Cross-Cultural Deep Learning Models for Sentiment Analysis Between Chinese and Japanese Using Sentiment Features

Fangyuan Su[1], Jinqing Liu[2] and Yuliang Hao[3*]

[1,3]College of Foreign Languages, Xingtai University, Xingtai, 054001, Hebei, China
[1]sufangyuan002@163.com,[3]yulianghao2011@163.com
[2]College of Foreign Languages, Handan University, Handan,056000, Hebei, China
[2]liujinqing1988@163.com

Corresponding author: Yuliang Hao, yulianghao2011@163.com

Abstract. With pre-trained bilingual word embedding (BWE) dictionaries, deep learning-based cross-language sentiment analysis models require text vector representations in both the source and target languages. This paper proposes a word vector sentiment feature representations-based cross-language text sentiment analysis method to address the challenge of hard-to-obtain BWE dictionaries. The technique explicitly introduces sentiment supervisory information to obtain word vector representations of the source language's sentiment perceptions. These representations of the word vectors consider both semantic and sentiment feature information and are used for cross-language text sentiment prediction. Chinese is the source language, and Japanese is the target language in cross-language sentiment analysis experiments. According to the experimental findings, the suggested model can increase prediction accuracy by roughly 9.3% and 8.7% compared to the cross-language sentiment analysis method without sentiment feature representation and the machine translation method. Given that Chinese and Japanese belong to the same language family and have similar syntax and semantics, the model performs best when used for cross-language sentiment analysis on Japanese, which is consistent with the experimental expectation.

Keywords: cross-language, sentiment analysis, text vectors, Chinese, Japanese, sentiment features; Cross-Cultural.
DOI: https://doi.org/10.14733/cadaps.2024.S20.126-142

1 INTRODUCTION

With a long history and rich cultural traditions, China and Japan have never had greater cooperation or communication than they do now, especially in the modern era. However, because of linguistic and cultural barriers, sentiment analysis can sometimes be exceedingly challenging. Previous approaches to cross-linguistic sentiment analysis have often relied on techniques such as machine translation, which have only sometimes yielded good results. New methods must be investigated to accurately convey the minute variations in sentiment expressions between China and Japan.

In intercultural communication, the focus is more on the cultural differences concealed behind language than on developing communicative competence, primarily "linguistic competence." Cross-cultural communication emphasizes the individuality and distinctiveness of many cultures and their mutual integration and promotion. It is flexible rather than rigid, pluralistic rather than monolithic. Therefore, it is beneficial to the continued development of communication and culture between China and Japan to examine the influence of Sino-Japanese homonyms from this communicative perspective.

Japanese words for "bed" and "floor" are easily misunderstood by Chinese learners who take after their mother tongue, literally translating the meaning to "book on bed." This makes communication between China and Japan much more challenging. Significant ambiguity in usage can also result from semantic differences. For example, the Chinese word "quiet" means "calm without noise" and can be used as a verb, adjective, or noun. Nevertheless, "quiet" is only a noun in Japanese and means "resting in illness.".

Cross-linguistic sentiment analysis faces many challenges due to the semantic and cultural differences between languages. Because they need help capturing the expression of sentiment in various contexts, traditional approaches to sentiment analysis frequently find it challenging to deal with these differences. This problem stems from the cultural variations that underlie multiple languages, which lead to the reality that the exact words can convey drastically different emotional hues in various settings. Conventional methods for solving this issue typically involve direct or machine translation, which frequently needs to be revised to the intricacies of semantics and culture.

Concurrently, the sentiment analysis model becomes more complex due to the presence of Chinese and Japanese homographs. Homographs may have different meanings in different languages but have the same form or spelling. Because the exact words can convey entirely different sentiments in two languages, homographs in sentiment analysis can result in false sentiment inferences. Deep learning models are expected to handle homographs better than traditional methods because they can learn richer semantic representations. There are some restrictions on how these issues can be addressed by current work. Traditional methods typically over-rely on grammatical constructions and literal word translations when addressing linguistic variations, ignoring the nuances of sentiment expressions. This renders inferences about sentiment in various contexts unclear. However, deep learning models have yet to be introduced for processing homographs to produce higher-level semantic representations.

By introducing a deep learning model based on sentiment features, this study seeks to overcome these obstacles and more accurately capture the subtle differences between semantics and culture. The model will be better equipped to comprehend sentiment expressions and handle homographs by gaining deeper semantic representations by adding sentiment-supervised data. This method has the advantage of taking sentiment analysis between China and Japan more thoroughly and adaptable. It is also anticipated to offer fresh perspectives and answers for ongoing cross-linguistic sentiment analysis research.

2 RELATED WORK

Sentiment analysis, a vital task in natural language processing, aims to determine the sentiment polarity of text. To address the increasing accuracy of sentiment classification algorithms, [10] suggests an integrated sentiment classification strategy that improves classification results on the Twitter dataset. This strategy is based on the majority voting principle of multiple classification methods, including simple Bayes, support vector machines (SVM) [19], bayesian networks [13], and C4.5 decision trees [6]. In the Twitter data set, the outcomes are superior. [9] shows that integrated learning is a workable approach for sentiment classification using five base classifiers. Most integration techniques used today are made for weak classifiers, as robust classifiers have already demonstrated superior performance in emotion classification tasks. As a result, applying robust classifiers to integrated learning and maximizing their benefits has emerged as a current area of intense research interest. [22] a sequential three-way decision (S3WC) cost-sensitive integration model for robust classifiers is suggested. Splitting the target into positive, negative, and boundary zones lowers the misclassification cost, and, as a result, the total cost is less than that of other integration combining techniques, like majority voting, weighted average, etc. However, isomorphic bases are not used in this procedure. However, the approach was not extended for experiments utilizing homogeneous and heterogeneous base classifiers. [4] The classification effect on the category fuzzy text is improved with the help of the three-way decision (3WD) method, which integrates CNN models and conventional machine learning techniques to classify the text twice.

In conclusion, the shortcomings of current multivariate sentiment classification models based on integrated learning include poor performance on short text classification, high similarity of base classifiers, and failure to utilize each classifier's advantages fully. In this paper, we argue that five aspects can be used to assess the performance of the integrated learning algorithm: (1) How many different contexts can be learned, and whether the model can combine them; (2) If the attention mechanism in the model is present and if it is capable of focusing on highly feature vocabulary; (3) Whether or not the model can retain vocabulary and sequential information; (4) Whether there is a relationship between the base classifiers' weights and the classifiers' relationships; (5) Whether it can address the multivariate sentiment dataset's category imbalance issue. These five factors serve as the foundation for the model-building work in this paper.

3 MODEL INTRODUCTION

3.1 Overview of the Model

Our paper presents an integrated learning model that utilizes heterogeneous classifiers. To achieve this, we first identify the fundamental position of base classifiers in the classification problem, and their effectiveness is a critical factor in determining the text classification outcome. We can enhance each other's strengths and achieve superior classification outcomes by simultaneously utilizing multiple classifiers that have already undergone classification training. The first sentiment classifier selected is the SelfAttention-BiLSTM model because it can extract text features with long-distance dependence, self-attention can calculate feature importance to remove semantic information from the sentence, and essential information can appear in any position within the sentence.CNN is the second text sentiment classifier; it learns the constructive and semantic features of the text by utilizing varying sizes of convolutional kernels to extract different text features. Ingeniously, the third text sentiment classifier uses the entropy value within and between text categories to filter the largest feature subwords. It then combines this method with simple Bayes to classify the text to be organized. It uses fuzzy integration to determine the weight coefficients of each classifier on each text to be classified to maximize the use of the classifiers' classification results. Random undersampling and random oversampling techniques are applied to the multivariate dataset to address the issue of category imbalance and optimize the model's classification performance. The first figure depicts the overall model.

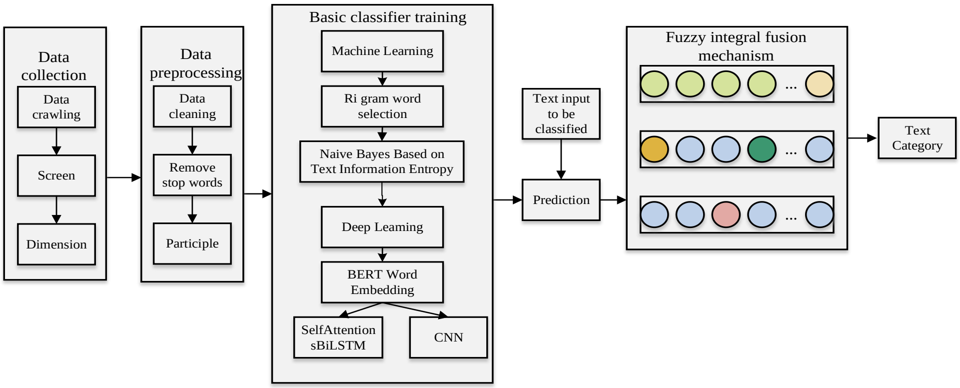

Figure 1: General structure of the model in this piece.

3.2 Self Attention ▪ BiLSTM Model

In the sentiment analysis task, the input sequence is temporal. By connecting the text's context and identifying the bidirectional semantic dependencies, BiLSTM can filter the data. To gather information from two opposing directions, BiLSTM builds two LSTM neural networks, which is more advantageous for recalling the intricate semantic expressions of the entire text and the lengthy dependencies between sentences. The BiLSTM model outperforms the LSTM model for classification in microblogging short text data, and both LSTM networks' pre- and post-BiLSTM structures are identical[2],[21],[1]. The forward computation of the memory unit of a single LSTM at time t governs the information transfer through the three types of gates that make up an LSTM: the input gate, forgetting gate, and output gate.

The BiLSTM model can increase the learning of textual reverse semantics by connecting the output vector $\overrightarrow{h_t}, \overleftarrow{h_t}$ of the LSTM model in the forward and reverse directions as the output of BiLSTM at time t, i.e., the semantic relation $B_t$of the context, as shown in Equation (1):

$$B_t = \left[\overrightarrow{h_t}, \overleftarrow{h_t}\right] \tag{1}$$

For feature classification, BiLSTM considers the text's temporal information and combines it with the context; however, it is challenging to discern the relative importance of various words and produce fine-grained sentiment features. The Bi-LSTM model alone may not accurately obtain feature information for long sentences with numerous complex words, such as microblog text. The SelfAttention-BiLSTM algorithm is selected as one of the base classifiers to address this problem since the SelfAttention mechanism can focus enough attention on the critical information in the text.

The nature of the self-attention mechanism function $attention(Q,K,V)$ can be described as a query of a series of key-value pairs (key-value) mapping, as shown in Equation (2) :

$$attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

Where Q, K, and V are vector forms and $Q \in \mathbb{R}^{n \times d_k}$, $K \in \mathbb{R}^{n \times d_k}$, $V \in \mathbb{R}^{n \times d_k}$, d_k denotes the dimension of Q, K .$\sqrt{d_k}$acts as a regulator, controlling the inner products of Q and K to be not too large. Self-attention mechanisms help readers understand how a text fits into a sequence. By adding these mechanisms at the word level, readers can also understand the significance of individual words in a text. First, feed the text word vectors into the BiLSTM. At each time step (O), the BiLSTM outputs the word vector sequence; at the final time step (H), it outputs the Hidden States. Equations (3) and (4) illustrate the calculation steps. The attention weight vector is set to Wattention, and the sentence vector (V) is computed using the attention mechanism.

$$W_{attention} = O \times H \tag{3}$$
$$V = O \times W \tag{4}$$

Following the acquisition of the sentence vector V, the fully connected layer and Softmax function output the likelihood that the text to be classified falls into each category.

3.3 Basic Textual Information Entropy-Based Bayesian Model

A probabilistic modeling algorithm based on statistics and the "full probability formula" is the Naive Bayes Classifier [8]. The algorithm for text classification views the relationships between words in a text as statistically independent, meaning that every dimension in each text's feature vector is independent of every other dimension. It's described as：

Let $x = \{a_1, a_2, \ldots, a_m\}$ be a text to be classified, each a_i is a feature attribute of x, the set of categories $C = \{y_1, y_2, \ldots, y_n\}$ Calculate $\left(P(y_1 \mid x), P(y_2 \mid x), \cdots, P(y_n \mid x)\right)$, so that $P(y_k \mid x) = max\left(P(y_1 \mid x), P(y_2 \mid x), \cdots, P(y_n \mid x)\right)$ Calculate. It is necessary first to identify a set of items to be classified with a known classification—referred to as the training set—and then compute the data from the training set to estimate the conditional probability of each feature attribute under each category. Equation (5) shows the computation using the Bayes theorem and the assumption that each feature attribute is conditionally independent.

$$P(y_i \mid x) = \frac{P(x|y_i)}{p(x)} \tag{5}$$

Equation (6) illustrates how the maximum likelihood category of the text to be classified can be obtained by introducing the significant a posteriori probability from Equation (5)：

$$P(x \mid y_i)P(y_i) = P(a_1 \mid y_i)P(a_2 \mid y_i) \cdots P(a_m \mid y_i)P(y_i) = P(y_i)\prod_{j=1}^{m} P\left(a_j \mid y_i\right) \tag{6}$$

In this paper, a simple-text information entropy-based Bayesian algorithm is proposed. Text information entropy is first used to extract the highly featured and distinguishable sentiment words from the identified text. These words are combined with the plain Bayesian model to improve the text classification effect.

The list of words with high differentiation ability for the critical incident corpus is obtained using text information entropy to determine a word's differentiation ability to text categories from two perspectives: intra-category and inter-category. Setting the text information entropy threshold yields the domain sentiment vocabulary; further judgment of the polarity and weighting of words with high distinguishing ability delivers the domain sentiment lexicon. The principle of text information entropy is based on information entropy. Information entropy is a quantitative index of the degree of uncertainty of information content in a system. Assuming that any probability event x occurs with the probability of generating n mutually independent results is$p(x_i)(i = 1,2, \cdots, n)$, then the information entropy of event x, H(x), can be calculated according to formula (7):

$$H(x) = -\sum p\left(x_i\right) lg\left(p(x_i)\right) \tag{7}$$

The mathematical expectation of the amount of information contained in a probabilistic event is known as information entropy, as stated in Equation (7). Higher degrees of uncertainty about whether an event will result in a particular outcome suggests that there is less information contained in the event, which causes the information entropy $H(x)$ to decrease and vice versa. The following is the definition of the information entropy principle about the computation of a text's information entropy within and between categories.

Suppose a corpus has K categories $(c_1, c_2, \ldots, c_k)$ of texts, each category contains N texts $(t_1, t_2, \ldots, t_n)$, and the word w occurs $(g_1, g_2, \ldots, g_k)$ times in each category and $(s_1, s_2, \ldots, s_N)$ times in N samples within the category.

Definition 1 The inter-category information entropy of words $H1\,(w)$ is mainly used to measure the ability of a word w to distinguish between categories among different categories. If the probability distribution of word w in each category is more uniform, the word cannot determine the category; if not, the ability to differentiate is more remarkable. The information entropy definition indicates that the information entropy value in Equation (8), which is determined in the following manner, should be inversely proportional to the size of information entropy between word categories:

$$H_1(w) = -\frac{1}{\sum_{k=1}^{K} \frac{g_k}{G} lg \frac{g_k}{G}} \tag{8}$$

Where $\frac{g_k}{G}$ denotes the probability distribution of word w across categories, and G is the number of occurrences of word w across categories, represented as $G = \sum_{k=1}^{K}$.

Definition 2: The intra-category information entropy of words $H_2(w)$ is mainly used to measure the size of category differentiation ability of word w from within the same category. A more uniform probability distribution of the word w within a category, in contrast to the inter-category information entropy of words, indicates that the word has a more remarkable ability to discriminate between categories; otherwise, it has a lesser ability to discriminate. The value of information entropy in Equation (9), which is determined as follows, is proportional to the magnitude of information entropy within word categories：

$$H_2(w) = -\sum_{n=1}^{N} \frac{s_n}{S} lg \frac{s_n}{S} \tag{9}$$

Where $\frac{s_n}{S}$ denotes the probability distribution of the word w in a category c_k, and S is the total number of occurrences of the word w within the category, denoted $S = \sum_{n=1}^{N} S_n$.

Definition 3 Textual information entropy $HE(w)$ Combining Eq. (8) and Eq. (9), the definition $HE(w)$ of the textual information entropy of word w is obtained. The vocabulary set with high category differentiation ability can be acquired according to the size of the textual information entropy value.

$$HE(w) = H_1(W) + H_2(W) \tag{10}$$

It is evident from the analyses above that the computation design and the definition of text information entropy are appropriate for determining how much n-gram subwords contribute to text category differentiation. The set of emotion words with high category differentiation ability can be obtained by filtering out the subwords that contribute little to category differentiation. Then, using simple Bayes, the probability of each category to which the text needs to be classified is jointly obtained, allowing the text's category to be determined. Fig. 2 displays the overall model framework.

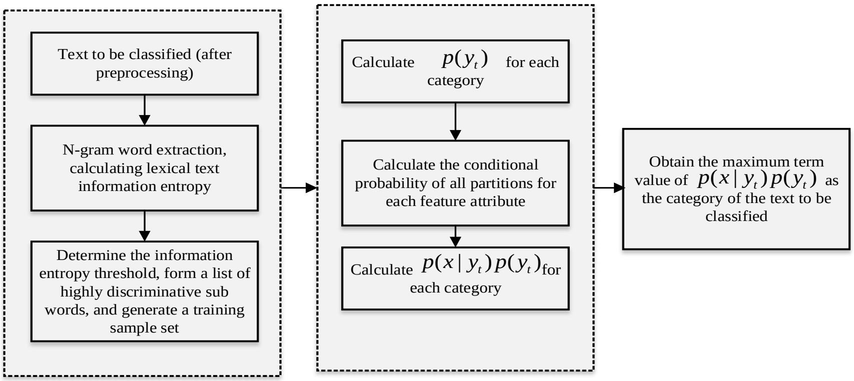

Figure 2: Basic Bayesian model based on entropy of textual information.

3.4 Converting Word Vector Space in Source and Target Languages

The Wasserstein distance is used in the paper instead of the JS scatter distance calculation primarily because of its more stable performance in hyperparameter selection. The model in this paper aims to minimize the JS scatter distance between the word vector distributions in the source and target languages[18],[17],[14]. Consequently, to calculate the separation between the source language word vector distribution and the generator, the Wasserstein distance is employed , and p_d is the word vector distribution in the target language, wto minimize$W_{asserstein}(p_s, \quad pdf)$. The language discriminator D is a binary classifier that takes $g(\boldsymbol{e}^d)$ as an input. It outputs a discriminator to determine whether it is from the target or the source language.

Back-propagation neural networks G and D are optimized through mutual game learning, generative adversarial training, iterative gradient updating, and Adam-based optimization. The iteration ends when generator G successfully converts the target language's word vector space to the source language's word vector space, and a trained discriminator D cannot distinguish between the target language and the source language in the word vector distribution that generator G converted.

The cross-entropy loss function defines the discriminator and generator loss functions. Equation (11) shows the loss function of the generator.

$$L_G = -\log\left(D\left(g(\boldsymbol{e}^d)\right)\right) \tag{11}$$

$D\left(g(\boldsymbol{e}^d)\right)$ denotes the probability that the discriminator will discriminate the word vector converted by the generator as the source language. Equation (12) illustrates the discriminator's loss function, distinguishing between the conversion of source and target language vectors.

$$L_D = -\log\big(D(\boldsymbol{e}^s)\big) - \log\left(1 - D\left(g(\boldsymbol{e}^d)\right)\right) \tag{12}$$

This paper implements language generator G using Deep Averaging Network (DAN) and CNN, respectively; DAN has a faster convergence time than CNN. A multilayer perceptron with several hidden layers of 1 is selected for the language discriminator D.

3.5 Cross-Language Emotion Discrimination

To distinguish the sentiment polarity of the output, the sentiment classifier is trained on the annotated text of the source language and then fed the text of the target language represented in the same semantic space. This is done using the source and target languages' word vector representations in the same semantic space.

The mean of all word vectors in the document is the same for the vectors of the text in the source and target languages. When both source and target language document vectors are present in the same space, they are consistently represented as $\widetilde{d_e}$, and $\widetilde{d_e}$is input to the Softmax layer of the sentiment classifier to output the predicted sentiment polarity.

For the sentiment classifier, the objective is to minimize the distance y between the sentiment prediction $\hat{p}(y \mid d_e)$ and the actual value of the target language document, with a loss function as shown in Equation (13).

$$L_p = -\frac{1}{N'}\sum_{k=1}^{N'}\sum_{y\in\{1,-1\}} y\, log\left(\hat{p}(y \mid d_e)\right) \tag{13}$$

In cross-language sentiment analysis, the sentiment polarity discrimination and the target language's word vector space transformation are handled as one cohesive unit. To maximize the target language's feature semantic extraction during training, the discriminative outputs of the language discriminator D and the sentiment classifier are simultaneously fed back to the language generator G. Equation (14) illustrates the definition of the language generator G's loss function, which is achieved by adjusting a hyperparameter λ to balance the effects of the two.

$$L = L_p + \lambda L_G \tag{14}$$

4 RESULTS

This experiment, which compared five different algorithms to confirm the efficacy of the proposed Senti_Aware model for cross-language sentiment analysis based on sentiment feature representations targeting other languages, used annotated Chinese text as the source language and Japanese as the target language. The five opposed algorithms are：

1. The upper bound method (hereafter referred to as Upper) for sentiment prediction in a single language works by first training a sentiment classification model with annotated document data in the target language (Chinese or Japanese) and then directly predicting the unlabeled documents in the target language using the trained model. The upper method chooses the Support Vector Machine (SVM) classification model. The upper method selects the Support Vector Machine (SVM) model as the classification model. SVM performs better in sentiment classification than algorithms like Random Forest and Simple Bayes[15],[20].

2. Google Machine Translation Engine is a machine translation tool that translates text from one language to another. Using the annotated source language corpus, the SVM sentiment classifier model is trained and predicts the translated source language text.

3. The Bi_W2V model has the same cross-lingual joint feature extraction and sentiment classification prediction modules as the Senti_Aware model, with identical parameter settings. The source and target language vectors are obtained using Word2Vec rather than the sentiment-aware source-language word vectors.

4.Word2Vec word vector representations of the source and target languages are replaced with randomly generated word vector representations in the Bi_random model, which has the same model and parameter settings as the Bi_W2V model.

5. CLCDSA model: [11] proposes an unsupervised Cross-Lingual Cross-Domain Sentiment Analysis (CLCDSA) model based on encoder-decoder, which uses a lot of unlabeled target

language data and annotated source language data to predict textual sentiment within or between languages. The CLCDSA model [11] predicts cross-lingual cross-domain or cross-language cross-domain textual sentiment by utilizing annotated source language data and a substantial amount of unlabeled target language data. In the experiment, CLCDSA uses the same dataset as the suggested model, with no division of the data into domains (e.g., belonging to DVDs, books, or music), and all texts in the dataset input as the same domain. The experimental parameters are the same as those found in the literature [7]: the language discriminator uses a 3-layer multilayer perceptron, the language model uses the AWD-LSTM model [5], and there are 150 hidden units in each layer. The dropout rate is set to 0.5.

4.1 Experimental Data Set

Three datasets with six different target languages were chosen for the experiment because no dataset can provide cross-language sentiment evaluation data in more than five languages. This will allow us to test the suggested model's generalization performance across various datasets and languages. Additionally, this is the first time a cross-lingual sentiment analysis study has chosen more than five languages for experimental evaluation.

Among them, the multilingual dataset of product reviews on Amazon.com [16] is where the Chinese and Japanese data in the source and target languages are sourced from. For each language, there are 10,000 labeled data points, representing the product ratings of 1, 2, 4, and 5 stars, with a higher rating denoted by a more considerable star value. Reviews with three stars or more are classified as positive in the experiment, while reviews with fewer stars are classified as harmful. The dataset parameters for each language are shown in Table 1.

Languages		*Annotation situation*	
		Marked	*Unmarked*
Source language	*Chinese*	*10000*	*105200*
Target language	*Japanese*	*10000*	*258470*

Table 1: Parameters of the experimental data set.

The target language's annotated data is only used to validate the effectiveness of the cross-language sentiment analysis; 10,000 pieces of data are used to be predicted for the Japanese language, and 80,000 pieces of data are used as the data to be expected for the Thai language. The Chinese source language uses 5,000 pieces of annotated data as training and 5,000 pieces of test data. The proposed model does not need to train on the target language's annotated data. A significant amount of unlabeled data is also included in Chinese and Japanese. These unlabeled and labeled data are fed into the CLCDSA model, which is trained to produce binary-coded files for Chinese and Japanese.

4.2 Experimental Parameter Settings

The experiment's primary parameters are as indicated in Table 2. The word embedding vector has dimensions of 60 and 200, a batch size of 50, an epoch of 30, and a learning rate 5×10-4. With the hyperparameter λ set to 0.01, the sentiment classifier's discriminative output influences the language generator more than the language discriminator's output. The value of the hyperparameter λ is 0.01.

Parameter	*Value*
Word vector dimension	*60/200*
Batch size	*60*
Epoch	*40*

Learning rate	*5×10^-4^*
λ	*0.01*
Dropout	*0.25*

Table 2: Experimental main parameter settings.

4.3 Experimental Evaluation Indicators

F_1-value (F_1-measure) the evaluation metric for sentiment classification prediction, is accuracy (Accuracy). The confusion matrix for the sentiment prediction binary classification problem is shown in Table 3.

Among them, TP indicates that the model predicts the number of positive outcomes and the actual sentiment label of the document is positive; FP suggests that the model predicts the number of positive results despite the actual label of the document being negative; FN indicates that the document's actual label is positive even though the model predicts the number of negative results; TN suggests that the document's accurate label is negative even though the model also indicates the number of negative consequences.

Discriminant results	*Actual category*	
	Active	*Negative*
Active	*TP*	*FP*
Negative	*FN*	*TN*

Table 3: Dichotomous confusion matrix.

According to Table 3, the accuracy rate is calculated as shown in (15)

$$\text{accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}} \tag{15}$$

Precision and recall are computed as shown in (16).

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{16}$$

The F_1 value is used as a combined evaluation metric for precision and recall and is calculated as shown in (17).

$$F_1 = \frac{2\times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{17}$$

4.4 Analysis of Experimental Results

Table 4 shows the cross-lingual sentiment classification prediction results with Chinese as the source language and Japanese as the target language. Table 4 displays the F_1 values and optimal prediction accuracies for every target language. The suboptimal values are indicated by underlining. The suggested model uses CNN and DAN as the linguistic generators for joint feature extraction in the source and target languages, respectively, and sets the dimension of word vectors in the experiments in Table 4 to 50 dimensions, with ɑ set at 0.8. The following sections will address the effects on the experimental results of changing the word vector dimension size, ɑ value, and whether to use the pre-trained BWE lexicon as the joint feature extractor.

The quality, quantity, and degree of data preprocessing all impact how well sentiment prediction performs across various cross-language pairs. Sentiment tendency can be expressed more clearly in specific language datasets, which is advantageous for sentiment prediction. In contrast, sentiment tendency can be expressed more subtly in other language datasets, which is unfavorable for sentiment prediction. This is demonstrated by the fact that different languages exhibit different sentiment prediction performance levels for the upper method. It is essential to perform both vertical comparison—that is, evaluating the performance advantages and disadvantages of various algorithms by comparing their cross-language prediction performance in the same language—and horizontal comparison—that is, assess the cross-language prediction performance of the same algorithm in various languages—when it comes to cross-language sentiment prediction.

Method	*Japanese*		*Chinese*	
	Accuracy	F_1 *value*	*Accuracy*	F_1 *value*
Upper method	*0.768*	*0.754*	*0.795*	*0.716*
Machine translation	*0.648*	*0.657*	*0.632*	*0.688*
Bi_random	*0.586*	*0.5582*	*0.568*	*0.678*
Bi_W2V	*0.536*	*0.589*	*0.267*	*0.658*
Senti_Aware (DAN)	*0.655*	*0.642*	*0.653*	*0.724*
Senti_Aware (CNN)	*0.665*	*0.712*	*0.714*	*0.745*
CLCDSA	*0.656*	*0.714*	*0.648*	*0.658*

Table 4: Chinese is the source language, and japanese is the target language for cross-language affective predictions.

The experimental results in Table 4 demonstrate that the proposed model performs better for cross-language sentiment analysis in Chinese and Japanese than the machine translation-based Bi_random and Bi_W2V methods. This confirms the efficacy of the cross-language text sentiment analysis method based on the representation of sentiment features. The upper method gives the cross-language sentiment prediction and classification performance upper bound that the model can attain. As can be seen, the suggested Senti_Aware (DAN) model's accuracy and F1 value in Japanese are 0.812 and 0.840, respectively, comparable to the Upper method's 0.854 and 0.856, respectively.

When comparing the various languages, it becomes more evident that Japanese has superior affective features. Chinese performs the best in the cross-language emotion classification of Chinese-Japanese language pairs, which aligns with the experimental expectation. Chinese and Japanese belong to the same language family, and while Chinese is closer to French in vocabulary, it is closer to Japanese in grammar and phonology.

Examine the effects of creating Word2Vec word vectors on cross-lingual sentiment analysis. Comparing the performance of Bi_random and Bi_W2V in various languages shows that Bi_W2V does not significantly outperform Bi_random. This suggests that, compared to randomly generated word vector representations, using Word2Vec to generate independent word vector spaces for the source and target languages does not significantly improve cross-language sentiment categorization prediction. Instead, it is more crucial to figure out how to map the two independent word vector spaces to the same semantic space. This demonstrates even more how vital it is to use deep learning models to learn and migrate two languages' word vector feature spaces when doing cross-lingual sentiment analysis.

The Bi_W2V algorithm outperforms the machine translation-based method regarding cross-language sentiment prediction in Chinese, Japanese, and French. Implementing machine translation in the experimental process necessitates splitting the dataset into multiple parts, translating each separately, and then merging them, which takes much time for translation and data processing. The existing translation engine API interface cannot support text translations longer than 5,000 words. The Bi_W2V algorithm performs better than this because it uses Word2Vec to generate word vectors and then performs cross-language sentiment extraction and prediction. This demonstrates that the deep learning-based approach has clear advantages over the machine translation-based approach for cross-lingual sentiment prediction and that this is where cross-lingual sentiment analysis is headed in the future.

Examine and contrast how various feature extraction networks impact the suggested model. The experiments employ CNN and DAN as feature extraction networks, respectively, and it is discovered that while the feature extraction network is changed, Senti_Aware's performance varies but remains essentially constant. Senti_Aware (DAN) and Senti_Aware (CNN) continue to have clear advantages over other comparison algorithms, demonstrating the model's usefulness for cross-lingual sentiment analysis tasks. According to the experimental results, the model's accuracy can increase by 0.6% to 1% by switching from the DAN feature extraction network to the CNN feature network. The average accuracy is also slightly higher when using the CNN feature network. Throughout the training phase, CNN has a slower convergence speed than DAN. For instance, the Senti_Aware based on the DAN feature extraction network takes approximately 6 minutes and 11 seconds. In comparison, the Senti_Aware based on the CNN feature extraction network takes about 12 minutes and 3 seconds, respectively, for the Thai dataset text prediction when the same dataset and the same experimental settings are run on a Tesla V100 GPU server with 31GB of memory. The CLCDSA model takes forty-two minutes and fifty seconds.

4.5 Examination of Variables Influencing Sentiment Analysis Across Languages

The variables influencing the cross-language sentiment analysis model are covered in this section, with particular attention paid to the effects of various alpha values, word vector dimensions, and the model's use of a pre-trained BWE lexicon. The target language for the analysis is Japanese, and the comparative analysis's findings on several other languages are comparable but aren't listed because of space restrictions.

4.5.1 The Impact of Alpha Values on Sentiment Analysis Across Languages

The representation ability of word embeddings will be affected by the size of the ɑ value during the training process of word embeddings that incorporate emotional semantics. The Japanese dataset is selected to investigate the effects of varying ɑ values with a step size of 0.1, as the classification effect performs optimally on this dataset. Fig. 3 displays the experimental results.

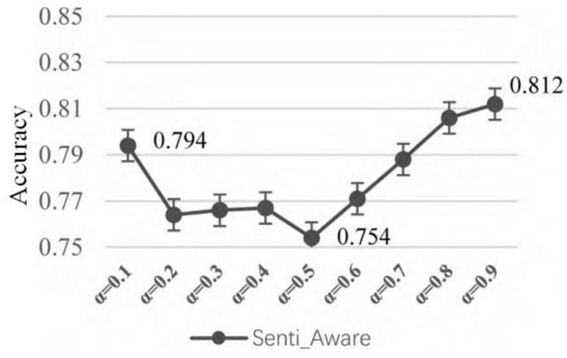

Figure 3: Differing ɑ the influence of values on sentiment classification across languages.

Figure 3 illustrates that the classification accuracy rate can reach 0.794 when ɑ is 0.1, at which point the weight of emotional information at the document level is at its highest; the classification accuracy rate gradually decreases when ɑ gradually increases; the accuracy rate is at its lowest when ɑ is 0.5. When this happens, the emotional information weights at the word and document levels are equal. When the value of ɑ continues to rise, the weight of dynamic information at the word level is greater than that of emotional information at the document level. When the value of ɑ is 0.9, the classification accuracy rate increases and reaches its maximum accuracy rate of 0.812.

The experimental findings demonstrate the superior independent supervision of word- and document-level sentiment information. However, when the two weights are close to one another, the sentiment information is used less frequently, which impacts the word embedding representation effect and lowers the accuracy of cross-linguistic sentiment classification.

4.5.2 The Impact of Word Vector Dimensionality on Sentiment Analysis Across Languages

To examine the impact of word vector dimensions on cross-lingual sentiment analysis, the experiments in this section set the word vector dimensions to 50, 100, and 150 dimensions, respectively. The dimension of word vectors has a specific effect on their ability to represent the semantics of words. Table 5 displays the experimental results. The experiment still uses the Japanese dataset, and DAN is chosen for the feature extraction network.

Method	*50 dimensions*		*100 dimensions*		*150 dimensions*	
	Accuracy	F_1 *value*	*Accuracy*	F_1 *value*	*Accuracy*	F_1 *value*
Upper method	*0.865*	*0.855*	-	-	-	-
Machine translation	*0.725*	*0.720*	-	-	-	-
Bi_random	*0.565*	*0.712*	*0.568*	*0.642*	*0.556*	*0.708*
Bi_W2V	*0.724*	*0.712*	*0.745*	*0.755*	*0.726*	*0.788*
Senti_Aware	*0.588*	*0.654*	*0.685*	*0.721*	*0.716*	*0.784*
CLCDSA	*0.813*	*0.845*	*0.745*	*0.756*	*0.788*	*0.685*

Table 5: Word vector dimensions' impact on sentiment analysis across languages.

The experimental results show that the improvement is most noticeable when the word vector dimension increases to 100 dimensions. At this point, the classification accuracy of the Bi_random method using only random word embedding can reach 0.568, and the F1 value is 0.642. It demonstrates that the larger dimension of word vectors can represent more information and have a better effect when using the Bi_random method with random initialization of text vectors. The highest F1 value of 0.745 and the highest accuracy of 0.716 are obtained when the word vector reaches 100 and 150 dimensions, respectively, when the Bi_W2V method is applied. The accuracy is marginally improved by increasing the word vector dimension.

Changing the word vector's dimension size does not seem to improve the classification accuracy for the Senti__Aware method; at 50 dimensions, it can already integrate emotional-semantic information well; the highest accuracy is 0.588, and the F1 value is 0.654, both of which are highly stable[12],[3].

4.6 Examination of Word-Vector Representations Visually

In this section, we compare the word vector representations obtained from the Senti_Aware and Word2Vec models using the visualization method to analyze from the linguistic and semantic point of view that the word vector representation based on the sentiment features of the source language can better take into account the information of the semantic and sentiment features of the words compared with Word2Vec. The word vector representations are reduced in dimensionality using the principal component analysis (PCA) method, which ultimately outputs them in the two-dimensional plane. The word vector representations obtained by Word2Vec or Senti_Aware are 50-dimensional high-dimensional vectors, which cannot be visualized in the 2D plane. When high-dimensional data is being dimensionality reduced, A is frequently used to extract and map the critical feature components of the high-dimensional data to the low-dimensional plane.

The 2D planar visualization outputs of two sets of words under the Word2Vec and Senti_Aware word vector representations, respectively, are displayed in Figures 4 and 5. A restricted set of words was selected as examples for the experiment so that the outcomes of the visualized representations could be observed with clarity. Based on PCA dimensionality reduction, each point in the figure represents the 2D plane embedding result of a word's high-dimensional word vectors; the closer two points are in the 2D plane, the closer their word vectors are. The word vector representation of Senti_Aware is on the right side of the figure, and that of Word2Vec is on the left.

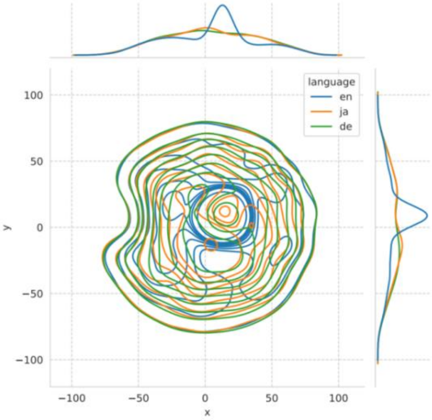

Figure 4: An illustration of senti-aware word vector representation using word2vec.

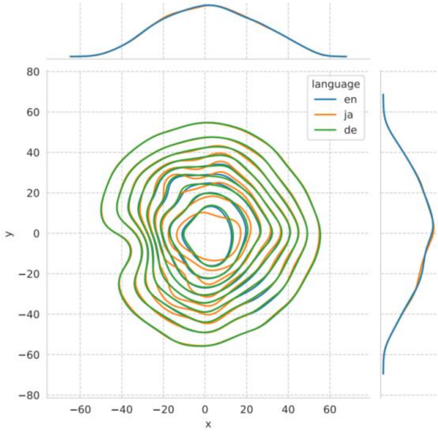

Figure 5: Example 2: Senti-aware word vector representation using word2vec.

A set of words, including "good, delicious, hate, bad, exciting, happy," and "beautiful," are visualized in a 2D plane in Figure 4. The terms "good, delicious, hate, bad, exciting, happy," and "beautiful" are rendered in two dimensions. These words' emotional polarity is apparent, and it is clear that Senti_Aware's word vector representation considers these emotional qualities and can discern between words with various emotional polarities. For instance, the negative sentiment polarity words "hate" and "bad" are grouped, whereas the positive words "good" and "delicious" are closer to one another. When comparing the Word2Vec word vector representations, it is evident that the terms "happy, bad," and "beautiful" are grouped together and ineffectively differentiated. Because the terms "happy," wrong," and "beautiful" are grouped, it is difficult to discern between them in terms of their emotional polarity.

A few words that are semantically closer are added, such as "dog, cat," and "bird," based on Fig. 4. A few words are randomly eliminated, and the visualization results are displayed in Fig. 5. As can be observed, the Word2Vec model performs better in semantic representation and can cluster semantically similar words such as "dog, cat," and "bird." Semantically similar words like "dog, cat," and "bird" can be combined, but words like "hate" and "exciting" still overlap. As a word with a negative emotion polarity, "hate" has a clear semantic distance from other words in the Senti_Aware word vector representation, which can still distinguish word emotion polarity.

5 CONCLUSION

This paper presents a novel cross-language text sentiment analysis approach based on word vector sentiment feature representations. This method solves the challenge of obtaining a BWE lexicon for deep learning-based cross-language sentiment analysis. It achieves cross-language sentiment polarity prediction from Chinese to other target languages. To get the word vector representation of the sentiment perception of the source language, the proposed method incorporates the sentiment supervisory information of the language into the cross-language sentiment analysis model. This allows the word vector representation to consider semantic and sentiment data, enhancing the sentiment prediction performance. The experiments use the unlabeled text in the target languages (Chinese and Japanese) as the source language for sentiment polarity prediction and the annotated text data in Chinese as the source language. Developing cross-cultural deep learning models for sentiment analysis between Chinese and Japanese is a complex yet rewarding endeavor. By bridging linguistic and cultural gaps, these models aim to capture sentimental nuances across two distinct languages accurately.

Fangyuan Su, https://orcid.org/0009-0008-6965-7074
Jinqing Liu, https://orcid.org/0009-0003-9578-9226
Yuliang Hao, https://orcid.org/0009-0003-1548-4511

REFERENCE

[1] Abdullah, T.; Ahmet, A.: Deep Learning in Sentiment Analysis: Recent Architectures, ACM Computing Surveys, 55(8), 2022, 1-37. https://doi.org/10.1145/3548772
[2] Balahur, A.; Turchi, M.: Comparative Experiments Using Supervised Learning and Machine Translation for Multilingual Sentiment Analysis, Computer Speech & Language, 28(1), 2014, 56-75. https://doi.org/10.1016/j.csl.2013.03.004
[3] Bashath, S.; Perera, N.; Tripathi, S.; Manjang, K.; Dehmer, M.; Streib, F. E.: A Data-Centric Review of Deep Transfer Learning with Applications to Text Data, Information Sciences, 585, 2022, 498-528. https://doi.org/10.1016/j.ins.2021.11.061
[4] Chan, J. Y. L.; Bea, K. T.; Leow, S. M. H.; Phoong, S. W.; Cheng, W. K.: State of the Art: A Review of Sentiment Analysis Based on Sequential Transfer Learning, Artificial Intelligence

Review, 56(1), 2023, 749-780. https://doi.org/10.1007/s10462-022-10183-8
[5] Darwish, K.; Habash, N.; Abbas, M.; Al-Khalifa, H.; Al-Natsheh, H. T.; Bouamor, H.; Mubarak, H.: A Panoramic Survey of Natural Language Processing in the Arab World, Communications of the ACM, 64(4), 2021, 72-81. https://doi.org/10.1145/3447735
[6] Hu, G.; Zhao, Q.: Multi-Model Fusion Framework Based on Multi-Input Cross-Language Emotional Speech Recognition, International Journal of Wireless and Mobile Computing, 20(1), 2021, 32-40. https://doi.org/10.1504/IJWMC.2021.10035664
[7] Jingchun, Z.; Dehuan, Z.; Weishi, Z.: Cross-View Enhancement Network for Underwater Images, Engineering Applications of Artificial Intelligence, 121, 2023, 105952. https://doi.org/10.1016/j.engappai.2023.105952
[8] Jingchun, Z.; Qian, L.; Qiuping, J.; Wenqi, R.; Kin-Man, L.; Weishi, Z.: Underwater Image Restoration Via Adaptive Dark Pixel Prior and Color Correction, International Journal of Computer Vision, 2023.
[9] Lo, S. L.; Cambria, E.; Chiong, R.; Cornforth, D.: Multilingual Sentiment Analysis: from Formal to Informal and Scarce Resource Languages, Artificial Intelligence Review, 48, 2017, 499-527. https://doi.org/10.1007/s10462-016-9508-4
[10] Mercha, E. M.; Benbrahim, H.: Machine Learning and Deep Learning for Sentiment Analysis Across Languages: A Survey, Neurocomputing, 531, 2023, 195-216. https://doi.org/10.1016/j.neucom.2023.02.015
[11] Nazir, A.; Rao, Y.; Wu, L.; Sun, L.: Issues and Challenges of Aspect-Based Sentiment Analysis: A Comprehensive Survey, IEEE Transactions on Affective Computing, 13(2), 2020, 845-863. https://doi.org/10.1109/TAFFC.2020.2970399
[12] Ntalampiras, S.: Toward Language-Agnostic Speech Emotion Recognition, Journal of the Audio Engineering Society, 68(1/2), 2020, 7-13. https://doi.org/10.17743/jaes.2019.0045
[13] Peng, S.; Cao, L.; Zhou, Y.; Ouyang, Z.; Yang, A.; Li, X.; Yu, S.: A Survey on Deep Learning for Textual Emotion Analysis in Social Networks, Digital Communications and Networks, 8(5), 2022, 745-762. https://doi.org/10.1016/j.dcan.2021.10.003
[14] Purwarianti, A.; Tsuchiya, M.; Nakagawa, S.: A Machine Learning Approach for an Indonesian-English Cross-Language Question Answering System, IEICE Transactions on Information and Systems, 90(11), 2007, 1841-1852. https://doi.org/10.1093/ietisy/e90-d.11.1841
[15] Vilares, D.; Alonso, M. A.; Gómez-Rodríguez, C.: Supervised Sentiment Analysis in Multilingual Environments, Information Processing & Management, 53(3), 2017, 595-607. https://doi.org/10.1016/j.ipm.2017.01.004
[16] Wan, X.: Bilingual Co-Training for Sentiment Classification of Chinese Product Reviews, Computational Linguistics, 37(3), 2011, 587-616. https://doi.org/10.1162/COLI_a_00061
[17] Wang, D.; Jing, B.; Lu, C.; Wu, J.; Liu, G.; Du, C.; Zhuang, F.: Coarse Alignment of Topic and Sentiment: A Unified Model for Cross-Lingual Sentiment Classification, IEEE Transactions on Neural Networks and Learning Systems, 32(2), 2020, 736-747. https://doi.org/10.1109/TNNLS.2020.2979225
[18] Wang, D.; Wu, J.; Yang, J.; Jing, B.; Zhang, W.; He, X.; Zhang, H.: Cross-Lingual Knowledge Transferring by Structural Correspondence and Space Transfer, IEEE Transactions on Cybernetics, 52(7), 2021, 6555-6566. https://doi.org/10.1109/TCYB.2021.3051005
[19] Xu, Y.; Cao, H.; Du, W.; Wang, W.: A Survey of Cross-Lingual Sentiment Analysis: Methodologies, Models and Evaluations, Data Science and Engineering, 7(3), 2022, 279-299. https://doi.org/10.1007/s41019-022-00187-3
[20] Yilmaz, S.; Toklu, S.: A Deep Learning Analysis on Question Classification Task Using Word2vec Representations, Neural Computing and Applications, 32, 2020, 2909-2928. https://doi.org/10.1007/s00521-020-04725-w
[21] Zhang, J.; Jin, H.: Method of Subjective Lexicon Creation for Chinese Sentiment

Analysis, Applied Mechanics and Materials, 34, 2010, 801-805. https://doi.org/10.4028/www.scientific.net/AMM.34-35.801
[22] Zvarevashe, K.; Olugbara, O. O.: Recognition of Cross-Language Acoustic Emotional Valence Using Stacked Ensemble Learning, Algorithms, 13(10), 2020, 246. https://doi.org/10.3390/a13100246