



International Journal of Intelligent Information and Database Systems

ISSN online: 1751-5866 - ISSN print: 1751-5858

<https://www.inderscience.com/ijids>

Sentiment classification method of fresh agricultural product reviews based on semantic and emotional optimisation

Yindong Dong, Yu Zhang, Guodong Wu, Xia Chen, Lijing Tu, Guohua Fan

DOI: [10.1504/IJIDS.2025.10068156](https://doi.org/10.1504/IJIDS.2025.10068156)

Article History:

Received:	28 February 2024
Last revised:	04 June 2024
Accepted:	04 June 2024
Published online:	23 December 2024

Sentiment classification method of fresh agricultural product reviews based on semantic and emotional optimisation

Yindong Dong*, Yu Zhang, Guodong Wu,
Xia Chen, Lijing Tu and Guohua Fan

School of Information and Artificial Intelligence,
Anhui Agricultural University,
130 Changjiang West Road, Hefei City,
Anhui Province, 230036, China
Email: dongyindong66@163.com
Email: 2582921962@qq.com
Email: 8978850@qq.com
Email: 787112809@qq.com
Email: tlj@ahau.edu.cn
Email: fghvan@yeah.net

*Corresponding author

Abstract: The complexity of semantics and structure in fresh agricultural product reviews can lead to sparsity in the distribution of text sentiment information. To address this issue, a sentiment classification model (Electra XLNet BIGRU multi-head attention – EXBMA) for fresh agricultural product reviews is proposed, which combines semantic and sentiment information optimisation. Firstly, the Electra model and XLNet model were used to obtain word level and sentence level information of fresh agricultural product reviews, respectively, and the semantic features obtained were fused using (multimodal compact bilinear – MCB) algorithm. Secondly, TextRank is used to extract keywords to construct an emotional key dictionary and combined with a multi-head attention mechanism to enhance emotional attention. Finally, to enhance the contextual representation, the BIGRU is used to learn contextual information to improve the classification performance. The experimental results indicate that the EXBMA can better achieve collaborative optimisation of semantic and emotional information, and performs better than other existing classification models in emotional classification of fresh agricultural product reviews.

Keywords: fresh agricultural products; emotional classification; semantic optimisation; sentiment lexicon; attention mechanism.

Reference to this paper should be made as follows: Dong, Y., Zhang, Y., Wu, G., Chen, X., Tu, L. and Fan, G. (2025) ‘Sentiment classification method of fresh agricultural product reviews based on semantic and emotional optimisation’, *Int. J. Intelligent Information and Database Systems*, Vol. 17, No. 1, pp.32–56.

Biographical notes: Yindong Dong received his PhD in Computer Application Technology from Hefei University of Technology, China in 2021. He works for the School of Information and Artificial Intelligence, Anhui Agricultural University in June 2021. He has participated in or presided over more than ten projects, including the Anhui Provincial Department of Education Key Project,

Anhui Provincial Natural Science Foundation, and National Natural Science Foundation. His research interests are affective computing, natural language processing, and intelligent recognition based on physiological signals.

Yu Zhang is currently pursuing a Master's degree at Anhui Agricultural University. His research focuses on recommendation systems and natural language processing.

Guodong Wu received his Doctorate's in Intelligent Decision and Knowledge Management from the University of Donghua University, China in 2018. He joined in the School of Information and Artificial Intelligence, Anhui Agricultural University of China in July 2023. His research interests are artificial intelligence, and recommendation system.

Xia Chen received her PhD in Computer Science and Technology from Anhui University of China in 2019. She joined in the School of Information and Artificial Intelligence, Anhui Agricultural University of China in 2020. Her research interests are computer vision and image recognition.

Lijing Tu is currently studying computer applications at Anhui Agricultural University as a Doctoral student, mainly focusing on artificial intelligence and machine learning. She leads the General Program of Natural Science Research in Anhui Province and the Youth Fund Program of Anhui Agricultural University.

Guohua Fan received his Master's in Computer Science from the University of Hefei University of Technology, China in 2007. He joined in the School of Information and Artificial Intelligence, Anhui Agricultural University of China in September 2007. His research interests are game design, virtual reality, and image recognition.

1 Introduction

In recent years, with the rapid development of the internet and e-commerce, fresh agricultural products online have also opened a sales channel, the cultivation and production of fresh agricultural products by the seasons, climate, and regional influence is greater, the offline purchase of greater limitations and less optional, but the network of fresh agricultural products on the rich variety of fresh agricultural products and convenient, most consumers choose to buy fresh agricultural products online, the scale of fresh agricultural products reviews has also been The scale of fresh produce reviews has also increased. For producers and fresh produce operators, the sentiment categorisation of fresh produce reviews can help them get feedback and suggestions for product improvement, as well as improve product quality based on the issues raised in negative reviews, and also understand consumer preferences through positive reviews, and develop corresponding marketing strategies to attract more consumers. For consumers, other consumers' reviews of products can increase their awareness of fresh produce and make more informed purchasing decisions. Therefore, how to quickly and accurately emotionally categorise the huge amount of online reviews of fresh produce is a popular research area nowadays.

The dictionary and rule-based method is the main method used in the early research on sentiment analysis of fresh produce reviews, which matches and compares the words in the text and the sentiment dictionary to derive the sentiment tendency in the text, and then classifies the sentiment according to the pre-set rules. The classification speed of this method is faster, but it needs to be premised on a high-quality sentiment dictionary and reasonable design rules, and the manual intervention is large, so the actual classification effect is not ideal. The emergence of machine learning brings new inspiration to the sentiment classification of fresh produce reviews. Traditional machine learning extracts feature information from the text and then uses classification algorithms to make judgments about the sentiment polarity of the text. plain Bayes proposed by Lewis (1998), the k-neighbourhood algorithm proposed by Cover and Hart (1967), and the support vector machine proposed by Cortes and Vapnik (1995) are the three most commonly used machine learning models. Using machine learning this method is improved compared to dictionary-based sentiment analysis methods, but the emergence of new data can lead to poor generalisation ability, and for new text, it needs to be re-trained to adapt to the new data distribution; in addition, this method is also insensitive to the sequential information of the text, and it is difficult to obtain contextually relevant information.

With the wide application of deep learning in various fields, more and more scholars have begun to shift the focus of sentiment analysis of fresh produce reviews to deep learning. Compared with traditional machine learning models, deep learning models have greater advantages in extracting text sentiment features. CNN and RNN are two of the more widely used classes of existing deep learning models for the extraction of emotional features. CNNs (LeCun et al., 1998) can automatically learn the features in the text and capture the relevance of specific words or phrases in a sentence to the sentiment through convolutional and pooling layers for a better understanding of the sentiment expression. RNN (Mikolov et al., 2010) is a recursive neural network structure that captures temporal dependencies in sequential data and can maintain the information from previous time steps and pass it on to the current time step when processing textual data, which helps the model to better understand and analyse the contextual information of the text, and therefore can effectively improve the accuracy of sentiment classification. In the described literature (Yin et al., 2017) it is mentioned that CNN performs well in local learning but is unable to acquire global semantic information and RNN can capture global information about the input sequence but performs poorly in acquiring local key features. To solve these problems, many scholars have combined the advantages of CNN and RNN and proposed a hybrid model that captures both local features and global semantic information, which has shown good results in natural language processing tasks. Zhang et al. (2022) proposed a CNN-LSTM model for Chinese sentiment classification. LSTM (Hochreiter and Schmidhuber, 1997) possesses excellent long-term dependency processing capability compared to traditional RNN and has high flexibility to realise effective processing of long sequences through gating mechanisms and memory units. Guo et al. (2020) proposed a CNN-BiLSTM-based online review sentiment analysis method, which can not only establish temporal relationships but also portray local spatial features like CNN. Although the above model can obtain more complete semantic information, the time cost consumed by the CNN model is high due to the large computational volume of the convolutional layer structure and the recursive structure of the RNN model.

The emergence of pre-trained models represented by BERT (Devlin et al., 2018) has further advanced textual sentiment research in recent years. Pota et al. (2021) utilised BERT to evaluate currently available tweet pre-processing operations as a way to enhance the performance of sentiment analysis. Zhao and Yu (2021) proposed a knowledge-enabled linguistic representation model BERT for aspect-based sentiment analysis. Zhang et al. (2021) proposed an extensive multi-task Transformer network BMT-Net to explore the robustness and contextual representation of high-level information, which effectively improves the effectiveness of sentiment analysis. However, the above models only focus on a single aspect of information features and ignore other factors that may have an impact on sentiment categorisation, thus failing to capture the complex relationships and interactions between different features. In addition, BERT mainly relies on literal semantics and local context to understand texts and performs poorly when confronted with texts with deeper contexts and implicit intentions such as sarcasm, puns, or sentiments relying on a specific cultural context (Wang et al., 2022).

With the in-depth research of many scholars, multi-feature fusion models appeared one after another. Li et al. (2021) proposed a two-way LSTM sentiment classification method based on multi-channel features and self-attention, which combines the word vectors in the input text sentences with lexical feature vectors, positional feature vectors, and dependent feature vectors to enhance the effect of sentiment classification. Gan et al. (2021) proposed a scalable multi-channel convolutional neural network and bi-directional long and short-term memory model CNN-BiLSTM, which simultaneously extracts raw contextual features and multi-scale high-level contextual features. Abdi et al. (2019) feature vectors based on statistical, linguistic, and sentiment knowledge, feature vectors based on sentiment shift rules, and feature vectors based on word embeddings improve the classification accuracy of text-level sentiment analysis. Zhuang et al. (2022) proposed a multi-feature neural network model that considers three aspects: contextual, syntactic, and semantic information, which performs well in sentiment recognition of implicit sentences that do not contain sentiment words or phrases. However, the above methods only use simple splicing for fusion, which directly connects all the features from different sources, which not only leads to a significant increase in the dimension of the feature vector and contains a large amount of redundant information, but also may introduce noise and irrelevant information, which affects the overall effect (Lahat et al., 2015). In addition, simple splicing cannot capture the complex associations between features, such as the interaction between syntactic structure and word emotion, and many emotional expressions are difficult to be accurately identified, and the weights cannot be dynamically adjusted according to the importance of the features, resulting in that the information of the key features may be diluted by a large number of unimportant features, which reduces the model's ability to identify the important emotional features, and it is difficult to realise the effective utilisation and fusion of multiple features.

Some cross-domain sentiment analysis challenges have emerged from the wide range of applications of sentiment analysis in various domains. Most approaches focus on information common to the source and target domains, ignoring information specific to the target domain. To address this problem, Li et al. (2020) proposed a model SAMF-BiLSTM with a self-attention mechanism and multi-channel features. The method models existing linguistic knowledge and sentiment resources in a sentiment analysis task forms different feature channels, and enhances the sentiment information using

self-attention mechanisms. Huang et al. (2021) designed an emotion-enhanced ELSTM to improve the feature learning capability of LSTM networks by utilising EI, and the emotion regulation of their learning system is accomplished by the proposed emotion regulator and emotion estimator. Zhao et al. (2021) proposed the domain-specific sentiment word attention model DSSW-ATT. the model uses the attention mechanism as an encoder based on the extraction of shared and unique information, encodes the sentiment word information as the core and combines the shared sentiment word information with the unique sentiment word information, and achieves a better result on the task of sentiment analysis of the datasets in the two domains of hotels and cell phones. Zhu et al. (2022) proposed FinLexNet, a fine-grained sentiment analysis model for financial texts that combines a financial domain sentiment lexicon and an attention mechanism, based on a financial domain dataset, and used an attention mechanism based on a financial domain sentiment lexicon to obtain important sentiment information for different entities, which effectively improves the accuracy of the sentiment analysis on a corpus of fine-grained sentiment analysis in the financial domain. However, there are fewer studies on the application of these sentiment analysis methods to fresh produce reviews, and there is a lack of targeted methodology and modelling (Misra et al., 2020)[. Users in fresh produce reviews are more subjective, and there are more domain-specific sentiment vocabularies and expressions. These unique vocabularies are not adequately identified and utilised in general-purpose models. As a result, there will be greater difficulty in sentiment categorisation at the semantic and emotional levels constrained by the attributes of the fresh produce types.

To achieve effective sentiment classification of fresh produce reviews in complex semantic environments, a semantic and sentiment joint optimisation of sentiment classification model (EXBMA) for fresh produce reviews is proposed, and the comparison of each classification performance index has a more obvious improvement through ablation experiments. The main contributions of this paper are as follows three aspects:

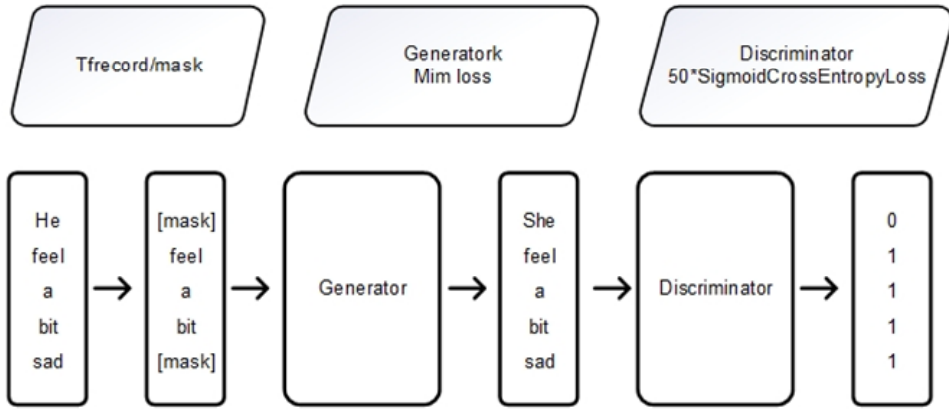
- 1 Fusion of word-level and sentence-level information acquired by Electra and XLNet in terms of structure through multimodal compact bilinear (MCB), effectively avoiding the problems of feature loss and large noise in traditional splicing methods.
- 2 Introducing the multi-attention mechanism to give attention to the sentiment information in the reviews of fresh produce with the help of sentiment key dictionaries to enhance the sentiment expression of the model; adopting BIGRU to obtain the contextual information in the reviews, and improving the performance of sentiment classification through the joint action of semantics, sentiment and structure.
- 3 By crawling fresh produce reviews in Jingdong Fresh Zone to obtain a category-rich dataset, and conducting ablation experiments on this dataset for comparison, the model can achieve better classification performance in semantic information extraction and optimised fusion of sentiment information.

2 Relevant theories and techniques

2.1 Electra algorithm

The Electra model (Clark et al., 2020) is a pre-trained model based on self-supervised learning that has achieved good performance in NLP tasks. The basic idea of the Electra model is to use a classifier to predict whether a word is masked out or not and to train the generator and the discriminator against each other under different input problems. When the generator and discriminator training processes converge, the parameters of the discriminator are used to generate word-level representations. The Electra model learns word-level representations by constructing a masking task, which gives it an obvious advantage over models that only learn contextual representations (e.g., BERT) in extracting word-level textual features, and can better capture the nuances of the words and semantic relationships, which gives it a stronger fine-grained modelling capability. At the same time, it combines generative and discriminative tasks to realise self-supervised learning. As shown in Figure 1.

Figure 1 Structure of Electra model



The generator model selects some input words at random and replaces them with random words or leaves the original words unchanged, and then uses this processed text as input to generate a new text. The text contains two different types of words: real words and alternative words. By calculating the performance of real words and the performance of alternative words, a loss function for evaluating the performance of the generator model can be obtained. Electra's generative loss function is shown in equation (1):

$$\mathcal{L}_{gen} = \sum_t \max(-\log D_{\theta'}(\mathbf{w}t | \mathbf{x}) - \log(1 - (\tilde{\mathbf{w}}t | \mathbf{x}), -\log D_{\theta'}(\tilde{\mathbf{w}}t | \mathbf{x}) - \log(1 - \tilde{\mathbf{w}}t | \mathbf{x}))) \quad (1)$$

$\mathbf{x}$ denotes the input token sequence, $\mathbf{w}t$ denotes the t^{th} token in the original token sequence, $\tilde{\mathbf{w}}t$ denotes the new token obtained after generator transformation, $D_{\theta'}$ denotes the parameter of the discriminator, and θ' denotes the parameter that gets frozen in the generator network.

The discriminator model is a binary classification model where the samples generated by the model are labelled as real samples and those created by the generator model are labelled as forged samples. By alternately training the generator and the discriminator, the discriminator can learn to distinguish forged samples from real samples, and the generator can optimise its performance by learning to generate adversarial forged samples. The following is Electra’s discriminant loss function formula (2):

$$\mathcal{L}_{disc} = \sum_t t \sum_{c \in \{0,1\}} -\log D_{\theta^*}(c | \mathbf{w}t, c = g(\tilde{\mathbf{w}}t, \mathbf{x}; \theta, \tilde{\theta})) \quad (2)$$

D_{θ^*} denotes the network of the discriminator, g denotes the network of the generator, θ and $\tilde{\theta}$ denote the parameters used to train the discriminator and generator, respectively. In prediction, we directly use the parameters of the discriminator for feature extraction. The total objective function is computed by adding the two parts of generator loss and discriminator loss in equation (3):

$$\underset{\theta, \tilde{\theta}, \theta', \theta''}{\text{minimise}} \quad \mathcal{L}_{gen}(\theta, \tilde{\theta}, \theta') + (\theta, \tilde{\theta}, \theta'') \quad (3)$$

2.2 XLNet algorithm

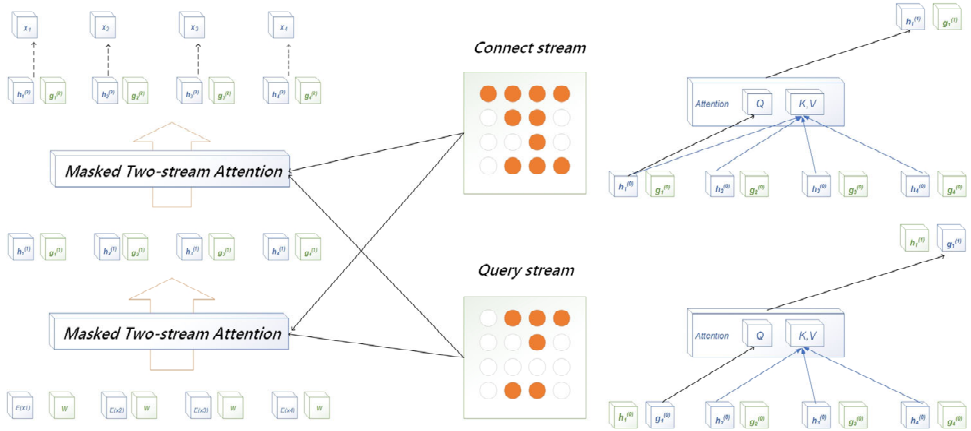
The XLNet model (Yang et al., 2019) is a pre-training model based on an autoregressive language model with two basic pre-training tasks: the unmasked language model (ULM) and the permutation language model (PLM). The ULM is similar to the traditional autoregressive language model, with the difference that it uses a more efficient transformer decoder for simultaneous processing of multiple locations. PLM is a unique pre-training task that performs better in processing long text sequences and extracting sentence-level text features.

In the PLM pre-training task, XLNet randomly arranges the sentences and then randomly selects words at each position for the operation of whether to mask or not, so that each position has a different word and masking pattern. The task of the model is to predict the masked word, all the words preceding it, and the masked word. This is shown in Figure 2.

The XLNet model predicts the i^{th} word by first inputting all words after that word as a mask, (i.e., the entire sentence as a context), and then generating probability distributions for the i^{th} word and previous words simultaneously. The loss function of the model is the sum of the negative log-likelihoods at all positions. The loss function of XLNet is given by equation (4):

$$\mathcal{L} = -\sum_{i=1}^n \log p(w_i | w_{<i}, w_{>i}) \quad (4)$$

$\mathcal{L}$ represents the loss function of XLNet, $w_{<i}$ denotes all the words before the i^{th} word, $w_{>i}$ denotes all the words after the i^{th} word, and $p(w_i | w_{<i}, w_{>i})$ denotes the probability of predicting the i^{th} word when the contexts $w_{<i}$ and $w_{>i}$ are known. When predicting, XLNet coordinates all the contextual information and considers all the words for each position, which enables a better understanding of the semantics of the sentence.

Figure 2 Structure of XLNet model (see online version for colours)

2.3 Feature fusion

Text feature fusion can combine multiple information or data sources with different characteristics to achieve complementarity between advantageous features to obtain richer text features. Feature fusion can be categorised into early fusion, intermediate fusion and late fusion (Gandhi et al., 2023). In this paper, we apply the early fusion method, which utilises a pre-training model to transform the original word vectors into high-dimensional feature representations, then fuses the obtained high-dimensional features, and adopts the fused feature information to be used for training classifiers. Common feature fusion means include splicing, attention mechanism and fusion gate mechanism (Poria et al., 2018). In this paper, the MCB algorithm is chosen for feature fusion.

MCB algorithm (Fukui et al., 2016) is a method for fusing multiple text features. It can handle multiple sources of information such as text, images, speech, etc. and fuse them into a vector representation for tasks such as classification or retrieval. The principle of the MCB algorithm is to transform each feature into a low-dimensional vector, and then these vectors are subjected to Hadamard product and stochastic mapping. Hadamard product is an element-by-element multiplication operation that fuses multiple vectors and does not require them to be expanded into a matrix. Stochastic mapping is a technique for mapping vectors into higher dimensional spaces, which captures more nonlinear relationships. The MCB algorithm for equation (5):

$$f(x, y) = \phi_w(x) \odot \phi_w(y) \quad (5)$$

where ϕ_w denotes the mapping of vectors into a low-dimensional space and represents this mapping with a random matrix w . $\odot$ denotes the bit-by-bit product or Hadamard product, and x and y represent different input features.

The advantage of the MCB algorithm is that it can process information from multiple input sources simultaneously and learn the nonlinear relationships between them. Also, the algorithm does not need to expand high-dimensional vectors into matrices for processing, so the operation is faster.

2.4 TextRank algorithm

TextRank algorithm (Mihalcea and Tarau, 2004) is a graph algorithm for text summarisation and keyword extraction. It is based on the PageRank (Brin and Page, 1998) algorithm, which constructs a word graph by calculating the similarity between words in the text, then, the importance of each word is calculated by using the connectivity between words in the graph, and finally, the keywords or summaries are obtained by sorting the nodes according to their scores. In comments, words closely related to emotional expressions tend to appear frequently together with other emotionally related words. The TextRank algorithm not only takes into account the importance of the keywords themselves, but also their position in the text, the frequency of contextual appearances, etc. and determines their importance by analysing the co-occurrence relationship between words, which can more accurately capture keyword semantic information than other algorithms. In addition, fresh produce reviews are large in scale and obtaining large amounts of labelled data may be difficult or costly, whereas TextRank is an unsupervised algorithm that does not require pre-labelled data for training, and can effectively improve the efficiency of processing large amounts of text data.

The TextRank algorithm works as follows: suppose there is a text that contains N words, denote these words as $V = V_1, V_2, \dots, V_N$. Next, construct an undirected weighted graph G , where each word corresponds to a node in the graph. The edges between nodes represent the similarity between words, and the weights of the edges represent the magnitude of the similarity. An $N \times N$ matrix W is used to represent the weights of the edges of the graph G . The element w_{ji} of matrix W represents the similarity between words V_i and V_j . When constructing the graph G , the similarity between words needs to be calculated. Commonly used calculation methods are cosine similarity, Jaccard similarity, etc. Next, the importance score of each node is calculated based on the connectivity between nodes in the graph. Initially, each node has a score of 1. where an N -dimensional vector S is used to represent the score of a node. The element $Score(V_i)$ of the vector S represents the score of the node V_i . Then, the score of each node is computed iteratively until convergence. At each iteration, the score of the node is updated based on the scores of the node's neighbouring nodes. The iterative update is shown in equation (6):

$$Score(V_i) = (1 - d) + d \times \sum_{V_j \in In(V_i)} \frac{w_{ji}}{\sum_{V_k \in Out(V_j)} w_{jk}} \times Score(V_j) \quad (6)$$

where $Score(V_i)$ denotes the score of the node V_i , d is the damping factor (generally takes the value of 0.85), $In(V_i)$ denotes the set of edges pointing to the node V_i , $Out(V_i)$ denotes the set of edges departing from the node V_i , and w_{ji} denotes the weights of the edges V_j, V_i . Finally, the keywords are extracted based on the score ordering of the nodes.

2.5 Attention mechanisms

Attention mechanism (Vaswani et al., 2017) is a type of data processing in machine learning that filters and focuses the input information in time or space in response to the input information to help the model learn the critical information. In deep learning, common attention mechanisms are dot product attention, self-attention, multi-head

attention, bilinear attention, etc. (Niu et al., 2021). Since the multi-head attention mechanism is to perform multiple different linear transformations of the input vectors and map them into multiple different attention subspaces, it has the advantage of considering the representation of the inputs more comprehensively, and at the same time, it can amicably improve the computational efficiency when dealing with large-scale data is to use parallel computing. The expression of multiple attention is equation (7):

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W_O \quad (7)$$

where the formula for each head is (8):

$$\text{texthead}_i = \text{Attention}(QW_{Qi}, KW_{Ki}, VW_{Vi}) \quad (8)$$

Attention here uses scaled dot-product attention, the formula is (9):

$$\text{Attention}(Q, K, V) = \text{Soft max} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (9)$$

In the above equations (7)–(9), Q , K , and V stand for query, key, and value, respectively, which are all different representations of the input sequence. W_{Qi} , W_{Ki} , W_{Vi} , and W_O are the parameters to be learned by the model. $\sqrt{d_k}$ is the dimension of the key, which is used to scale the result of the dot product to prevent its value from being too large.

2.6 BIGRU mechanism

BIGRU is a recurrent neural network model developed based on GRU (Cho et al., 2014). GRU, as a gated recurrent neural network, simplifies the three gating units of the LSTM into two, i.e., the reset gate and the update gate, which effectively reduces the computational and parametric quantities of the model. Based on GRU, BIGRU further adds a reverse loop layer, which enables the model to read input sequences from two directions at the same time, thus better utilising contextual information. At each time step, BIGRU maintains both a forward and a backward state vector. The forward state vector takes all input information from the beginning of the sequence to the current position, and the backward state vector takes all input information from the end of the sequence to the current position. In this way, the model can simultaneously capture the correlation between the front and back of the sequence and use this information to predict the output of the next state. In BIGRU, different types of gating units are introduced, including reset gates, update gates, and output gates. The reset gate decides how much information about the previous state is to be discarded, while the update gate decides how much new information is to be retained. Output gates, on the other hand, decide how much information to output and pass the output information to the neurons in the next layer. With these gating units, the model can control the flow of information and pruning, thus reducing the problem of gradient vanishing. The core formulation of BIGRU is shown in (10)–(11):

- Forward GRU:

$$h_t^f = \text{GRU}^f(x_t, h_{t-1}^f) \quad (10)$$

- Backward GRU:

$$h_t^b = \text{GRU}^b(x_t, h_{t-1}^b) \quad (11)$$

Among them, x_t is the t^{th} element of the input sequence h_t^f and h_t^b are the hidden states of the forward and backward GRUs at time step t , respectively, and GRU^f and GRU^b are the function representations of the forward and backward GRUs, which contain the internal computation process of the GRU unit in equation.

Alternatively, the hidden states of the forward and backward GRUs can be merged by splicing or other means to obtain a synthesised representation of equation (12), where h_t is the final representation that contains contextual information in both directions.

$$h_t = [h_t^f, h_t^b] \quad (12)$$

3 A sentiment classification model for fresh produce reviews based on joint semantic and sentiment optimisation

The structure of the sentiment classification model for fresh produce reviews (EXBMA) based on joint optimisation of semantics and sentiment is shown in Figure 7, which consists of six main parts: input layer, feature extraction layer, feature fusion layer, multi-head attention layer, BIGRU layer and output layer. The workflow of the model is shown in Figure 3.

3.1 Input layer

This layer first converts the review columns in the dataset into list type and encodes them using the tokeniser of the Electra model and XLNet model respectively. The encoded samples are padded and truncated using padding and truncation to ensure that all the samples are of the set maximum length.

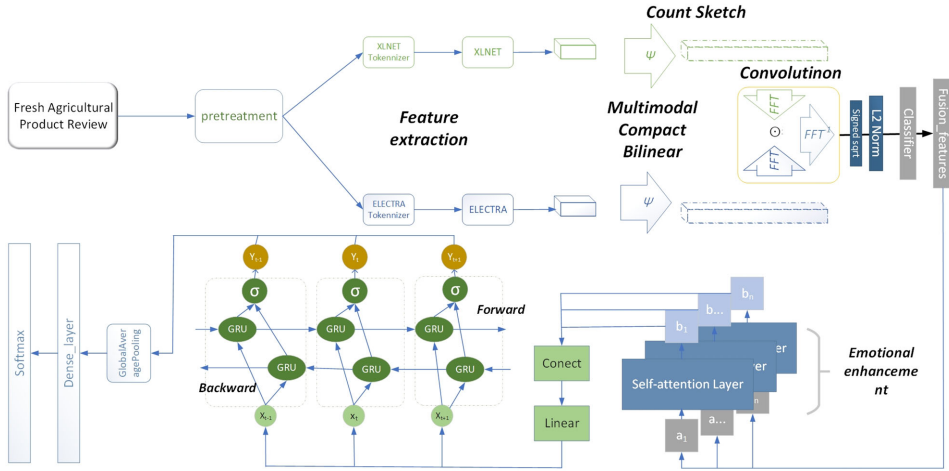
3.2 Semantic feature extraction layer

As consumers are more subjective in their reviews, it makes them somewhat transitive. For example: ‘Pretty fresh and crispy did a good job of keeping it fresh. But quite small and flavourless’. The first half of the sentence praises the product, while the second half raises some shortcomings, if only one part of the one-sided analysis is easy to classify the medium review as a good or bad review, so it can be seen that the semantics of the sentence as a whole is crucial to the impact of the sentiment orientation of the fresh produce review. Here, we used XLNet to extract sentence-level features of fresh produce reviews. XLNet model is a pre-training model based on the autoregressive language model, which uses the transformer architecture to build a decoder that can take into account the contextual information of the entire sentence, thus enabling a better understanding of the semantics at the sentence level; there are many types of fresh produce, and consumers are prone to use words with a high degree of similarity in their comments. For example: ‘authentic Lipu taro, soft and sticky texture, sweet flavour, positive feedback’ and ‘cooking hot pot to eat is the kind of super powder sticky, very delicious, the favourite taro cooking hot pot to eat, spicy hot pot, spicy, taro is also a little

sweet'. The meanings of 'soft and sticky' 'pink and sticky', 'sweet' and 'somewhat sweet' are very similar. Electra's word-level masking approach and adversarial generation mechanism are effective in capturing word-level features in texts that are semantically similar but use different word representations.

In this paper, we combine the advantages of sentence-level and word-level pre-training models to synergistically extract text features for fresh produce reviews using XLNet and Electra. The XLNet model employs permutation operation during training to better capture the local dependencies in the text. The Electra model, on the other hand, employs Adversarial training during training, which can better learn the global features in the text. The difference in the way the text is processed and the pre-training task makes them complementary in feature extraction.

Figure 3 Structure of the EXBMA (see online version for colours)



3.3 Feature fusion layer

The fusion of features extracted by Electra and XLNet can obtain more accurate semantic representations from sentence-and word-level perspectives. Due to the sparse sentiment features of fresh produce reviews, ordinary splicing or weighted summation will lead to a large amount of noise and outliers in the fused features, which seriously affects the results of sentiment classification, and the pre-training model extracted text features of high dimensionality, the use of traditional fusion algorithms, such as CCA, DCA and so on, has poorer results (Zhao et al., 2017; Joshi et al., 2018) and it needs to consume a large amount of memory and resources in the processing, which greatly increases the time cost of training. To solve the above problems, this paper chooses the MCB algorithm to fuse the text features of fresh produce reviews. MCB algorithm adopts the modular clustering method, dividing the data into multiple subspaces for clustering, which effectively solves the problem of the high dimensionality of the text features of the reviews of fresh produce. The MCB algorithm can capture the interactions between features using bilinear pooling, improve the effect of the interactions between features, learn the complex combinations between different features, and dynamically adjust the importance of features so that the key features are given more attention in the feature

fusion process. This not only increases the tightness of the textual features of fresh produce reviews but also reduces the noise and outliers in the data, which can help the model classify the sentiment of fresh produce reviews more stably.

In this layer, we first convert the outputs of Electra and XLNet into two-dimensional matrices, respectively, and then map the outputs of Electra and XLNet into a low-dimensional space using two random matrices, W_1 and W_2 , respectively, using random projection. Next, the eigenvalues of the outputs of Electra and XLNet in each dimension are discretised using a hash function. By taking the dot product of each eigenvalue and a random vector, and then obtaining the sign function (i.e., taking 1 for positive numbers and -1 for negative numbers), a binary vector of -1 or 1 is obtained, i.e., the hash code. The two hashed feature vectors are then fused using the Hadamard product. The Hadamard product is the new vector obtained by multiplying the two vectors bitwise. This operation effectively preserves the important information in both vectors. Finally, the function maps the fused vectors back to the original high-dimensional space using the inverse discrete cosine transform (IDCT) and converts them to three-dimensional vectors as inputs to the subsequent layers, a process that effectively reduces the loss of information, resulting in a better feature representation.

The fusion process of the MCB algorithm is as follows:

- 1 Convert the outputs of Electra and XLNet into a two-dimensional matrix:
 - Electra output matrix: $E = [e_{ij}]$, $1 \leq i \leq m, 1 \leq j \leq n$, which is a matrix with m rows and n columns, where each element e_{ij} denotes the output value of the Electra model on the i^{th} sample and the j^{th} feature.
 - XLNet output matrix: $X = [x_{ij}]$, $1 \leq i \leq p, 1 \leq j \leq q$, which is a p -row q -column matrix where each element x_{ij} denotes the output value of the XLNet model on the i^{th} sample and the j^{th} feature.
- 2 Multiply the Electra output matrix (E) and the XLNet output matrix (X) with the random matrices W_1 and W_2 to obtain the Electra randomly projected matrix (E') and the XLNet randomly projected matrix (X') as shown in equations (13)–(14):

$$E' = E * W_1 \quad (13)$$

$$X' = X * W_2 \quad (14)$$

The dimension of E' is the same as the number of columns of W_1 , and each element e'_{ij} denotes the value of the j^{th} feature obtained by the Electra model after random projection on the i^{th} sample, and the same for X' . The use of random projection and random matrix mapping is to map the outputs of Electra and XLNet into a low-dimensional space, thus reducing the dimensionality of the features and extracting important information.

- 3 The Electra discretised matrix (E') and the XLNet discretised matrix (X') are obtained by applying the hash function to discretise each element of the Electra randomly projected matrix (E'') and the XLNet randomly projected matrix (X'') as shown in equations (15)–(16):

$$E'' = \text{Hash}(E') \quad (15)$$

$$X'' = \text{Hash}(X') \quad (16)$$

Using (15)–(16), continuous eigenvalues can be mapped to integers or binarised to -1 or 1 . The discretised matrix E'' has the same dimensions as E' , and each element e''_{ij} denotes the discretised value of the corresponding element in E' after hash function processing, and the same for X'' .

- 4 The elements of the discretised matrix are passed through the sign function (Sign) to obtain the hash code vector $H_E = [h_{e1}, h_{e2}, \dots, h_{em}]$ and the hash code vector $H_X = [h_{x1}, h_{x1}, \dots, h_{xp}]$. Where $h_{ei} = \text{Sign}(E''_i)$, $h_{xi} = \text{Sign}(X''_i)$, the sign function maps positive numbers to 1 and negative numbers to -1 . Thus, for Electra's hash-code vector H_E , each element h_{ei} denotes the sign of the eigenvalue of the corresponding position in the discretised matrix E'' . Similarly, for XLNet's hash code vector H_X , each element h_{xi} denotes the sign of the eigenvalue of the corresponding position in the discretised matrix X'' .
- 5 The hash code vector H_{Fusion} after feature fusion is obtained by performing the Hadamard product operation on the hash code vector (H_E) of Electra and the hash code vector (H_X) of XLNet. The Hadamard product operation is an element-by-element multiplication operation, i.e., multiplying the elements at corresponding positions in (H_E) and (H_X). The fused hash code vector H_{Fusion} has the same dimension and each element represents the feature representation after feature fusion. Finally, the fused hash code vector H_{Fusion} is subjected to the IDCT to obtain a fused 3D vector V_{Fusion} , where the elements represent the final features.

3.4 Multi-head attention layer

Fresh produce reviews contain a large amount of emotional information related to the attributes of fresh produce such as taste, freshness, nutritional value, etc. which may differ from other commodities. The use of the attention mechanism enables automatic learning to focus on one aspect of the sentiment representation in a review sequence. The multi-head attention mechanism allows the model to compute the attention weights in parallel in different representation subspaces, which helps to improve the computational efficiency. In this paper, we use the information from the sentiment dictionary of fresh produce reviews to optimise the allocation of multi-head attention weights and design a sentiment-enhanced multi-head attention mechanism that can capture the sentiment information in reviews more comprehensively.

In this layer, the TextRank algorithm is first used to extract sentiment keywords from a large number of fresh produce review datasets to construct a fresh produce review sentiment dictionary, and then a multi-head attention mechanism is constructed to compute the similarity of each word to the sentiment vocabulary in the sentiment dictionary, and the similarity between the word and the sentiment vocabulary in the dictionary is used as part of the attention weights, and the following is the computational formula for updating the attention weights (17):

$$a_{ij} = \text{soft max} \left(\left(w_i \cdot x_j + \sum (w_{ip} \cdot \text{sim}(x_j, p_k)) \right) / \sqrt{d} \right) \quad (17)$$

In equation, w_i is the weight matrix of the attention head i , w_{ip} is the weight matrix used for sentiment lexicon fusion, $\text{sim}(x_j, p_k)$ is the similarity between the word x_j and the

sentiment lexicon p_k , and $\sqrt{d}$ is a scaling factor to control the range of the attention weights.

Incorporating the information from the sentiment lexicon into the attention distribution, with the help of the multi-head attention mechanism by learning different attention patterns can help the model to capture the long-distance dependencies in the reviews of fresh produce, which is useful for understanding the sentiment-related semantic connections in the reviews.

3.5 BIGRU layer

BIGRU can simultaneously consider the contextual information before and after the review sequences, and learn the complex features in the information from a structural perspective, which helps the model better understand the semantic information and emotional tendencies in the fresh produce reviews. In addition, the text sequences of fresh produce reviews have a large gap in length, and BIGRU can handle sequences of arbitrary length, which enables the model to better handle fresh produce review data. In addition, by placing the BIGRU in the last layer of the model, this paper can minimise the loss of context information problem that may occur in other layers before the BIGRU, and ensure that the model fully considers all the relevant information of the whole input sequence before making the final prediction, and the gating unit of the BIGRU can also alleviate the gradient vanishing problem and ensure that the gradient can efficiently flow to the previous layers.

3.6 Output layer

The BIGRU output information `gru_output` is integrated using global average pooling to get a global feature `global_avg_pool_output`, which reduces the number of parameters of the model to avoid overfitting, and then the output feature information is accessed to the full connectivity layer, and the Softmax function is used to achieve the classification of the text, and the formula is as (18) is shown:

$$\text{softmax}(Z_i) = \frac{e^{Z_i}}{\sum_{j=1}^n e^{Z_j}} \quad (18)$$

In equation, Z_i is the i^{th} element in the input vector Z , and $\sum_{j=1}^n e^{Z_j}$ is the sum of the exponential operation performed on all input vector elements.

The steps of EXBMA model implementation are shown in Table 1.

3.7 Computational complexity analysis

This section focuses on analysing the computational resource requirements of the EXBMA model in practical computations. Below is an explanation of the primary modules' computational complexity within the EXBMA.

- **Electra model computational complexity:** This paper employs Electra's discriminator to extract word-level semantic features from text sequences. The discriminator is primarily constructed with a transformer module, within which the self-attention

mechanism constitutes the dominant source of computational complexity. For a self-attention mechanism operating on a sequence of length L with vector dimension D , its computational complexity is $O(L^2 \cdot D)$.

- XLNet model computational complexity: the computational complexity of XLNet predominantly arises from its masked bi-stream attention mechanism. Notably, masking sets certain attention weights to minimal values (often negative infinity) during computation, causing them to be disregarded in subsequent Softmax operations. In theory, if masking effectively reduces the number of query-key pairs involved in calculations, computational load can be mitigated. However, in most cases, masking does not significantly alter the order of the overall complexity, as computations must still traverse the entire sequence to identify masked positions. Consequently, the self-masked bi-stream attention mechanism, akin to the self-attention mechanism, has a computational complexity approximately $O(2 \cdot L^2 \cdot D)$.
- MCB algorithm computational complexity: the analysis of MCB algorithm's computational complexity encompasses multiple steps, including matrix transformations, application of hash functions, sign function applications, Hadamard product operations, and IDCT. The primary contributors to MCB's computational complexity are the matrix transformation process, Hadamard products, and IDCT operations. Under simplified assumptions, the overall complexity can be estimated as:

$$O(mn \times M_{W_1} + pq \times M_{W_2} + N^2 + 3N \log N)$$

where m and n denote dimensions of Electra's output matrices, p and q represent dimensions of XLNet's output matrices, and M_{W_1} and M_{W_2} are the last dimensions of parameter matrices W_1 and W_2 , respectively. It is noteworthy that lower-order terms and constant factors are ignored for simplification, and $N = \max(m, n, p, q)$ is assumed.

- Multi-head attention module: the overall computational complexity of the multi-head attention module is primarily influenced by the accumulation of cosine similarity computations. In a simplified form, the total complexity can be approximated as:

$$O(L^3 \cdot n_{heads} \cdot d_k + L^2 \cdot d_{model})$$

where L is the sequence length, n_{heads} is the number of heads in the multi-head attention mechanism, d_k is the dimension per head, and d_{model} is the model dimension.

- BiGRU module: for a single-layer BiGRU network, each layer consists of three sets of weight matrices (input-to-hidden, hidden-to-hidden, and bias terms), with identical hidden layer sizes, leading to a computational complexity of:

$$O(2dh + 2h^2 + 4h)$$

where h is the hidden layer size and d is the input dimension.

These modules constitute the primary sources of computational complexity in the EXBMA model. Actual operation counts and parameter quantities in practical computations are detailed in Table 2 (assuming an input sequence length of 128).

Table 1 EXBMA modelling process

<i>Semantic and sentiment optimisation based sentiment classification model for fresh produce reviews EXBMA</i>	
Input: user reviews P ; sentiment dictionary of fresh produce reviews emotion_weights_dict	
Output: Emotional tendency of user reviews S	
/* Feature extraction layer */	
1	Use the electra model to extract word-level semantic features $E = [e_{ij}]$, $1 \leq i \leq m$, $1 \leq j \leq n$ in user comments P
2	Use the XLNet model to extract sentence-level semantic features $X = [x_{ij}]$, $1 \leq i \leq p$, $1 \leq j \leq q$ in user comments P
/* Fusion layer */	
3	Use the MCB algorithm to fuse electra output matrix E and XLNet output matrix X to get V_{Fusion}
/* Multi-attention layer */	
4	Calculate the similarity of each word to the sentiment words in the sentiment dictionary $\text{sim}(x_j, p_k)$ and update the attention weights, refer to equation (17)
5	Output attn_output via multi-head attention
/* BiGRU layer */	
6	Input attn_output into BiGRU to get contextual features in comments gru_output
/* Output layer */	
7	Take the output from BiGRU and get a global feature global_avg_pool_output by global average pooling
8	Use the Softmax function to output the sentiment tendency of user comments S

Table 2 Computational complexity analysis

<i>Work flow</i>	<i>Output shape</i>	<i>Flops</i>	<i>Parameter</i>
Electra-base-discriminator	(128, 768)	21,914.12 M	85.65 M
Chinese-XLNet-base	(128, 768)	38,553.84 M	56.71 M
Multimodal compact bilinear pooling	(128, 768)	301.98 M	1.18 M
Multi head attention 1	(768, 256)	12,884.9 M	0.26 M
bidirectional 1	(128, 256)	48,695.86 M	1.68 M
global_average_pooling1d_1	(256,)	0.06 M	-
dense_layer(Dense)	(128,)	0.06 M	0.03 M
dense_3(Dense)	(3,)	-	-

Table 2 primarily reflects the number of operations consumed and the parameter size occupied by each module during the actual computation process, wherein the BiGRU layers and the XLNet module account for the majority of the computational operations. The Electra discriminator module and the XLNet module, on the other hand, dominate in terms of parameter volume.

4 Experiments

4.1 Experimental data and environment construction

In this experiment, Python crawler is used to build the experimental dataset by crawling the data from Jingdong Fresh Food Zone and obtaining 19,753 online reviews of several kinds of products, including fruits, vegetables, meat, eggs, etc. including 8,232 good reviews, 5,682 medium reviews, and 5,839 bad reviews. The constructed dataset is segmented using Jieba segmentation, and the Chinese deactivation word list is utilised to remove some words with no practical significance to complete the pre-processing of the original data. Finally, 80% of the dataset is used for model training and 20% is used for model testing, where 20% of the training set is used as the validation set.

The experimental platform is Windows 10 operating system, the hardware is Intel i9-13900k processor, 64 GB RAM, RTX3090 graphics card, the model is mainly implemented using Python language, Python version is 3.8.15, the deep learning libraries used include Tensorflow2.5.0, Keras2.10.0, Gensim4.2.0, Numpy1.20.3, and the coding was done through Visual Studio Code development tool.

4.2 Evaluation criteria

In this paper, accuracy (A), precision (P), recall (R), and F1 values are used as the evaluation criteria for the experiment, the computation of these metrics relies on the confusion matrix, which in a binary classification problem usually contains four elements: true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), as shown in Table 3 shown.

Table 3 Confusion matrix

<i>Real results</i>	<i>Predicted results</i>	
	<i>Positive example</i>	<i>Negative example</i>
Positive example	TP	TN
Negative example	FP	FN

Accuracy (A) is the proportion of correctly categorised samples to the total sample and is calculated as (19):

$$A = \frac{T_{TP} + T_{TN}}{T_{TP} + T_{TN} + F_{FP} + F_{FN}} \quad (19)$$

Precision (P) is the proportion of samples predicted as correct to the proportion of correct samples in all predicted categories and is calculated as (20):

$$P = \frac{T_{TP}}{T_{TP} + F_{FP}} \quad (20)$$

Recall (R) is the proportion of positive cases correctly identified by the model to all true positive cases and is calculated as (21):

$$R = \frac{T_{TP}}{T_{TP} + F_{FN}} \quad (21)$$

The F1 value is the reconciled average of precision and recall, calculated as (22):

$$F1 = \frac{2 \times P \times R}{P + R} \quad (22)$$

4.3 Experimental results

4.3.1 Experimental parameter configuration

The pre-trained models used in this model are Electra-base-Discriminator and Chinese-XLNet-base and the optimal parameters are derived after several experiments as shown in Table 4.

Table 4 Parameter settings

<i>Parameters</i>	<i>Value</i>
MATT-num_heads	4
MATT-key_dim	512
GRU-units	384
Dropout	0.5
Activation	Relu
L2 regularisation	0.009
learning_rate	0.0001
Optimiser	Adam
Loss	categorical_crossentropy
batch_size	32
Epochs	13

The changes in the accuracy and loss values of the model with the number of training rounds for both the training and validation sets during training are shown in Figures 4 and 5. Figure 4 demonstrates that the accuracy of the model on the training and validation sets steadily increases with the number of training rounds, which proves that the model gradually learns more accurate prediction ability during the training process. Meanwhile, Figure 5 shows that the loss value of the model steadily decreases with the increase in the number of training rounds, which indicates that the model gradually reduces the error during the learning process. This trend of steadily increasing and remaining stable during the experimental process indicates that no significant overfitting or underfitting of the model occurs, proving that the model has good performance and generalisation ability.

Figure 4 Variation of accuracy with the number of training in training and validation sets (see online version for colours)

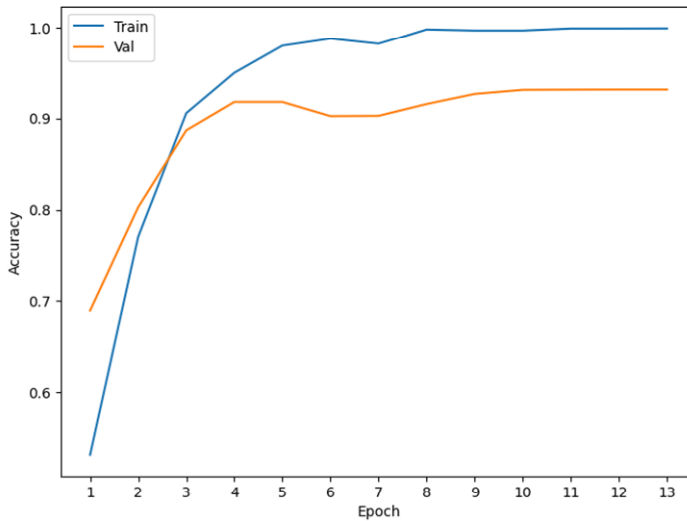
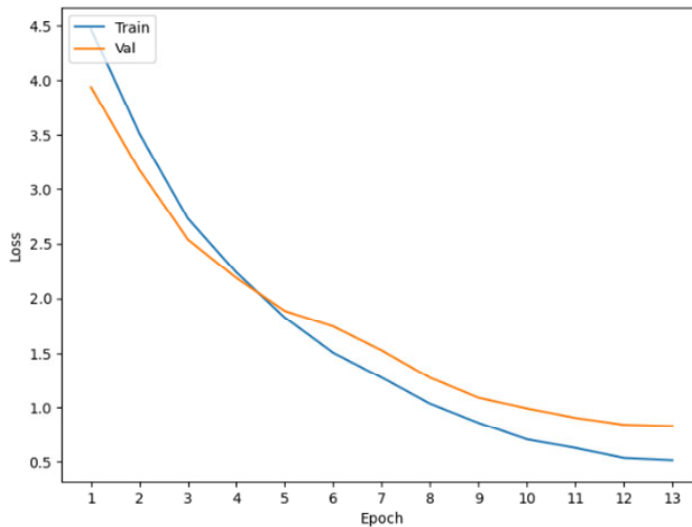


Figure 5 Variation of loss values with the number of training in training and validation sets (see online version for colours)



4.3.2 Comparison experiment

To verify the effectiveness of the proposed model, this paper is compared with the following models.

- 1 BIGRU (Cho et al., 2014): Word2Vec (Mikolov et al., 2013) is utilised to complete the word embedding representation of the input sequence, and bi-directional GRU is used to extract contextual semantic and sequential information of the text for classification.
- 2 TextCNN (Kim, 2014): using Word2Vec to complete the word embedding representation of the input sequence, using convolutional operations to extract local key features of the text, and then using maximum pooling to filter out the obvious features and classify the text based on this information.
- 3 Bert (Devlin et al., 2018): using Bert to complete the word embedding representation of the input sequence, and directly output to the Softmax layer for classification.

The results of the comparison experiments are shown in Table 5.

Table 5 Comparative experimental results

<i>Approaches</i>	<i>R/%</i>	<i>P/%</i>	<i>A/%</i>	<i>F1/%</i>
BIGRU	88.44	89.87	88.44	88.71
TEXTCNN	90.67	90.77	90.66	90.54
BERT	90.68	91.70	91.74	91.61
EXBMA	93.77	93.80	93.77	93.75

As shown in Table 5, the sentiment classification model for fresh produce reviews proposed in this paper based on the joint optimisation of semantics and sentiment is improved in terms of recall, accuracy, and F1 value compared to the mainstream models. Among them, BIGRU and TEXTCNN have accuracies of 88.44% and 90.54%, respectively, and the classification results are unsatisfactory since Word2Vec maps each word to a unique word-vector representation, which is not able to deal with polysemous words well. Bert, which can learn the contextual information of polysemous words from a large-scale corpus, can provide a more accurate word-vector representation, and the accuracy value is 3.3% and 1.08% higher than that of BIGRU and TEXTCNN by 3.3% and 1.08% respectively. Compared with the Bert model, the EXBMA selected in this paper contains richer and more accurate text features and pays more attention to the sentiment features, which improves by 2.03% in accuracy. The experimental results show that the EXBMA is better for sentiment classification of fresh produce reviews.

4.3.3 Semantic enhancement ablation experiment

To verify the effect of fusing two features in the experimental results, three ablation experiments are conducted in this paper, and except for the modification of the feature extraction part, other structures and parameters are kept unchanged, and the specific experimental settings are as follows:

- XBM: only use the XLNet pre-training model to complete the feature extraction of the input sequence.
- EBM: only use the Electra pre-training model to complete the feature extraction of the input sequence.

- EXBMA: use the XLNet pre-training model and Electra pre-training model to extract features and then perform feature fusion.

Table 6 Results of semantic enhancement ablation experiments

<i>Ablation experiments</i>	<i>R/%</i>	<i>P/%</i>	<i>A/%</i>	<i>F1/%</i>
XBM	91.85	92.01	91.88	91.76
EBM	91.58	91.66	91.66	91.65
EXBMA	93.77	93.80	93.77	93.75

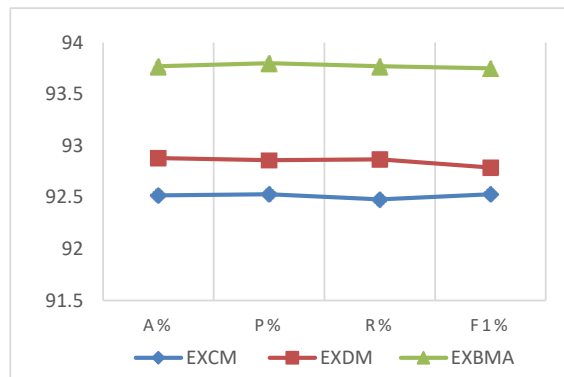
As shown in Table 6, this paper's method improves in recall, accuracy and F1 value compared to XBM and EBM. The above results show that the fusion of sentence-level and word-level text features in this paper can obtain more complete semantic features.

4.3.4 Feature fusion ablation experiment

To verify the effect of the feature fusion method on the experimental results, three ablation experiments were conducted in this paper, and the structure and parameters were kept unchanged except for the modification of the feature fusion part, and the specific experimental settings are as follows:

- EXCM: feature fusion using the splicing Concat method.
- EXDM: feature fusion using DCA algorithm.
- EXBMA: feature fusion using MCB algorithm.

As shown in Figure 6, EXBMA is higher than EXCM and EXDM in recall, precision, accuracy and F1 value, the above results show that compared to the algorithms, ordinary splicing is prone to produce more noise values when features are fused, while MCB algorithm fuses the high-dimensional features with less noise, and the fused information is more expressive.

Figure 6 Results of feature fusion ablation experiments (see online version for colours)

4.3.5 Attention mechanism ablation experiment

To verify the effect of the emotion lexicon combined with the multi-head attention mechanism on the experimental results, the following ablation experiments were used, with the structure and parameters remaining unchanged except for the part of the multi-head attention mechanism, which was modified, and the specific experimental setup was as follows:

- EXBA: using the attention mechanism.
- EXBM: using the multi-head attention mechanism.
- EXBMA: using the emotionally enhanced multi-head attention mechanism.

Table 7 Results of the attention mechanism ablation experiment

<i>Ablation experiments</i>	<i>R/%</i>	<i>P/%</i>	<i>A/%</i>	<i>F1/%</i>
EXBA	92.05	92.25	92.27	92.30
EXBM	92.52	92.63	92.72	92.56
EXBMA	93.77	93.80	93.77	93.75

As shown in Table 7, EXBMA is higher than EXBA and EXBM in terms of recall, precision, accuracy and F1 value, which shows that multiple attentions combined with sentiment lexicon can further improve the accuracy of sentiment classification.

5 Conclusions

In this paper, a sentiment classification model (EXBMA) based on joint optimisation of semantics and sentiment is proposed to address the problem of underutilisation of semantic and key sentiment information in fresh produce reviews. The model obtains sentence-level and word-level semantic information at the feature extraction layer, and thereafter realises semantic fusion characterisation through the feature fusion layer; to enhance the expression of sentiment information in the reviews, the Multi-attention mechanism is used to enhance the model's attention to the sentiment words, and finally, the contextual features of the reviews are obtained at the BIGRU layer from the structural point of view to realise the final classification. In this paper, experiments are conducted on a dataset of self-constructed fresh produce reviews to evaluate the effectiveness of the proposed model. The experimental results show that the model proposed in this paper achieves significant improvements in performance metrics compared to other baseline models, and the ablation study demonstrates the feasibility of these improvements. In future work, we will build on this paper to investigate the sentiment analysis of product reviews in multimodal contexts by fusing and co-processing mediums such as images, videos, and texts to further strengthen and stabilise the classification performance of the model in this paper.

Acknowledgements

This work was supported by Anhui Department of Education Research Foundation (2022AH050896) and Anhui Natural Science foundation (2108085MF209) grant funded.

References

- Abdi, A., Shamsuddin, S.M., Hasan, S. et al. (2019) 'Deep learning-based sentiment classification of evaluative text based on Multi-feature fusion', *Information Processing & Management*, Vol. 56, No. 4, pp.1245–1259.
- Brin, S. and Page, L. (1998) 'The anatomy of a large-scale hypertextual web search engine', *Computer Networks and ISDN Systems*, Vol. 30, Nos. 1–7, pp.107–117.
- Cho, K., Van Merriënboer, B., Gulcehre, C. et al. (2014) *Learning Phrase Representations using RNN Encoder-Decoder For Statistical Machine Translation*, arXiv preprint arXiv:1406.1078.
- Clark, K., Luong, M.T., Le, Q.V. et al. (2020) *Electra: Pre-Training Text Encoders as Discriminators Rather Than Generators*, arXiv preprint arXiv:2003.10555.
- Cortes, C. and Vapnik, V. (1995) 'Support-vector networks', *Machine Learning*, Vol. 20, No. 3, pp.273–297.
- Cover, T. and Hart, P. (1967) 'Nearest neighbor pattern classification', *IEEE Transactions on Information Theory*, Vol. 13, No. 1, pp.21–27.
- Devlin, J., Chang, M.W., Lee, K. et al. (2018) *Bert: Pre-Training of Deep Bidirectional Transformers for Language Understanding*, arXiv preprint arXiv:1810.04805.
- Fukui, A., Park, D.H., Yang, D. et al. (2016) 'Multimodal compact bilinear pooling for visual question answering and visual grounding', arXiv preprint arXiv:1606.01847.
- Gan, C., Feng, Q., Zhang, Z. (2021) 'Scalable multi-channel dilated CNN-BiLSTM model with attention mechanism for Chinese textual sentiment analysis', *Future Generation Computer Systems*, Vol. 118, pp.297–309.
- Gandhi, A., Adhvaryu, K., Poria, S. et al. (2023) 'Multimodal sentiment analysis: a systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions', *Information Fusion*, Vol. 91, pp.424–444.
- Guo, X., Zhao, N. and Cui, S. (2020) 'Emotional analysis of consumer online comments based on CNN-BiLSTM', *System Engineering Theory and Practice*, Vol. 40, No. 3, pp.653–663.
- Hochreiter, S. and Schmidhuber, J. (1997) 'Long short-term memory', *Neural Computation*, Vol. 9, No. 8, pp.1735–1780.
- Huang, F., Li, X., Yuan, C. et al. (2021) 'Attention-emotion-enhanced convolutional LSTM for sentiment analysis', *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 33, No. 9, pp.4332–4345.
- Joshi, G., Vig, R. and Singh, S. (2018) 'DCA-based unimodal feature-level fusion of orthogonal moments for Indian sign language dataset', *IET Computer Vision*, Vol. 12, No. 5, pp.570–577.
- Kim, Y. (2014) *Convolutional Neural Networks for Sentence Classification*, arXiv preprint arXiv:1408.5882.
- Lahat, D., Adali, T. and Jutten, C. (2015) 'Multimodal data fusion: an overview of methods, challenges, and prospects', *Proceedings of the IEEE*, Vol. 103, No. 9, pp.1449–1477.
- LeCun, Y., Bottou, L. Bengio, Y. et al. (1998) 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, Vol. 86, No. 11, pp. 278–2324.
- Lewis, D.D. (1998) 'Naive (Bayes) at forty: the independence assumption in information retrieval', in Nédellec, C. and Rouveirol, C. (Eds.): *Machine Learning: ECML-98. ECML 1998. Lecture Notes in Computer Science*, Vol. 1398, Springer, Berlin, Heidelberg.

- Li, W., Qi, F., Tang, M. et al. (2020) 'Bidirectional LSTM with self-attention mechanism and multi-channel features for sentiment classification', *Neurocomputing*, Vol. 387, pp.63–77.
- Li, W.J., Qi, F. and Yu, Z.T. (2021) 'Sentiment classification method based on multi-channel features and self-attention', *Journal of Software*, Vol. 32, No. 9, pp.2783–2800.
- Mihalcea, R. and Tarau, P. (2004) 'TextRank: bringing order into text', *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pp.404–411.
- Mikolov, T., Chen, K., Corrado, G. et al. (2013) *Efficient Estimation of Word Representations in Vector Space*, arxiv preprint arxiv:1301.3781.
- Mikolov, T., Karafiát, M., Burget, L. et al. (2010) 'Recurrent neural network-based language model', *Interspeech*, Vol. 2, No. 3, pp.1045–1048.
- Misra, N.N., Dixit, Y., Al-Mallahi, A. et al. (2020) 'IoT, big data, and artificial intelligence in agriculture and food industry', *IEEE Internet of Things Journal*, Vol. 9, No. 9, pp.6305–6324.
- Niu, Z., Zhong, G. and Yu, H. (2021) 'A review on the attention mechanism of deep learning', *Neurocomputing*, Vol. 452, pp.48–62.
- Poria, S., Majumder, N., Hazarika, D. et al. (2018) 'Multimodal sentiment analysis: Addressing key issues and setting up the baselines', *IEEE Intelligent Systems*, Vol. 33, No. 6, pp.17–25.
- Pota, M., Ventura, M., Fujita, H. et al. (2021) 'Multilingual evaluation of pre-processing for BERT-based sentiment analysis of tweets', *Expert Systems with Applications*, Vol. 181, p.115119.
- Vaswani, A., Shazier, N., Parmar, N. et al. (2017) 'Attention is all you need', *Advances in Neural Information Processing Systems*, p.30.
- Wang, H., Li, J., Wu, H. et al. (2022) 'Pre-trained language models and their applications', *Engineering*, Vol. 25, pp.51–65.
- Yang, Z., Dai, Z., Yang, Y. et al. (2019) 'Xlnet: generalized autoregressive pretraining for language understanding', *Advances in Neural Information Processing Systems*, p.32.
- Yin, W., Kann, K., Yu, M. et al. (2017) *Comparative study of CNN and RNN for natural language processing*, arXiv preprint arXiv:1702.01923.
- Zhang, T., Gong, X. and Chen, C.L.P. (2021) 'BMT-Net: broad multitask transformer network for sentiment analysis', *IEEE Transactions on Cybernetics*, Vol. 52, No. 7, pp.6232–6243.
- Zhang, W., Li, L., Zhu, Y. et al. (2022) 'CNN-LSTM neural network model for fine-grained negative emotion computing in emergencies', *Alexandria Engineering Journal*, Vol. 61, No. 9, pp.6755–6767.
- Zhao, A. and Yu, Y. (2021) 'Knowledge-enabled BERT for aspect-based sentiment analysis', *Knowledge-Based Systems*, Vol. 227, p.107220.
- Zhao, F., Qiao, L., Shi, F. et al. (2017) 'Feature fusion via hierarchical supervised local CCA for diagnosis of autism spectrum disorder', *Brain Imaging and Behavior*, Vol. 11, No. 4, pp.1050–1060.
- Zhao, G., Lv, C., Fu, G. et al. (2021) 'Cross-domain attribute sentiment analysis based on domain-specific sentiment word attention model', *Chinese Journal of Information Technology*, Vol. 35, No. 6, pp.93–102.
- Zhu, Q., Liang, B., Xu, R. et al. (2022) 'A fine-grained sentiment analysis combining sentiment dictionaries and attention mechanisms in the financial field', *Chinese Journal of Information Technology*, Vol. 36, No. 8, pp.109–117.
- Zhuang, Y., Liu, Z., Liu, T.T. et al. (2022) 'Implicit sentiment analysis based on multi-feature neural network model', *Soft Computing*, Vol. 26, No. 2, pp.1–10.