. To analyze this relationship, we manually segmented each sentence in the *THUCNews* dataset into sentences comprising 5, 10, 30, and 100 words. Subsequently, we measured the information gain and time consumption for each scale. The experimental results are shown in Fig. 8. The delta gains of the information

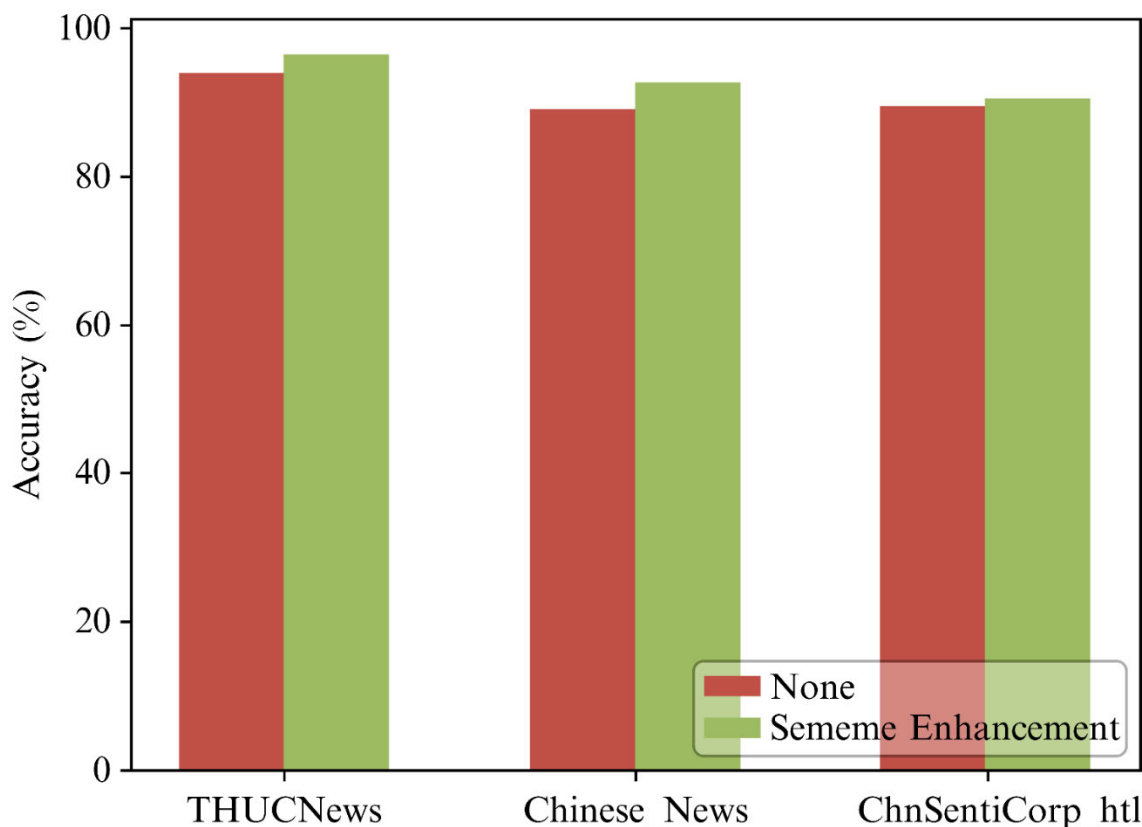


FIGURE 7. Results of ablation study of sememe enhancement.

gain were 1.59%, 2.30%, 1.67%, and 1.38%, respectively, indicating a process of rapid increase followed by a subsequent decline. In other words, the performance under sememe enhancement exhibits a sharp rise before gradually stabilising over time. Simultaneously, time consumption increases with the expansion of the data scale. Therefore, a balance between performance and time consumption must be attained in downstream tasks. This result further elucidates why sememe enhancement proves more effective on larger datasets.

IV. DISCUSSION

In this study, we successfully constructed a novel framework, Word-unit Broad Learning System, *a.k.a* Word-unit BLS, which is capable of understanding natural language based on the theory that a language is composed of sequence and word information. The framework aims to enhance the representation of both sequence and word information in NLP tasks. In this study, we applied this framework to various NLP tasks.

A. KEY FINDING

The key contribution of Word-unit BLS lies in its ability to improve the representation of sequence information by combining text information with context information. This integration enables Word-unit BLS to capture more detailed information from the input data, leading to enhanced accuracy and reduced computing time for sequence understanding in the *feature-mapped* layer of the BLS. Additionally,

Word-unit BLS enhances word representation performance by leveraging sememes in the *enhancement node* layer. Sememes are obtained from HowNet, an external language knowledge base that encompasses concepts and relationships in Chinese. By incorporating sememes into the word representation process, Word-unit BLS can capture more fine-grained semantic information, thus offering potential benefits across various NLP tasks.

Our experiments demonstrated the performance of the proposed framework in terms of effectiveness and efficiency. The training time of the proposed framework was 1/33 of that of Naive Bayes (*baseline*) whereas the accuracy was improved by 8.26% for the *THUCNews* dataset. Furthermore, we explored the performance of sequence information learning and word importance learning by conducting ablation studies that considered various factors that may affect the performance. Although we merely evaluated the performance of Word-unit BLS for text classification and sentiment analysis tasks, the methodology can be applied to various NLP tasks, e.g. machine translation and information extraction [44].

B. LIMITATIONS

The limitations of this study include the reliance on HowNet to extract sememes for enhancing the word representation in the *enhancement node* layer. This reliance raises concerns about the availability, quality, and coverage of the knowledge

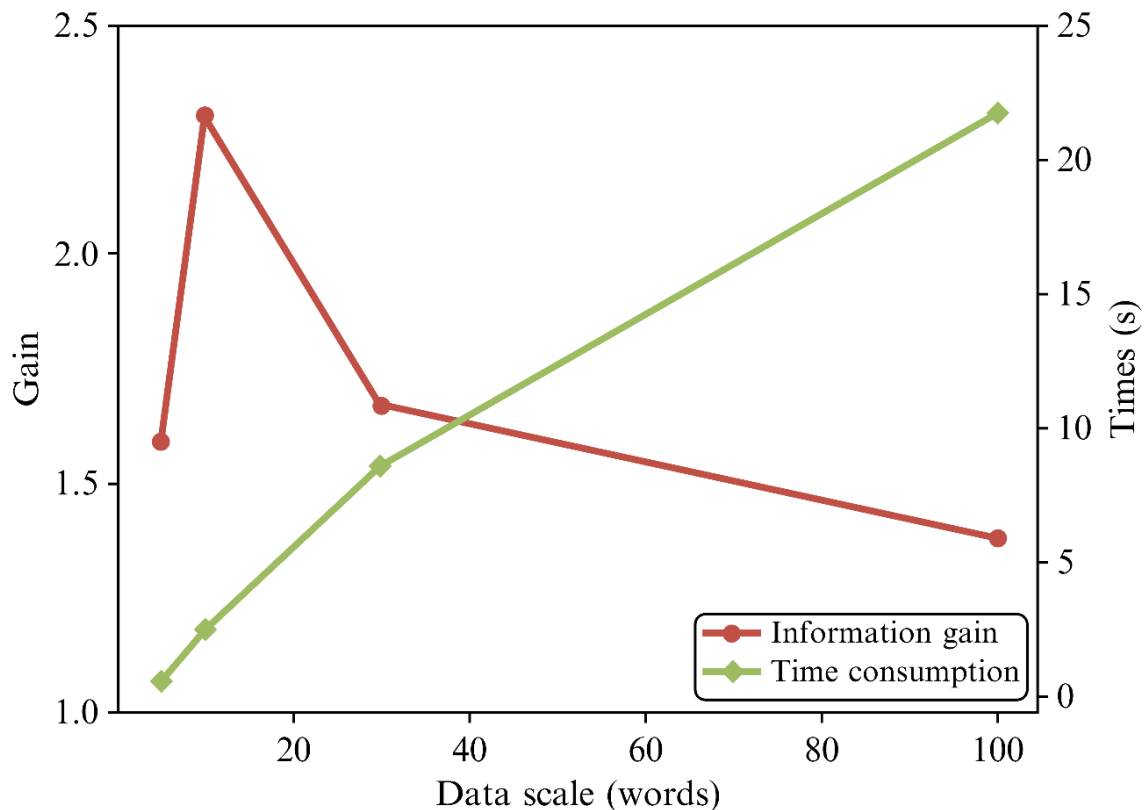


FIGURE 8. Information gain and time consumption with sememe enhancement.

base. By excluding the influence of the external knowledge language base, the analysis becomes incomplete or biased, potentially affecting the evaluation of the performance and reliability of Word-unit BLS. Additionally, the limited scope of the language knowledge base restricts its performance and practical applicability. However, manual construction of external knowledge bases is challenging due to the potential consumption of excessive resources. To address these issues, we suggest integrating multiple external language knowledge bases to expand the scale of data and translating the existing datasets into other languages to incorporate additional linguistic contexts beyond Chinese.

Moreover, we did not incorporate multi-features such as phonetics information (another type of sequence information) [41] or graph technology to extract latent entity relationships containing auxiliary knowledge [45]. Moreover, a foreseeable challenge in integrating multi-features into Word-unit BLS is the substantial increase in training time. One potential solution to this challenge is the Zhu method [46], which reduces computational complexity by utilizing the inverse Cholesky factor of the Hermitian matrix during pseudoinverse computation and factor updates. Despite the limitations of the study, this paper introduced a novel framework that integrates BLS with linguistic theory. However, additional research is needed to explore language understanding through the combination of multi-features.

V. CONCLUSION

In this paper, we proposed the Word-unit BLS framework for text classification and sentiment analysis, enabling simultaneous learning of multiple types of information at the word level, including sequential information and word importance. The main contributions of this study are as follows: 1) Achieving higher accuracy and shorter training time with the Word-unit BLS framework by enabling simultaneous learning of sequence information and word importance. 2) Introducing a non-iterative context matrix constructed using an MLP with an attention mechanism into the BLS. 3) Employing a novel approach to improve word representation by incorporating sememe information while learning word importance. We conducted several experiments that demonstrated the superior performance of Word-unit BLS compared with conventional machine learning methodologies in terms of the accuracy and training time when compared. Furthermore, when compared with R-BLS and G-BLS (the SOTA methods in learning sequential information based on BLS), Word-unit BLS exhibited superior sequence learning performance without increasing the training time. However, when applying sememe enhancement to small datasets in downstream tasks, there is a possibility of information redundancy.

Overall, Word-unit BLS demonstrates excellent performance in terms of both effectiveness and efficiency.

Experimental results demonstrated its potential in various NLP tasks, showcasing its excellent performance in natural language understanding. While the sememe enhancement algorithm significantly improves accuracy, it does lead to high computational time for large-scale data, particularly long sentences. In future research, further investigation will be conducted to reduce the construction time of the algorithm.

COMPETING INTERESTS

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

AUTHOR CONTRIBUTIONS

Yuchao Jiang: Methodology, Software, Validation, Funding acquisition, Writing—original draft, Writing—review and editing; Jiadong Lu: Validation; Tongfei Li: Validation; and Wei Lv: Funding acquisition, Project administration. All authors have read and agreed to the published version of the manuscript.

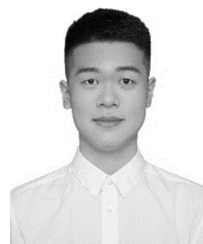
DATA AVAILABILITY STATEMENT

The datasets used in this study are available at the following locations: *THUCNews* dataset: <http://thuctc.thunlp.org/>; *Chinese news* dataset: <https://www.kaggle.com/datasets/noxmoon/chinese-official-daily-news-since-2016>; *ChnSentiCorp_htl* dataset: https://github.com/SophonPlus/ChineseNlpCorpus/blob/master/datasets/ChnSentiCorp_htl_all/ChnSentiCorp_htl_all.csv; *apple-twitter-sentiment dataset*: <https://www.kaggle.com/datasets/seriousran/appletwitter-sentiment-texts>.

REFERENCES

- [1] O. Khattab and M. Zaharia, "ColBERT: Efficient and effective passage search via contextualized late interaction over BERT," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2020, pp. 39–48, doi: [10.1145/3397271.3401075](https://doi.org/10.1145/3397271.3401075).
- [2] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *J. Mach. Learn. Res.*, vol. 21, no. 1, pp. 5485–5551, 2020.
- [3] M. Labonne and S. Moran, "Spam-T5: Benchmarking large language models for few-shot email spam detection," 2023, *arXiv:2304.01238*.
- [4] Y. Zhang and B. Wallace, "A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification," 2015, *arXiv:1510.03820*.
- [5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.
- [6] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 115–124, doi: [10.1145/3077136.3080834](https://doi.org/10.1145/3077136.3080834).
- [7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.
- [8] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543, doi: [10.3115/v1/D14-1162](https://doi.org/10.3115/v1/D14-1162).
- [9] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov, "FastText.Zip: Compressing text classification models," 2016, *arXiv:1612.03651*.
- [10] Z. Dong and Q. Dong, "HowNet—A hybrid language and knowledge resource," in *Proc. Int. Conf. Natural Lang. Process. Knowl. Eng.*, Oct. 2003, pp. 820–824.
- [11] J. Mei, Y. Zhu, Y. Gao, and H. Yin, *Tongyici Cilin (Dictionary of Synonymous Words)*. Shanghai, China: Shanghai Cishu Publisher, 1983.
- [12] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, 1995.
- [13] K. N. Singh, S. D. Devi, H. M. Devi, and A. K. Mahanta, "A novel approach for dimension reduction using word embedding: An enhanced text classification approach," *Int. J. Inf. Manage. Data Insights*, vol. 2, no. 1, Apr. 2022, Art. no. 100061, doi: [10.1016/j.jjime.2022.100061](https://doi.org/10.1016/j.jjime.2022.100061).
- [14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*.
- [15] T. B. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1877–1901.
- [16] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter," 2019, *arXiv:1910.01108*.
- [17] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A robustly optimized BERT pretraining approach," 2019, *arXiv:1907.11692*.
- [18] T. Hoefler, D. Alistarh, T. Ben-Nun, N. Dryden, and A. Peste, "Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks," *J. Mach. Learn. Res.*, vol. 22, no. 1, pp. 1–127, 2021.
- [19] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018, doi: [10.1109/TNNLS.2017.2716952](https://doi.org/10.1109/TNNLS.2017.2716952).
- [20] X. Gong, T. Zhang, C. L. P. Chen, and Z. Liu, "Research review for broad learning system: Algorithms, theory, and applications," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8922–8950, Sep. 2022, doi: [10.1109/TCYB.2021.3061094](https://doi.org/10.1109/TCYB.2021.3061094).
- [21] S. Feng and C. L. P. Chen, "Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 414–424, Feb. 2020, doi: [10.1109/TCYB.2018.2857815](https://doi.org/10.1109/TCYB.2018.2857815).
- [22] J. Du, C.-M. Vong, and C. L. P. Chen, "Novel efficient RNN and LSTM-like architectures: Recurrent and gated broad learning systems and their applications for text classification," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1586–1597, Mar. 2021, doi: [10.1109/TCYB.2020.2969705](https://doi.org/10.1109/TCYB.2020.2969705).
- [23] X. Hu, X. Wei, Y. Gao, H. Liu, and L. Zhu, "Variational expectation maximization attention broad learning systems," *Inf. Sci.*, vol. 608, pp. 597–612, Aug. 2022, doi: [10.1016/j.ins.2022.06.074](https://doi.org/10.1016/j.ins.2022.06.074).
- [24] H. Xia, J. Tang, W. Yu, and J. Qiao, "Tree broad learning system for small data modeling," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Nov. 3, 2022, doi: [10.1109/TNNLS.2022.3216788](https://doi.org/10.1109/TNNLS.2022.3216788).
- [25] L. Su, L. Xiong, and J. Yang, "Multi-Attn BLS: Multi-head attention mechanism with broad learning system for chaotic time series prediction," *Appl. Soft Comput.*, vol. 132, Jan. 2023, Art. no. 109831, doi: [10.1016/j.asoc.2022.109831](https://doi.org/10.1016/j.asoc.2022.109831).
- [26] L. Liu, L. Cai, T. Xie, and Y. Wang, "Self-paced broad learning system," *IEEE Trans. Cybern.*, vol. 53, no. 6, pp. 4029–4042, Jun. 2022, doi: [10.1109/TCYB.2022.3181449](https://doi.org/10.1109/TCYB.2022.3181449).
- [27] L. Liu, T. Liu, C. L. P. Chen, and Y. Wang, "Modal-regression-based broad learning system for robust regression and classification," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 22, 2023, doi: [10.1109/TNNLS.2023.3256999](https://doi.org/10.1109/TNNLS.2023.3256999).
- [28] H. Ramchoun, M. A. J. Idrissi, Y. Ghanou, and M. Ettaouil, "Multilayer perceptron: Architecture optimization and training," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, no. 1, pp. 26–30, 2016, doi: [10.9781/ijimai.2016.415](https://doi.org/10.9781/ijimai.2016.415).
- [29] C. L. P. Chen and J. Z. Wan, "A rapid learning and dynamic stepwise updating algorithm for flat neural networks and the application to time-series prediction," *IEEE Trans. Syst., Man Cybern., B, Cybern.*, vol. 29, no. 1, pp. 62–72, Feb. 1999, doi: [10.1109/3477.740166](https://doi.org/10.1109/3477.740166).
- [30] C. L. P. Chen, S. R. LeClair, and Y.-H. Pao, "An incremental adaptive implementation of functional-link processing for function approximation, time-series prediction, and system identification," *Neurocomputing*, vol. 18, nos. 1–3, pp. 11–31, Jan. 1998, doi: [10.1016/S0925-2312\(97\)00062-3](https://doi.org/10.1016/S0925-2312(97)00062-3).
- [31] C. L. P. Chen, "A rapid supervised learning neural network for function interpolation and approximation," *IEEE Trans. Neural Netw.*, vol. 7, no. 5, pp. 1220–1230, Sep. 1996, doi: [10.1109/72.536316](https://doi.org/10.1109/72.536316).

- [32] C. L. P. Chen, Z. Liu, and S. Feng, "Universal approximation capability of broad learning system and its structural variations," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 1191–1204, Apr. 2019, doi: [10.1109/TNNLS.2018.2866622](https://doi.org/10.1109/TNNLS.2018.2866622).
- [33] I. Alshubaily, "TextCNN with attention for text classification," 2021, *arXiv:2108.01921*.
- [34] Y. J. Yu, S. J. Yoon, S. Y. Jun, and J. W. Kim, "TABAS: Text augmentation based on attention score for text classification model," *ICT Exp.*, vol. 8, no. 4, pp. 549–554, Dec. 2022, doi: [10.1016/j.ict.2021.11.002](https://doi.org/10.1016/j.ict.2021.11.002).
- [35] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 577–585.
- [36] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.
- [37] X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, and H. Liu, "Expectation-maximization attention networks for semantic segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9166–9175.
- [38] Q. Liu, "Word similarity computing based on HowNet," *Comput. Linguist. Chin. Lang. Process.*, vol. 7, no. 2, pp. 59–76, 2002.
- [39] F. Xianghua, L. Guo, G. Yanyan, and W. Zhiqiang, "Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon," *Knowl.-Based Syst.*, vol. 37, pp. 186–195, Jan. 2013, doi: [10.1016/j.knsys.2012.08.003](https://doi.org/10.1016/j.knsys.2012.08.003).
- [40] F. Qi, C. Yang, Z. Liu, Q. Dong, M. Sun, and Z. Dong, "Open-HowNet: An open sememe-based lexical knowledge base," 2019, *arXiv:1901.09957*.
- [41] Y. Jiang, X. Li, C. Huang, W. Lu, and M. Xu, "A phonetics and semantics-based Chinese short text fusion algorithm," in *Proc. Int. Conf. Comput. Inf. Sci. Cham, Switzerland: Springer*, 2022, pp. 153–166.
- [42] C. Huang, X. Li, Y. Jiang, W. Lv, and M. Xu, "Feature extension for Chinese short text based on Tongyici Cilin," in *Proc. Int. Conf. Comput. Inf. Sci. Cham, Switzerland: Springer*, 2022, pp. 167–180.
- [43] Y. Niu, R. Xie, Z. Liu, and M. Sun, "Improved word representation learning with sememes," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, 2017, pp. 2049–2058.
- [44] D. Khurana, A. Koli, K. Khatter, and S. Singh, "Natural language processing: State of the art, current trends and challenges," *Multimedia Tools Appl.*, vol. 82, no. 3, pp. 3713–3744, Jan. 2023, doi: [10.1007/s11042-022-13428-4](https://doi.org/10.1007/s11042-022-13428-4).
- [45] L. Li, X. Zhang, Y. Ma, C. Gao, J. Wang, Y. Yu, Z. Yuan, and Q. Ma, "A knowledge graph completion model based on contrastive learning and relation enhancement method," *Knowl.-Based Syst.*, vol. 256, Nov. 2022, Art. no. 109889, doi: [10.1016/j.knsys.2022.109889](https://doi.org/10.1016/j.knsys.2022.109889).
- [46] H. Zhu, Z. Liu, C. L. P. Chen, and Y. Liang, "An efficient algorithm for the incremental broad learning system by inverse Cholesky factorization of a partitioned matrix," *IEEE Access*, vol. 9, pp. 19294–19303, 2021, doi: [10.1109/ACCESS.2021.3052102](https://doi.org/10.1109/ACCESS.2021.3052102).



YUCHAO JIANG was born in 2001. He is currently pursuing the B.E. degree in software engineering with the School of Computer Science, Zhuhai College of Science and Technology, Zhuhai, China. He is also a Software Developer Intern with the Information Technology Application Innovation Platform Department, Ygsoft Inc., Zhuhai. His research interests include syntactic analysis, semantic analysis, text representation and its applications to natural language processing, and image retrieval in computer vision. He was a 2023 honour recipient of the Computer Society of Zhuhai Future Stars of Science and Technology.



JIADONG LU was born in Chaozhou, Guangdong, China, in 2002. He is currently pursuing the bachelor's degree in data science with the Zhuhai College of Science and Technology, Zhuhai, China. He is also with the Meteorologic Forecast Project of the Guangdong-Hong Kong-Macao Applied Mathematics Centre, School of Aliyun Big Data Applications, Zhuhai College of Science and Technology. His research interests include data mining, machine learning, ensemble deep learning, natural language processing, and the application of artificial intelligence in meteorology.



TONGFEI LI received the M.S. degree from the Institute of Data Science, City University of Macau, Macau, China, in 2021. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Faculty of Innovation Engineering, Macau University of Science and Technology, Macau. He has lectured with the Huike Group and has additional teaching experience in many universities, mainly on subjects, such as hadoop, machine learning, deep learning, and computer vision.



WEI LV received the B.S., M.S., and Ph.D. degrees from the Mathematics Department, Software Research Institute, Sun Yat-sen University, in 2009. He has been a Visiting Scholar with Princeton University, Nanyang Technological University, The City University of New York, and RWTH Aachen University. He is currently the Dean of the School of Aliyun Big Data Applications, Zhuhai College of Science and Technology, and a Visiting Professor with the Institute of Data Science, City University of Macau. His research interests include big data and cloud computing.

...