# An Intelligent Algorithm for Semantic Feature Analysis and Translation of English Texts

Sha Chen

*School of Foreign Languages, Nanchang Institute of Technology, Nanchang, Jiangxi 330044, China*
*E-mail: chens_cs@outlook.com*

Received 21 September 2024; Accepted 15 October 2024

## Abstract

Accurate and rapid translation is conducive to the cultural communication of different languages. This paper briefly introduces the long short-term memory (LSTM) algorithm. To enhance the performance of the LSTM algorithm, semantic features were introduced, and semantic similarity was used to screen the translations that are more in line with the semantics of the source text. Then, simulation experiments were conducted. The experiments first examined the effects of the quantity of hidden layer nodes and the type of activation function in LSTM on the translation performance. Then, the LSTM algorithm was compared with the recurrent neural network (RNN) and traditional LSTM algorithms.The proposed translation algorithm showed the best performance when there were 512 hidden layer nodes and the activation function was sigmoid, it performed better than the other two translation algorithms, and the obtained result was consistent with the semantic meaning of the source text and smooth.

**Keywords:** Semantic feature, semantic similarity, machine translation, LSTM.

*Journal of ICT Standardization, Vol. 12_3,* 271–282.
doi: 10.13052/jicts2245-800X.1232

## 1 Introduction

In the process of internationalization, the communication between different countries is inevitable, and English is a common language in the communication process [1]. Communication includes both oral and written communication. For individuals who are not native English speakers, acquiring and effectively utilizing English language skills requires a significant amount of time [2]. The emergence and development of intelligent algorithms have greatly increased their application fields, including the realm of natural language processing for English translation. Intelligent algorithms can greatly enhance the efficiency of English translation. As translation algorithms develop, it is no longer limited to the translation of words and phrases, but to the translation of sentence groups, paragraphs, chapters, and other long texts [3]. Although it further improves the efficiency of translation, it also increases difficulty. The increase of text length means that the contextual content of words or phrases in the text increases and the context is more complex. However, no matter what language, there will always be similar words, that is, words with similar semantics, whose features are usually close, which will lead to understanding errors in different contexts [4]. To enhance the precision of the translation algorithm, the semantic features of English text are also introduced into the intelligent translation algorithm. Sentence semantic matching serves as a fundamental investigation in addressing various natural language processing tasks, including but not limited to question answering and machine translation. Zhang et al. [5] proposed a deep feature fusion model and combined it with the deep learning architecture for the most popular sentence matching tasks. Wang et al. [6] designed a visual topic semantic enhanced translation model. introduced a translation model that incorporates visual topic semantics. This model leverages subject-specific images to construct semantic spaces across languages and modalities, facilitating the simultaneous integration of syntactic structures and semantic attributes. Wu et al. [7] developed a bilingual word embedding framework that leverages contextual information and found its superior performance in enhancing the quality of machine translation compared to previous approaches. Zheng et al. [8] proposed an approach of splicing word vector with character- level and word-level encoding vector and applied it in machine translation. The researcher found that the approach was effective in enhancing the translation performance of the translation model. Yu et al. [9] put forward a new neural machine translation architecture based on the similarity between Thai language and Lao and found that it was effective.

Lu et al. [10] improved the ability of the network to capture contextual features by using feature extraction, data preprocessing, and introducing an end-to-end dual-loop neural network and an attention mechanism. The ultimate goal was to eliminate English event pronouns and enhance the accuracy of machine translation. The aforementioned studies are all related to language translation and have adopted deep learning algorithms in the translation process. Moreover, they used semantic features to enhance translation performance. Similarly, this paper used deep learning algorithms for translating English and utilized semantic features to select the most appropriate translated text from candidate translations. This paper briefly introduces the long short-term memory (LSTM) algorithm. To improve the performance of the translation algorithm, semantic features were introduced, and semantic similarity was employed to screen the translations that are more in line with the semantics of the source text. Finally, simulation experiments were carried out. The limitation of this article lies in the fact that the translation algorithm only translates English into Chinese, without considering translating English into other languages. Therefore, a future research direction is to expand the translation scope of the algorithm so that it can translate English into a wider range of languages.

## 2 Long Short-Term Memory Algorithm

For machine translation, it is essentially the conversion of one kind of sequence into another kind of sequence, i.e., the processing of sequence data. LSTM, as an improvement of a recurrent neural network (RNN), is also suitable for treating sequence data. Compared with a RNN, it introduces a gating mechanism in the hidden layer, so that the algorithm can "forget" the historical data, thus reducing the calculation amount. The calculation formula is:

$$\begin{cases} f_t = f(W_f \cdot [h_{t-1}, x_t] + \theta_f) \\ i_t = f(W_i[h_{t-1}, x_t] + \theta_i) \\ \tilde{C}_t = \tanh(W_C[h_{t-1}, x_t] + \theta_C) \\ C_t = f_t \cdot C_{t-1} + i \cdot \tilde{C}_t \\ o_t = f(W_o[h_{t-1}, x_t] + \theta_o) \\ h_t = o_t \cdot \tanh(C_t) \end{cases}, \tag{1}$$

where $\tilde{C}_t$ and $C_t$ are the temporary and update states of the current memory unit [11], $h_t$ is the hidden state of the data input at the current moment, $x_t$ is

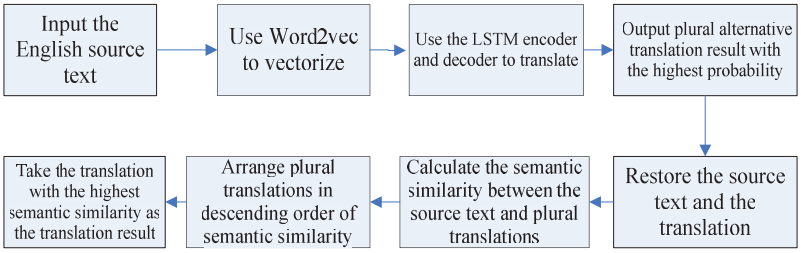


**Figure 1** The flow of the intelligent translation algorithm based on semantic features.

the current input text vector, $f_t$, $i_t$, and $o_t$ are the outputs of forgetting, input, and output gating units at the current moment, $W_f$, $W_i$, and $W_o$ are weights in the corresponding gating unit, $\theta_f$, $\theta_i$, and $\theta_o$ are biases in the corresponding gating unit.

## 3 Intelligent Translation Algorithm Based on Semantic Features

Machine translation algorithms are those that combine corpora and intelligent algorithms to convert one language into another language with the same or similar meaning. In essence, the intelligent algorithm transforms one set of vector sequences into another set with the aid of the corpus. In the process of converting the vector sequence, the computer and the algorithm themselves do not understand the meaning contained in the sequence, but only speculate the vector sequence with the highest sorting probability according to the law obtained by training and combining the prior data [12]. Because of this, translation results are more prone to errors when faced with words or phrases with similar semantics. While a human translation can be determined by context, a machine translation algorithm regularly outputs the most likely sequence. To improve the accuracy of machine translation, this paper incorporates semantic attributes.

When the algorithm is employed to translate English into Chinese, the translation result given by the algorithm is the text sequence with the maximum ranking probability predicted according to the law obtained by training [13]. In other words, the translation result itself does not consider semantic information, and the intelligent translation algorithm can also obtain the translation result with relatively low ranking probability when decoding the output translation result [14]. Considering that the ultimate goal of the translation algorithm is to convert the English source text into a semantically

equivalent Chinese translation, a translation that is semantically closer to the source text may exist in a translation result with a slightly lower ranking probability. To enhance the accuracy of the intelligent translation algorithm, this paper introduces semantic similarity from semantic features and uses semantic similarity to select translation results [15]. The semantic similarity is measured by the Jaccard coefficient, which is used to compare the similarity and difference between effective sample sets. The higher the value of the coefficient, the higher the degree of similarity. In this paper, this coefficient is used to select the most suitable translation from a set of alternative translations. However, traditional Jaccard coefficient only considers the number of co-occurring vocabulary and does not take into account their semantics, making it difficult to deal with polysemy. Therefore, word vectors are introduced to improve it. The process is shown in Figure 1.

(1) The English source text to be translated is inputted and pre-processed [16]. Preprocessing involves removing special characters from the source text and expanding abbreviations such as "it's" to "it is."
(2) The skipgram model in word to vector (Word2vec) is used to vectorize the source words [17].
(3) The word vector of the source text is input into the encoder and decoder for translation. The encoder transforms the word vector sequence into an intermediate sequence of vectors. Then, the decoder is employed to convert the intermediate sequence into the translation sequence. The encoder and decoder adopted in this paper use the LSTM algorithm.
(4) After decoding the intermediate vector sequence, the decoder obtains the distribution probability of each translation character in the sequence. Finally, the translated character with the highest probability is obtained by decoding the translation character distribution probability using the beam search algorithm [18]. When using this algorithm to decode the translation character sequence, it will also get the characters with a relatively lower probability, and these characters with a slightly lower probability will be used as alternative translation results.
(5) The source text and the plural alternative translation are restored. After part of speech tagging, prepositions and auxiliary words are removed from the source text. After removing the meaningless words such as prepositions and auxiliary words, the substantive words of the target text are restored to the source words through the bilingual corpus.
(6) To calculate the semantic similarity between the source text and the alternative translation, this paper uses the improved Jaccard coefficient [19] combined with the word vector to measure the semantic similarity.

The word vector obtained by Word2vec is a high-dimensional vector representation of words at the semantic level. The calculation formula of the improved Jaccard coefficient is:

$$Jaccard(T,S) = \frac{\sum_{t\in T}\sum_{s\in S}\cos(t_{emb}, s_{emb})}{\|T\| \cdot \|S\|}, \tag{2}$$

where $T$ represents the processed translation, $S$ is the source text after processing, $t$ and $s$ are one word in $T$ and $S$ respectively, $emb$ represents the word vector of the word, $\|T\|$ and $\|S\|$ are the length of the translation and source text respectively.

(7) The alternative translations are arranged in descending order of semantic similarity, and the one with the highest semantic similarity is selected as the output translation.

## 4 Simulation Experiment

### 4.1 Experimental Data

The data required comes from the English-Chinese Parallel Corpus of the University of Hong Kong, which contains many English-Chinese parallel texts, including documents in the fields of law, medicine, science and technology. Ten thousand sentences of parallel corpus were randomly selected as the training set, and then 5,000 sentences were randomly selected as the test set.

### 4.2 Experimental Setting

In the algorithm adopted in this paper, both encoder and decoder employed the LSTM algorithm, and the relevant parameters are shown in Table 1. The number of input layer nodes of the encoder depended on the dimension of word vector after vectorization of Word2vec, which was set as 200 here. In the output layer of the decoder, the beam search algorithm was utilized to convert the character probability distribution into translation, and the beam window size was set to 10. At the same time, the performance of the proposed algorithm was tested under different node quantities in the hidden layer and different activation functions. The node number in the hidden layer was set to 64, 128, 256, 512, and 1,024, respectively. The activation function was set as relu, sigmoid, and tahn, respectively.

As a contrast, two other translation algorithms were also tested. The other two algorithms also followed the encoder-decoder structure in the

**Table 1** Relevant parameters for the encoder and decoder

| Parameter | Encoder | Decoder |
|---|---|---|
| Input layer | Set the number of nodes to 200 | Set the number of nodes to 1,024 |
| Hidden layer | Two layers with 512 nodes per layer | Two layers with 512 nodes per layer |
| Output layer | 1,024 nodes | Using the beam search algorithm, the cluster window size is set to 10 |
| Activation function | Sigmoid function | Sigmoid function |
| Learning rate | 0.02 | 0.02 |
| Training sessions | 500 | |

main body. The encoder and decoder of one algorithm employed the RNN algorithm, while the other one adopts the LSTM algorithm as the encoder and decoder, but the difference was that when outputting the translation, only the sequence with the highest probability calculated by the beam search algorithm was chosen as the translation, without using semantic features to calculate semantic similarity and selecting the translation with the highest semantic similarity.

### 4.3 Evaluation Criteria

The word error rate of the translation was the first performance evaluation criteria of the algorithm [20], which is calculated as follows:

$$WER = \frac{X + Y + Z}{P} * 100\%, \tag{3}$$

where $X$ is the quantity of wrong words replaced, $Y$ is the quantity of wrong words deleted, $Z$ is the quantity of wrong words inserted, and $P$ is the quantity of all words in the test set. Moreover, the performance evaluation of the translation algorithm can be determined by comparing the resemblance between the translated output and the actual translation. The calculation formula is:

$$\begin{cases} BLEU = B \cdot \exp\left(\sum_{n=1}^{N} \omega_n \log p_n\right) \\ B = \begin{cases} 1 & c > r \\ \exp\left(1 - \frac{r}{c}\right) & c \leq r \end{cases} \end{cases}, \tag{4}$$

where $N$ is the maximum order of $n$-gram grammar, $\omega_n$ is the weight of $n$-gram grammar, $p_n$ is the phrase proportion of $n$-gram grammar, $B$ is the

**Table 2** The word error rate of the algorithm under different activation functions and different number of hidden layer nodes

| Number of hidden layer nodes | 64 | 128 | 256 | 512 | 1,024 |
|---|---|---|---|---|---|
| Relu | 4.3% | 2.9% | 1.7% | 1.4% | 1.9% |
| Sigmoid | 3.6% | 2.4% | 1.2% | 0.9% | 1.3% |
| Tahn | 4.4% | 3.0% | 1.9% | 1.5% | 2.1% |

penalty factor, $c$ is the quantity of words in the translation output by the machine, and $r$ is the quantity of identical words in the machine translated translation and the reference translation.

### 4.4 Test Results

This paper first used the word error rate to evaluate how well the algorithm performs with varying activation functions and hidden layer quantities, and the values are shown in Table 2. Under the same type of activation function, the word error rate of the algorithm decreased first as the number of hidden layer nodes increased, reached the minimum when there were 512 hidden layer nodes, and then increased. Under the same number of hidden layer nodes, the word error rate was the smallest when sigmoid was used as the activation function.

Table 3 presents the partial translation results of the three algorithms. The translated results obtained by the three translation algorithms all expressed the main meaning, but the translation provided by the RNN algorithm had a sense of incoherence when reading, and the translation provided by the LSTM algorithm only expressed similar semantics. The translation given by the LSTM algorithm combined with semantic features was basically the same as the reference translation.

Table 4 displays the translation performance of the three algorithms. The word error rate of the RNN algorithm was 4.4%, and the $BLEU$ of the translation was 21%. The word error rate of the LSTM algorithm was 1.5%, and the $BLEU$ of the translation was 48%. The word error rate of the LSTM algorithm combined with semantic features was 0.9%, and the $BLEU$ of the translation was 69%. It can be seen from the data in Table 4 that the LSTM algorithm combined with semantic features had the best performance, followed by the LSTM algorithm, and the RNN algorithm had the worst performance.

The reasons for the above results were analyzed. It can be seen that the RNN algorithm could process sequential data, but it induced “gradient

**Table 3** Partial translation results of the three algorithms

| Source text | This game is very interesting. | Did you finish yesterday's homework? | There is a high probability of rain tomorrow. Please prepare your umbrella. |
|---|---|---|---|
| Reference translation | 这款游戏很有趣。 | 昨天的作业完成了吗？ | 明天大概率会下雨，注意准备好雨伞。 |
| The translation of RNN | 游戏是有趣的。 | 你完成昨天的作业？ | 那里大概率下雨明天，请准备雨伞。 |
| The translation of LSTM | 这游戏好玩。 | 你完成了昨天的家庭作业吗？ | 高概率明天下雨，准备好你的伞。 |
| The translation of LSTM+semantic features | 这款游戏很有趣。 | 昨天的作业完成了吗？ | 明天大概率会下雨，注意准备好雨伞。 |

**Table 4** The translation performance of three algorithms

| | RNN | LSTM | LSTM+Semantic Features |
|---|---|---|---|
| Word error rate/% | 4.4 | 1.5 | 0.9 |
| *BLEU*/% | 21 | 48 | 69 |

explosion" or "gradient disappearance" in the face of long sequence data, thus affecting the accuracy of the algorithm. The LSTM algorithm introduced a gating mechanism for "forgetting" on the basis of RNN, which effectively solved the "gradient explosion" problem of long sequence data, so the translation performance was improved. The LSTM algorithm combined with semantic features further introduced the semantic similarity obtained by semantic features on the basis of the LSTM algorithm to screen multiple alternative translations to further enhance the translation performance.

## 5 Conclusions

This paper briefly introduces the LSTM algorithm for intelligent translation. To enhance the performance of the translation algorithm, semantic features were introduced, and semantic similarity was employed to screen the translations that are more in line with the semantics of the source text. Then, simulation experiments were conducted. The experiments first examined the effects of the quantity of hidden layer nodes and the type of activation function in the LSTM algorithm on the translation performance. Finally, the proposed algorithm was compared with the RNN and traditional LSTM

algorithms. The results are as follows. When 512 hidden layer nodes and the sigmoid activation function were used, the proposed translation algorithm had the best performance. In terms of translation results, compared with the reference translation, the translation of the three algorithms all expressed the main meaning. The translation of the RNN algorithm was not coherent enough, and the translation of the traditional LSTM algorithm only expressed similar semantics, while the translation of the LSTM algorithm combined with semantic features was basically consistent with the reference translation. The LSTM algorithm combined with semantic features had the best performance, followed by the LSTM translation algorithm, and the RNN algorithm had the worst performance.

## References

[1] Y. Zhao, M. Komachi, T. Kajiwara, and C. Chu, 'Region-attentive multimodal neural machine translation', Neurocomputing, 476, pp. 1–13, 2022.

[2] N. Q. Luong, L. Besacier, and B. Lecouteux, 'Towards Accurate Predictors of Word Quality for Machine Translation: Lessons Learned on French – English and English – Spanish Systems', Data Knowl. Eng., 96–97, pp. 32–42, 2015.

[3] M. Luo, L. He, M. Guo, H. Fei, L. Tian, H. Pu, and D. Zhang, 'Word-to-word Machine Translation: Bilateral Similarity Retrieval for Mitigating Hubness', IOP Conf. Ser. Mater. Sci. Eng., 533, pp. 1–9, 2019.

[4] B. Zhang, D. Xiong, J. Su, and Y. Qin, 'Alignment-Supervised Bidimensional Attention-Based Recursive Autoencoders for Bilingual Phrase Representation', IEEE T. Cybernetics, 50(2), pp. 503–513, 2018.

[5] X. Zhang, W. Lu, F. Li, X. Peng, and R. Zhang, 'Deep Feature Fusion Model for Sentence Semantic Matching', Comput. Mater. Con., 58(2), pp. 601–616, 2019.

[6] C. Wang, S. J. Cai, B. X. Shi, and Z. H. Chong, 'Visual Topic Semantic Enhanced Machine Translation for Multi-Modal Data Efficiency', J. Comput. Sci. Tech., 38(6), pp. 1223–1236, 2023.

[7] K. Wu, X. Wang, and A. T. Aw, 'Bilingual Word Embedding with Sentence Similarity Constraint for Machine Translation', International Conference on Asian Language Processing, pp. 119–122, 2017.

[8] P. Zheng, 'Multisensor Feature Fusion-Based Model for Business English Translation', Sci. Programming, 2022(Pt.3), pp. 3102337.1–3102337.10, 2022.

[9] Z. Yu, Y. Huang, and J. Guo, 'Improving Thai-Lao neural machine translation with similarity lexicon', J. Intell. Fuzzy Syst., 42(4), pp. 4005–4014, 2022.

[10] L. Qiu and W. Feng, 'Resolution of English Event Pronouns Based on Machine Learning', Mob. Inf. Syst., 2022(Pt.20), pp. 8560873.1–8560873.10, 2022.

[11] Q. Yang, L. Yu, S. Tian, and J. Song, 'Collaborative semantic representation network for metaphor detection', Appl. Soft Comput., 113(1), pp. 107911, 2021.

[12] C. Mi, L. Xie, and Y. Zhang, 'Improving data augmentation for low resource speech-to-text translation with diverse paraphrasing', Neural Networks, 148, pp. 194–205, 2022.

[13] X. Yang, T. Zhang, and C. Xu, 'Semantic Feature Mining for Video Event Understanding', ACM T. Multim. Comput., 12(4), pp. 1–22, 2016.

[14] J. Ari, S. Vrai, and S. Egvi, 'Dense Semantic Forecasting in Video by Joint Regression of Features and Feature Motion', IEEE T. Neur. Net. Lear., 34(9), pp. 6443–6455, 2023.

[15] H. Xi, 'The design of complex semantic machine translation model for foreign linguistics', Boletin Tecnico/Techn. Bull., 55(15), pp. 473–481., 2017

[16] M. Liu, L. Zhang, H. Hu, L. Nie, and J. Dai, 'A classification model for semantic entailment recognition with feature combination', Neurocomputing, 208(oct.5), pp. 127–135, 2016.

[17] K. Bansal, S. Malik, and H. Rohil, 'Multiclass Labelling For Bug Report Severity Prediction Using Semantic Analysis and Machine Learning', 2022 Fifth International Conference on Computational Intelligence and Communication Technologies (CCICT), pp. 275–277, 2022.

[18] C. Mi, L. Xie, and Y. Zhang, 'Improving data augmentation for low resource speech-to-text translation with diverse paraphrasing', Neural Networks, 148, pp. 194–205, 2022.

[19] T. Yoshioka, S. Karita, and T. Nakatani, 'Far-field speech recognition using CNN-DNN-HMM with convolution in time', IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 360–4364., 2015

[20] R. K. Chakrawarti, H. Mishra, and P. Bansal, 'Review of Machine Translation Techniques for Idea of Hindi to English Idiom Translation', Int. J. Comput. Intell. Res., 13(5(3)), pp. 1059–1071, 2017.

## Biography

**Sha Chen**, born in September 1982, has received a master's degree from Jiangxi University of Finance and Economics in June 2018. She is working at Nanchang Institute of Technology as an associate professor. She is interested in business English and intercultural communication.