



OPEN

# Design of intelligent module design for humanoid translation robot by combining the deep learning with blockchain technology

Fan Yang<sup>1,2✉</sup> & Jie Deng<sup>3</sup>

To accelerate the deep application of deep learning in text data processing, an English statistical translation system is established and applied to the question answering of humanoid robot. Firstly, the model of machine translation based on recursive neural network is implemented. A crawler system is established to collect English movie subtitle data. On this basis, an English subtitle translation system is designed. Then, combined with sentence embedding technology, the Particle Swarm Optimization (PSO) algorithm of meta-heuristic algorithm is adopted to locate the defects of translation software. A translation robot automatic question and answer interactive module is constructed. Additionally, the hybrid recommendation mechanism based on personalized learning is built using blockchain technology. Finally, the performance of translation model and software defect location model is evaluated. The results show that the Recurrent Neural Network (RNN) embedding algorithm has certain effect of word clustering. RNN embedded model has a strong ability to process short sentences. The strongest translated sentences are between 11 and 39 words long, while the weakest translated sentences are between 71 and 79 words long. Therefore, the model strengthen the processing of long sentences, especially character—level input. The average sentence length is much longer than word-level input. The model based on PSO algorithm shows good accuracy in different data sets. This model averages better performance on Tomcat, standard widget toolkits, and Java development tool datasets than other comparison methods. The average reciprocal rank and average accuracy of the weight combination of PSO algorithm are very high. Moreover, this method is greatly affected by the dimension of the word embedding model, and the 300-dimension word embedding model has the best effect. To sum up, this study proposes a good statistical translation model for humanoid robot English translation, which lays the foundation for intelligent interaction between humanoid robots.

Machine translation is a research field that combines artificial intelligence with natural language processing. The purpose is to realize the mutual translation between two natural languages and complete the equivalent conversion of information between different languages on the premise of retaining the meaning of the original text<sup>1</sup>. With the acceleration of globalization, the connection between users of different languages is getting closer and closer, and people's demand for bilingual interaction and conversion is stronger. In the era of underdeveloped computer technology, the traditional manual translation is inefficient. The emergence of automatic translation technology based on computer technology saves resource waste and improves translation efficiency<sup>2</sup>. Machine translation has become one of the hot topics in the field of applied technology. Machine translation research began in seventeenth century. At the beginning of the research, machine translation mainly adopts the rule-based method, and linguists input the conversion rules between two languages into computers<sup>3</sup>. However, this method is highly dependent on the quality of language rules, and its application is greatly limited<sup>4</sup>. Therefore, a corpus-based translation model, namely statistical machine translation, came into being, which can automatically learn language conversion rules from the corpus<sup>5</sup>. However, this translation method also has limitations, and cannot make good use of global features<sup>6</sup>. With the rapid development of modern science and technology, people pay more and more attention to robot education. In recent years, humanoid robots have gradually entered

<sup>1</sup>School of Foreign Languages, Fuzhou University of International Studies and Trade, Fuzhou 350202, China. <sup>2</sup>Graduate School, University of Perpetual Help System DALTA, 1740 Las Pinas, Philippines. <sup>3</sup>Rockchip Electronics Co., Ltd., Fuzhou 350007, China. ✉email: yangfan@fzfu.edu.cn

the classroom teaching. It has been recognized by colleges, teachers, students, and parents. The application of humanoid robot in classroom teaching has effectively improved the efficiency of management teaching, auxiliary teaching and presiding teaching. It can not only stimulate students' interest and enthusiasm in learning, but also promote the reform of the new curriculum to a certain extent, which is bound to play a powerful boosting role in improving the teaching effect and promoting students' all-round development. As a new and high technology, humanoid robot integrates artificial intelligence, mechanics, computer, communication, control, machinery, and other disciplines. The rational and scientific application of humanoid robot in education and teaching can not only create a comfortable learning environment for students, but also fully mobilize students' learning enthusiasm and initiative. This study is developed to explore the translation function of humanoid robot.

Blockchain is the combination of blocks with each other to form a blockchain database. These nodes of the blockchain abide by the exchange protocol together and join the transaction database network of the blockchain. In the traditional data centralized architecture, data is stored centrally, which has great risks caused by the architecture. Blockchain is an embodiment of the decentralization protocol determined according to distributed systems, and builds a distributed architecture with regional units. The information that is calculated and transmitted based on the value exchange protocol is copied to each unit of the entire network. The system determines the selection of information data content through accounting algorithm, and takes time stamp as an important reference, thus realizing distributed database storage based on blockchain. Blockchain is applied in the field of education, which can optimize the educational business process by building an educational trust system, strengthening the protection of intellectual property rights, and realizing efficient and low-cost educational resources. Decentralized education system can be constructed by using the characteristics of decentralization. Distributed storage is adopted to record trusted learning data, and educational intelligence of contract algorithm is developed, so as to build a new mode of network resources and platform operation. The application research of blockchain technology and intelligent learning robot is a frontier issue at home and abroad in recent years.

Deep learning technology can improve current machine translation<sup>7,8</sup>. Firstly, a machine translation framework based on deep learning is constructed, and its language model, word vector, output module, and word alignment are described in detail. Secondly, an English subtitle translation system is designed, and a translation robot automatic question and answer interaction module and a hybrid recommendation mechanism for personalized learning based on blockchain technology are constructed. Finally, sentence embedding and PSO are employed to locate the defects of translation software. The innovation lies in that this work proposes a software defect location method which combines sentence embedding technology and PSO algorithm. In addition, since there is no suitable data set, the system also generates Chinese and English subtitle corpus through crawler technology and corresponding preprocessing process. In short, this model provides a reference for the development of deep learning and blockchain technology in machine translation. It lays the foundation for accurate translation and intelligent interaction between humanoid translation robot and human.

## Material and methods

**Dataset and experiment establishment.** The data set based on the deep learning English subtitle translation part is acquired by crawling technology on the YYeTs website<sup>9,10</sup>. The specific method is expounded in the next section. The datasets used in PSO-based software maintenance come from three Java open-source projects, Standard Widget Toolkit (SWT), Java Development Tools (JDT), and Tomcat. The first 300 data are used in the training phase, and 40% of the three open-source projects are used as the validation set and 60% as the training set in the testing phase.

The specific parameters of the experimental equipment used are shown in Table 1.

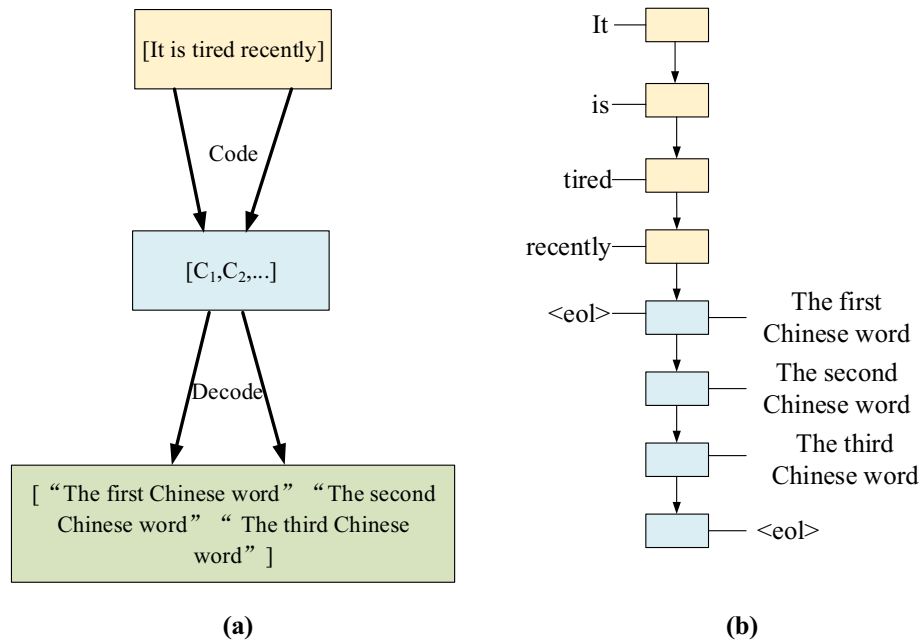
**Deep learning-based machine translation.** The application scenario of this deep learning is subtitle translation. First, the encoder-decoder framework shown in Fig. 1 is constructed<sup>11</sup>.

In Fig. 1a, the framework first completes the feature expression of the source language sequence, which is realized by encoder learning. Secondly, the corresponding target language sentences are generated by the decoder<sup>12</sup>. The training of the entire model under this framework does not require additional statistics of phrase occurrence probability, word frequency, etc., but only needs to provide corresponding bilingual pairs. The framework only needs to decode part of the network structure and concrete encoding. The order of the input and output sentences in Fig. 1a needs to be executed chronologically, which shows the Chinese translation of a four-word English sentence, where  $c$  is a fixed-length vector.

In Fig. 1b, each square represents a time, and the translation process of the model will predict the next word and will not stop until the end of the sentence is input. After the translated sentences are input in sequence, the model encodes a fixed-length vector, encodes all the information of the input sequence into the vector, and performs step-by-step prediction of the output sentence in chronological order<sup>13</sup>.

Item	CPU	Operating system	Graphics card
Deep learning-based English subtitle translation	Intel Core i7-4790K@4.00ghZ quad core	Windows 7 Ultimate 64-bit SPI	Nvidia GeForce GTX 970(4 GB/Colorful); Python: 2.7; Theano: 0.7; CUDA (Compute Unified Device Architecture): 7.0
PSO-based software maintenance	Inter Coretm I9-7900X; GeForce GTX 2080Ti	Ubuntu 16.04 64-bit	Pycharm 2019.2 IDE; Python: 3.7

**Table 1.** Experimental equipment parameters.



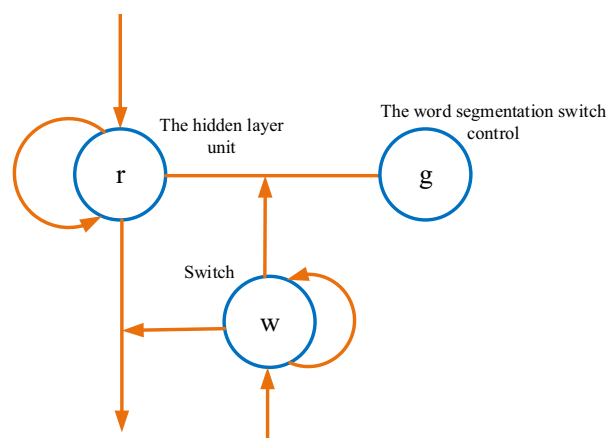
**Figure 1.** Encoder-decoder framework and time series [(a) is the encoder–decoder framework, (b) is the word-order time series].

The above is only the basic framework, and the next step of the network structure is designed. The first is the word vector generation area design. Word vectors can digitize natural language symbols<sup>14</sup>. Word vector generation methods include One-Hot-based methods and distributed representation-based methods<sup>15</sup>. Most of the methods based on distributed representation are at the word level. Of course, the current mainstream natural language processing also uses various word segmentation machines to first cut a sentence into words. In addition to this, however, this research also involves a training method of character data. To optimize the training of such data, a word vector generation method based on RNN is proposed (Fig. 2).

The word vector generation in Fig. 2 is based on each time unit, and it is also expanded in the chronological direction in the input sequence. The hidden layer unit is  $r$ , and the word segmentation switch control is  $g$ . Switch  $w$  clears  $g$  and  $r$  at the same time. The generation method of word vector is shown in Eqs. (1)–(5).

$$r_i = \tan r(W_h x_i + U_r r_{i-1}) \tag{1}$$

$$\underline{w}_i = \sigma(W_w x_i + U_w h_{i-1} + V_w w_{i-1}) \tag{2}$$



**Figure 2.** Word vector generation unit.

$$w_i = \begin{cases} 0, & \text{if } \underline{w_{i,1}} \geq \underline{w_{i,2}} \\ 1, & \text{if } \underline{w_{i,1}} < \underline{w_{i,2}} \end{cases} \quad (3)$$

$$g_i^x = w_i r_i \quad (4)$$

$$r_i = (1 - w_i) r_i \quad (5)$$

In Eqs. (1)–(5),  $\sigma$  is the sigmoid activation function. In Eq. (1),  $r_i$  is the output of the original hidden layer. In Eq. (2),  $w_i$  is used to determine whether  $r_i$  at time  $i$  is output. Equation (5) is used for further updating of  $r_i$ . All equations omit the bias term in order to reduce the confusion of the equation.

The encoder part uses a convolutional neural network, while the decoder part uses an RNN. For the encoder part, an output gate and two control gates are added based on the convolutional neural network. In the decoder language model, a vector containing the coding module information is added based on the RNN, and the vector belongs to a source language information. The word alignment module generation method adopts the “encoding–decoding” mode<sup>16</sup>. The word alignment step is added, and the encoding module information is integrated by adding a vector. The vector here is also expanded into two vectors according to the order of target sentence generation. The output module belongs to the correction module, which is embedded in the decoding part and needs to further output the translated words. Therefore, the output module must combine the previous output word again, decode the previous state of part of the hidden layer, and translate the required source language information.

### English subtitle translation system design

The used data belongs to the bilingual Chinese and English in the spoken language type, and the existing data is relatively small, so the crawler technology is adopted to collect the relevant data. The YYeTs website in the movie subtitle resource is used as the crawling object, and the subtitles of this website are more authoritative.

First, the data is located, and the pages that need to be crawled are manually observed. The subtitle web page is loaded, and the movie page link is extracted. Secondly, the Hyper Text Markup Language (HTML) source code of the individual movie subtitle page is analyzed, and the Chinese subtitle compressed file download link of the corresponding page is extracted. To ensure the complete extraction of the link information in the first two steps, it is necessary to carry out a step-by-step hierarchical screening of the required tags according to the HTML tags combined with Extensible Markup Language Path Language (XPath)<sup>17</sup>. Through the crawler program, all Chinese and English bilingual subtitle compressed packages on the YYeTs website are harvested, totalling 16,986. Due to the low download efficiency of the single-threaded crawler system, it takes a week to download all the data. Therefore, the multi-threaded parallel processing is performed. First, the original task is overwritten to get a subclass that inherits from the Thread class, and the thread pool is opened. The original task consists of four sub-tasks, namely, obtaining a list of movie links, a list of subtitle links, a list of compressed file links, and a letter compressed file. The four subtasks run in multiple threads, the thread pool size is 20, and a single thread waits for processing between the subtasks. The time for the crawler system to obtain data after multi-threading is one day.

Data pre-processing mainly refers to the structured processing of data. First, the compressed files with the suffixes of zip and Roshal A Rchive (RAR) in the downloaded files are decompressed, and two types of subtitle files, Advanced Sub Station Alpha (ASSA) and Sub Rip Text (SRT) are obtained<sup>18</sup>. These two kinds of files are small, and the modification to them can be handled by using the system’s own Notepad. Finally, the bilingual pairs with chronological order are harvested, monolingual files are filtered, and the selected documents are transcoded to 8-bit Unicode Transformation Format (UTF-8). A total of 13,717,370 Chinese and English bilingual corpora are obtained and stored in the *zh.pkl* file and the *en.pkl* file. Due to the different serialization requirements at different levels, data pre-processing adopts two processing methods, namely word-level and character-level.

The data structure processing adopts the current common processing method. The word-level dictionary retains the first 30,000 words with high frequency, and the number of character levels is small, which is used directly. For the expression of text, the One-Hot encoding is adopted. To optimize the training efficiency, the task processing is optimized by sorting sentences by sentence length at regular intervals, and a small block is synthesized nearby for task processing<sup>19</sup>. The above data is compressed and stored by the Hierarchical Data Format (HDF) algorithm.

The parameters in model training are set to 2000-word lengths in the language model. The word embedding module is finally compressed into a low-dimensional vector with a dimension of 1000. The word-aligned part vector size is 2000, and the length of the output module is 1000. During the experiment, a series of weight matrices in the recursive mode are initialized. The random orthogonal matrix can be randomly generated first and then obtained by the singular value decomposition method<sup>20</sup>. The matrices in the alignment module are randomly sampled using a Gaussian distribution with a variance of 0.0012 and a mean of 0. The bias vector of the weight matrix is 0. The weight matrix is randomly sampled from a Gaussian distribution with a variance of 0.0012 and a mean of 0.

The parameter training adopts the Mini-Batch method of the Stochastic Gradient Descent (SGD) algorithm. One gradient update of this method is shown in Eqs. (6) and (7).

$$x_{t+1} = x_t + \Delta x_t \quad (6)$$

$$\Delta x_t = -\delta \cdot \varepsilon_t \quad (7)$$

In Eq. (6),  $x$  is the parameter to be trained, and  $t$  is the time sequence. In Eq. (7),  $\delta$  represents the gradient, and  $\varepsilon$  is the learning rate. The automatic adjustment of the parameter learning rate in the experimental part is carried out by the AdaDelta algorithm. The calculations are shown in Eqs. (8)–(10).

$$\Delta x_t = - \frac{\text{RMS}[\Delta x]_{t-1}}{\text{RMS}[\varepsilon]_t} \cdot g_t \tag{8}$$

$$\text{RMS}[\varepsilon]_t = \sqrt{E[\varepsilon^2]_t + \epsilon} \tag{9}$$

$$E[\varepsilon^2]_t = \rho E[\varepsilon^2]_{t-1} + (1 - \rho) \varepsilon_t^2 \tag{10}$$

Equation (9) is the root mean square, and Eq. (10) is a similar approach to the momentum factor averaging algorithm.

The recursive part of the network is processed by the Backpropagation Through Time (BPTT) algorithm. The evaluation index of model translation quality is shown in Eq. (11).

$$\text{BLEU} = \xi \times \exp\left(\sum_{n=1}^N (w_n \log P_n)\right) \tag{11}$$

BLEU in Eq. (11) is Bilingual Evaluation Understudy.  $w$  is the weight,  $\xi$  is the penalty factor, and  $P$  is the accuracy. Two million pairs of bilingual data are selected as experimental data, and 1.8 million pairs of training set and 5000 sentences of test set are generated. The sequence length, model score, model translation comparison, etc. will be generated in the later part.

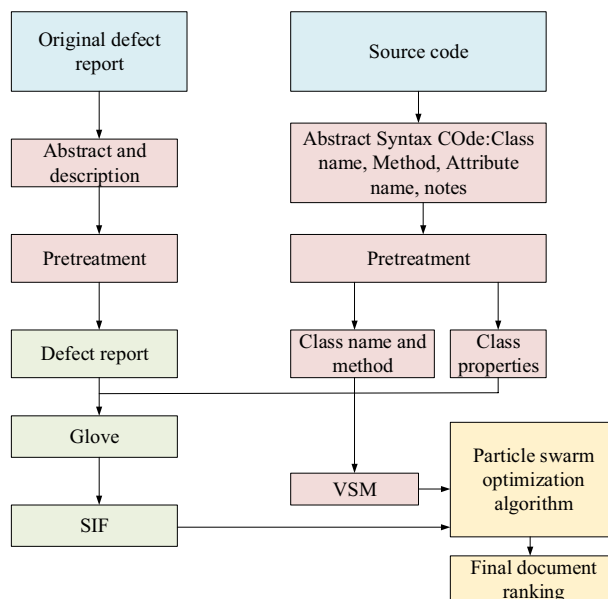
### Translation software maintenance of sentence embedding collaborative PSO

This section is maintained for meta-heuristic-based translation software. The structure of the software defect location method used is shown in Fig. 3.

Figure 3 shows that the software defect location process mainly includes extracting source code content with abstract syntax tree technology and preprocessing defect report with natural language technology. The semantic similarity between the source code and the software defect report is calculated, and the text information quantification is combined with the Vector Space Model (VSM) technology. The PSO algorithm is employed to sort the most weighted combination<sup>21</sup>.

The data pre-processing of the software defect location part includes the description and summary of the defect report. First, it creates an abstract syntax tree, tokenizes text with spaces, and removes numbers, punctuation, and stop words. Second, it splits the compound words. Finally, it stems all words, using tools in the Natural Language Toolkit (NLTK) package here.

Smooth Inverse Frequency (SIF) algorithm and word embedding technology are adopted to calculate the semantic similarity between software defect reports and source code files. The first is word vectorization, but the SIF model can only input vectorized words<sup>22</sup>. Descriptions, summaries, and source code files in software bug



**Figure 3.** Software defect location process.

reports are converted into vectors using Glove word embedding techniques<sup>23</sup>. The second is word probability calculation. In SIF, the probability of words in a sentence needs to be calculated, as shown in Eq. (12)<sup>24</sup>.

$$P(w|c_s) = \gamma p(w) + (1 - \gamma) \frac{\exp(\langle \tilde{c}_s, v_w \rangle)}{Z_{\tilde{c}_s}} \quad (12)$$

In Eq. (12),  $\gamma$  is the scalar hyperparameter,  $\tilde{c}_s$  is the vector representation of the sentence, and  $v_w$  is the word semantic vector.

The word embedding calculation is shown in Eqs. (13) and (14).

$$P(s|c_s) = \prod_{w \in s} P(w|c_s) = \prod_{w \in s} \left[ \gamma p(w) + (1 - \gamma) \frac{\exp(\langle \tilde{c}_s, v_w \rangle)}{Z_{\tilde{c}_s}} \right] \quad (13)$$

$$\operatorname{argmax}_{\tilde{c}_s} \sum_{w \in s} f_w(\tilde{c}_s) \propto \sum_{w \in s} \frac{a}{P(w)+a} v_w, a = \frac{1 - \gamma}{\gamma Z} \quad (14)$$

Equation (14) is the maximum likelihood estimate of the unit sphere in Eq. (13).

The SIF word weighting factor is set to 0.00001, the number of singular vectors is 1, and the dimension of the Glove model input into the SIF model is 300.

The surface lexical similarity is calculated using VSM, as shown in Eq. (15).

$$TF-IDF_{ij} = TF_{ij} * IDF_j \quad (15)$$

In Eq. (15),  $TF-IDF_{ij}$  is the weight of the  $j$ th word in the  $i$ th document,  $TF$  is the frequency, and  $n$  represents the number of documents, where  $IDF_j = \frac{n}{DF_j}$ .

The final obtained surface vocabulary similarity is shown in Eq. (16).

$$\operatorname{sim}(br, sr) = \max(\cos(\operatorname{pair}) | m_j \in sr) \quad (16)$$

In Eq. (16),  $br$  represents a software defect report, and  $sr$  represents a source code file.

The PSO algorithm-based optimal combination calculation of linear weights is shown in Eq. (17).

$$\operatorname{finalscore} = w_1 \cdot \operatorname{SIFScore} + w_2 \cdot \operatorname{VSMscore} + w_3 \cdot \operatorname{RecencyScore} + w_4 \cdot$$

$$\operatorname{FrequencyScore} + w_5 \cdot \operatorname{CollaborativeFilteringScore} \quad (17)$$

In Eq. (17),  $w_1$  is the similarity weight calculated by the SIF model,  $w_2$  is the cosine similarity obtained by the term frequency-inverse document frequency (TF-IDF) model,  $w_3$  is the defect repair record weight,  $w_4$  is the defect repair frequency weight, and  $w_5$  is the collaborative filtering score weight.  $\operatorname{SIFScore}$  refers to the score using the SIF source code file.  $\operatorname{VSMscore}$  refers to the score using the TF-IDF model.  $\operatorname{RecencyScore}$  refers to the defect repair score.  $\operatorname{FrequencyScore}$  refers to the defect repair frequency score.  $\operatorname{CollaborativeFilteringScore}$  refers to the collaborative filtering scores. The weight is calculated using the PSO algorithm, and the calculation of particle position and velocity is shown in Eq. (18).

$$v_{id}^{t+1} = w v_{id}^t + c_1 \operatorname{rand}(p_{id}^t - x_{id}^t) + c_2 \operatorname{Rand}(p_{gd}^t - x_{id}^t) \quad (18)$$

In Eq. (18),  $w$  is the inertia weight, and  $c$  is the learning factor.  $\operatorname{rand}$  is a random value, where,  $x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$ .

The evaluation metrics used in this experimental evaluation are Accuracy@k, Mean Average Precision (MAP), and Mean Reciprocal Rank (MRR). Accuracy@k is the percentage of software bug reports that match at least one correct suggestion in the top  $k$  files. The specific calculations of the MAP and MRR are shown in Eq. (19) and Eq. (20), respectively.

$$\operatorname{MAP} = \frac{1}{N} \sum_{i=1}^N \operatorname{AvgP}(i) \quad (19)$$

In Eq. (19),  $N$  is the number of source code files, where,  $\operatorname{AvgP}(i) = \sum_{j=1}^M \frac{Q(j) * \operatorname{bool}(j)}{B(i)}$ .  $M$  is the number of software defect reports.  $B$  is the number of defect files included in the software defect report.  $\operatorname{bool}(j)$  indicates whether the source code file ranked  $j$  is a defect file.  $Q$  indicates the number of source files related to the defect report in the source file.

$$\operatorname{MRR} = \frac{1}{N} \sum_{i=1}^N \frac{1}{\operatorname{first}_i} \quad (20)$$

In Eq. (20),  $N$  is the number of defect reports, and  $\operatorname{first}_i$  is the file location where the first defect is located.

The parameters for this experiment are set as the population size of 20, the inertia weight of 0.8, the learning factors of 1 and 2, and the maximum number of iterations of 1000.

The results in the following part include the performance of the proposed method and the benchmark method on different data sets. The compared models include Deep Neural Networks Loc (DNNLoc), BugLocator, and



Learning-to-Rank. Validity test of PSO algorithm and comparison of embedding effect of words of different dimensions are also performed.

### Construction of automatic question answering interactive module for the translation robot

On the basis of the above accurate translation, an intelligent interaction module of the robot is designed, and a humanoid translation robot with translation and interaction functions is obtained. Convolutional neural network has been applied in classification, pattern recognition and other fields with its powerful learning ability. The method achieves feature extraction based on convolution, and classifies on this basis to achieve higher classification accuracy. The network is consisted of convolution, pooling, and full connection layers. Given that the two sentences will be analyzed and classified, the structure of parallel convolutional neural network is adopted. ReLU function is taken as activation function. To avoid the overfitting problem, the dropout method is introduced to modify the convolutional neural network. AdaGrad adaptive learning algorithm is selected for learning rate. This module can realize the intelligent learning, classification, and answer of dialogues.

Although convolutional neural network can realize the extraction of deep features, there are still defects in the application, especially in the case of massive statements, and it is difficult to retain the underlying semantic information. To solve these problems, attention mechanism is introduced. Traditional Neural Machine Translation (NMT) includes encoder and decoder, which can get target language through decoding. Combined with attention mechanism, it is introduced into NMT neural machine translation. The whole system is divided into multiple modules. Each module has different functions and structures and achieves the overall function depending on the connection and coordination between each part. Question and answer engine is mainly used to process the input information. The semantic matching part is the algorithm based on the convolutional neural network constructed in this study.

Humanoid robot is a kind of high technology, which integrates artificial intelligence, mechanics, computer, communication, control, machinery, and other disciplines. The reasonable and scientific application of humanoid robot in the process of education and teaching can not only create a comfortable learning environment for students, but also fully mobilize their learning enthusiasm and initiative. At present, the teaching of humanoid robots is mostly carried out in the form of groups, which requires the members of the groups to work together and use the collective strength to solve problems that are difficult for individuals. By studying in this form, students can easily improve their teamwork ability. The reasonable application of humanoid robot in education and teaching can popularize more robot knowledge to students. Robot classroom teaching has become the core content of information technology education in colleges and universities. Robots are no longer mysterious, but have entered students' daily study and life, which is of great help to students' sustainable development. Therefore, the automatic question-and-answer interactive module of translation robot in this work is added to the production of humanoid robots to increase their teaching ability.

### Hybrid recommendation mechanism based on personalized learning

The system of blockchain technology and intelligent humanoid learning robot presents a distributed structure. The nodes of this structure include family, school, community, venue, training class, social welfare home, public place, and travel. The central node is family and school. The main functions of the node are block creation, recording, transmission, and encryption. The family node records its learning process in the Block. After confirmation, the system sends the learning record chain to all nodes. Over and over again, every node in the system keeps a uniform record. The intelligent learning service system records the learning process by creating blocks. The pre-recorded contents of each data block are magic number, block size mark, data block header information, learning activity count, and learning activity information. Where, the calculated value of the data block header information is the reference target number of the calculated value of the next new block, and the last learning activity details records all learning activities in this block.

The hybrid recommendation mechanism based on personalized learning is the fusion of collaborative filtering recommendation technology, content-based recommendation technology and knowledge-based recommendation technology. Collaborative filtering recommendation uses the historical behaviors or opinions of existing learners to predict the most recommended learning resources for current learners. The input information is the evaluation data of the resource rating matrix, and the output data is the predicted value of the learner's preference for this resource and a list of multiple recommended resources.

Content-based recommendation is a classic dynamic recommendation idea, which matches resource characteristics and learner preferences. The recommendation focus mainly includes two aspects: on the one hand, dynamic generation based on historical records. Although this recommendation method relies on additional information about resources and learner preferences, it has the advantage of generating recommendation lists in the absence of learner group information. Content-based recommendation mainly includes content similarity-based retrieval and probability-based methods, while knowledge-based recommendation is a recommendation method formed by clarifying recommendation rules in the form or basis of the similarity between learners' needs and resources. Including constraint-based recommendation and instance-based recommendation. The difference lies in that the constraint-based recommender system relies on a well-defined set of recommendation rules, and it searches for the set of resources to be recommended among all resource sets that meet the recommendation rules.

In addition, content-based recommendations include a big data approach, supported by large-scale learning services and blockchain technology. For example, different learning state transition graphs are formed through the matching of relevant learner trajectories, or the dynamic programming method of closed-loop dynamic learning data information collection is carried out through the self-quantization learning process of big data. The characteristics of learning resources pushed by the system are accuracy, comprehensiveness, and personalization.

Learners can quickly obtain this information, thus saving the time of searching and finding materials on the Internet. Personalized recommendation is the core of learning resource push. The system obtains the learner's specific learning situation from the learning management database, so as to screen out the relevant information that best matches the learner's knowledge level.

## Results and discussion

**Result analysis of deep learning-based English subtitle translation system.** The statistical results of different language lengths at the word level and character level are shown in Table 2.

Table 2 shows that the statistical results are divided into training data and test data. In terms of word-level and character-level comparisons, character-level modeling is more challenging.

The scoring results of different translation models are shown in Figs. 4 and 5.

The comparison model shown in Fig. 4 is the Moses model, and Recurrent Neural Network (RNN)-search is the current optimal neural network-based method. P1-P4 in the abscissa show the matching accuracy. In Fig. 4, the difference of scores of the three models is relatively small, and the score of the RNN-model is basically between the Moses model and the RNN-search model. The word-level RNN-model scores does not improve. See the "drawing data" table in the supplementary information for the drawing data of all figures in this paper.

In Fig. 5, there are many contrasting models due to the challenging character-level modeling. The asterisk represents the improved word vector generation algorithm of RNN, and RNN-embed is an RNN-based word embedding model. The curve in Fig. 5 shows that RNN-embed has superior performance in the rest of the evaluation results except for the slightly lower accuracy at point P3. The RNN-embed algorithm has a certain word clustering effect.

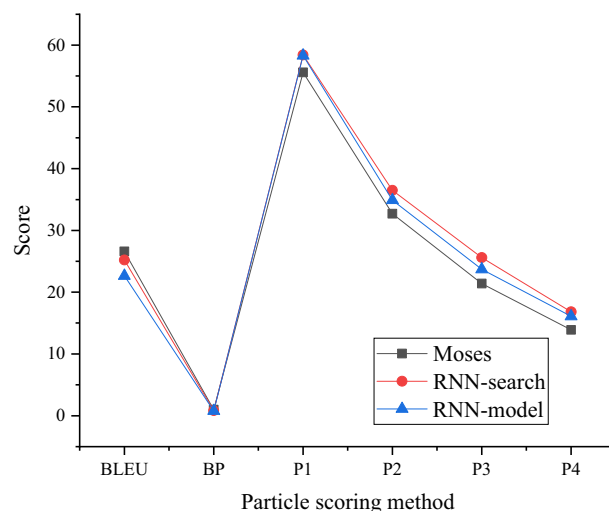
Table 3 shows the parameter tuning process of the RNN-embed model in Figs. 4 and 5.

Table 3 shows that the RNN-embed model has been significantly improved after the fourth parameter adjustment. At that time, the model is not overfitted, the results of the training model are improved, and the single process is slow.

Based on the previous content, the performance of the RNN-search model and the RNN-embed model is relatively better, so the actual translation capabilities of the two models are compared, and the comparison translation method also includes Baidu translation. Since the data used are movie subtitles, there are certain requirements for the spoken translation of the model. Examples of translations for different translation methods are shown in Table 4.

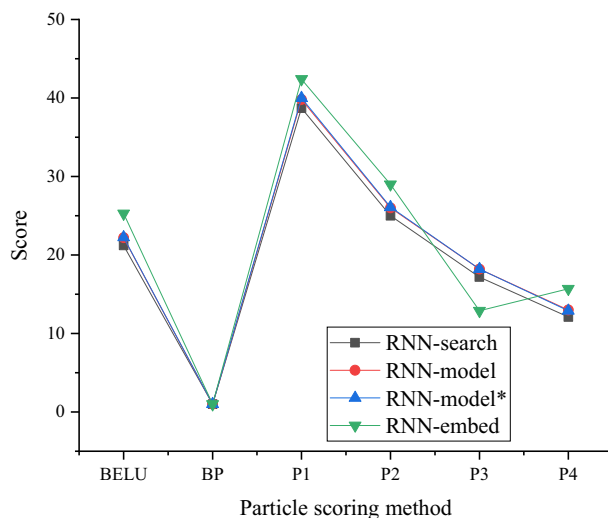
Length	Word level		Character level	
	Train/en	Test/zh	Train/en	Test/zh
1–9	1,270,522	1,366,950	97,169	873,449
10–19	520,850	432,225	256,970	904,566
20–29	4921	820	417,398	21,902
30–39	3137	5	456,448	78
40–49	541	–	339,007	3
50–59	29	–	181,520	2

**Table 2.** Language length statistics.



**Figure 4.** Scoring results of different word-level translation models.





**Figure 5.** Scoring results of different character-level translation models.

Parameter type	Number of parameter adjustments			
	1	2	3	4
Longest sentence threshold	50	80	100	80
Dim	1000	1000	2000	2000
Rank_n_approx	620	1000	1000	1000
Pn	39.9/26.4/18.9/14.0	40.8/27.3/19.7/14.7	40.7/27.4/19.8/14.8	42.4/29.0/21.2/15.7

**Table 3.** The parameter adjustment process of the combined model.

The original text	May I have a look at this picture?	I love you
RNN-search	May I see this photo?	I like you
RNN-embed	May I have a look at this picture?	I love you so much
Baidu translator	Can I see this photo?	I love you
Reference	May I see this photo?	I love you

**Table 4.** Translation examples of different translation methods.

In Table 4, the example sentences are common colloquial phrases, the training data of the two models is at the character level, and the training process of Baidu Translate is at the word level. Due to the addition of word embedding improvements, the RNN-embed model learns a more colloquial intonation.

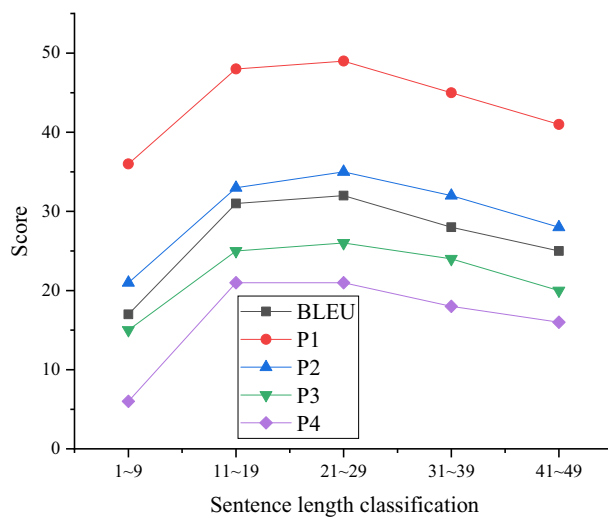
Figures 6 and 7 are comparisons of test results classified regarding sentence length. Among them, the average length of character-level data is 32, and the average length of word-level data is 7.

In Figs. 6 and 7, the curves show a clear downward trend, indicating that the RNN-embed model has a strong ability to process short sentences. The sentences with the strongest translation outcome are between 11 and 39 words. The sentences with the weakest translation outcome are 71–79 words in length. Hence, the model needs to strengthen the processing of long dramas, especially character-level input, and the average sentence length is much larger than that of word-level input.

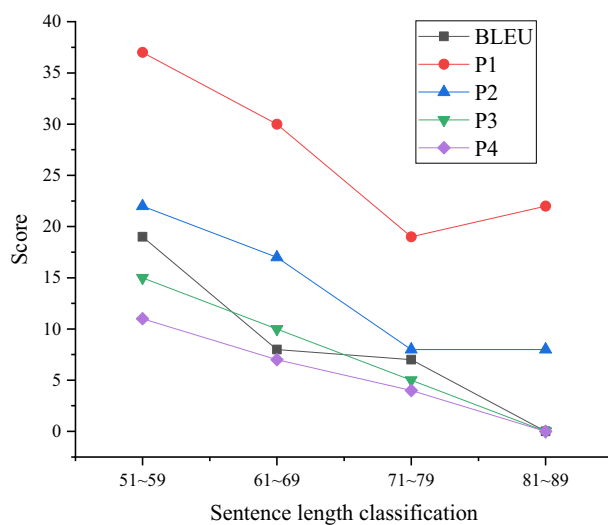
**Analysis of maintenance results of PSO-based translation software.** The Accuracy@k metric results of different models on the Tomcat dataset are shown in Fig. 8.

Figures 8 and 9 show that the proposed method shows an excellent accuracy on both the Tomcat dataset and the JDT dataset, which is better than other models. In Fig. 9, when the number of source code files  $k$  is 0, the accuracy of the proposed method is 34%. When there are 15 source code files, the accuracy reaches 73%. The BugLocator model has the lowest overall accuracy. The accuracy of the DNNLoc model and the Learning-to-Rank model is good. In Fig. 10, the average accuracy of the proposed method is basically the same as that of the Learning-to-Rank model.

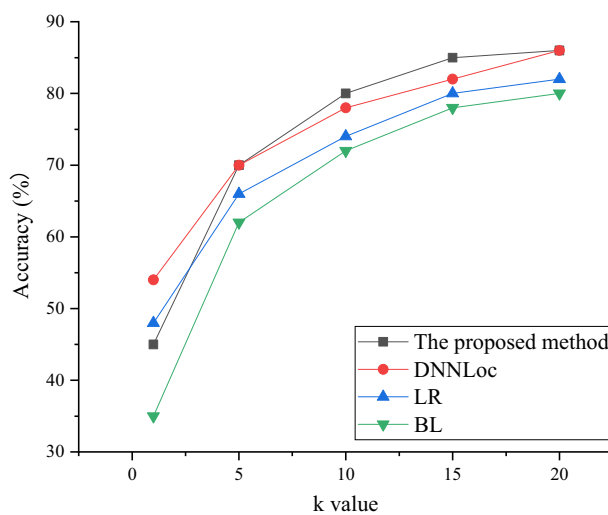
The MRR and MAP results on different datasets are shown in Figs. 11 and 12.



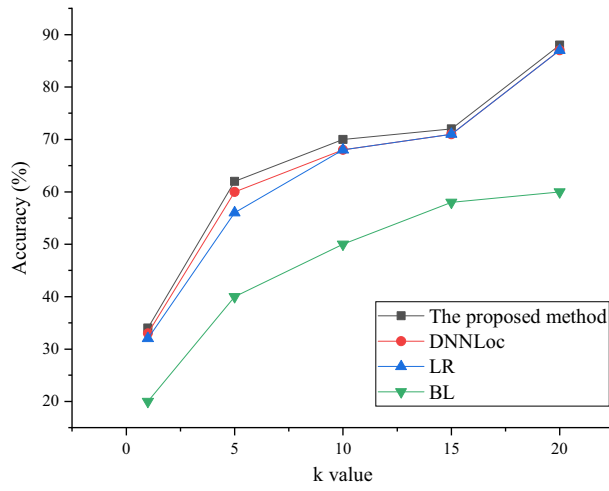
**Figure 6.** Test results for sentences with 1–49 words in length.



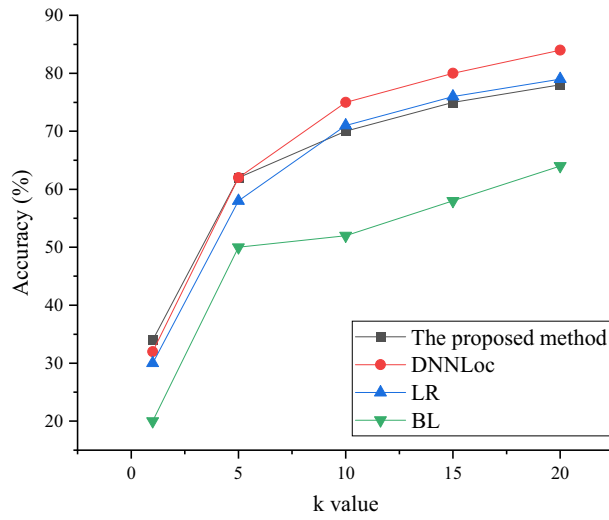
**Figure 7.** Test results for sentences with 51–89 words in length.



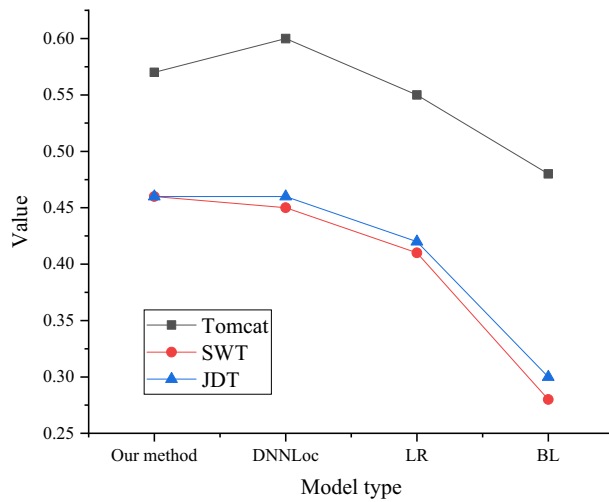
**Figure 8.** The accuracy on the Tomcat dataset.



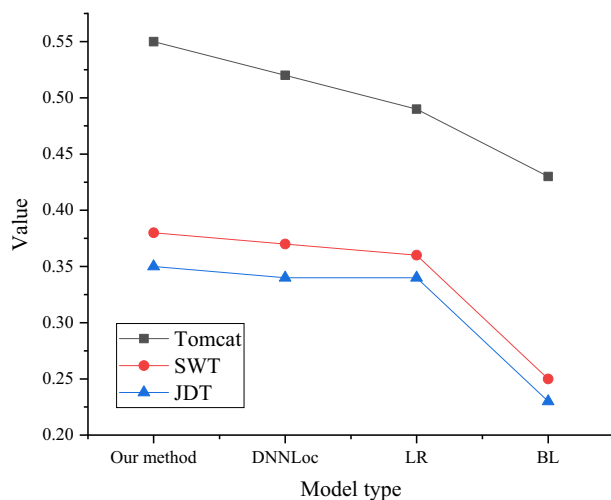
**Figure 9.** The accuracy on the JDT dataset.



**Figure 10.** The accuracy on the SWT dataset.



**Figure 11.** MRR on different datasets.



**Figure 12.** MAP on different datasets.

Figure 11 shows that the proposed model basically presents good results. The MRR of the proposed model on the Tomcat dataset is 0.54, that of the DNNLoc model is 0.6, that of the LR model is 0.55, and that of the BL model is 0.5. Only on the Tomcat dataset, the MRR of the proposed model is slightly lower than that of the DNNLoc model. On the SWT dataset and the JDT dataset, the MRR of the proposed model is the highest.

In the comparison of MAP on different datasets illustrated in Fig. 12, which is basically consistent with the results presented in Fig. 11, whether it is on the Tomcat dataset, SWT dataset, or JDT dataset, the MAP of the proposed model is the highest. The average effect of the proposed model is better than other comparison methods on the Tomcat, SWT, and JDT datasets.

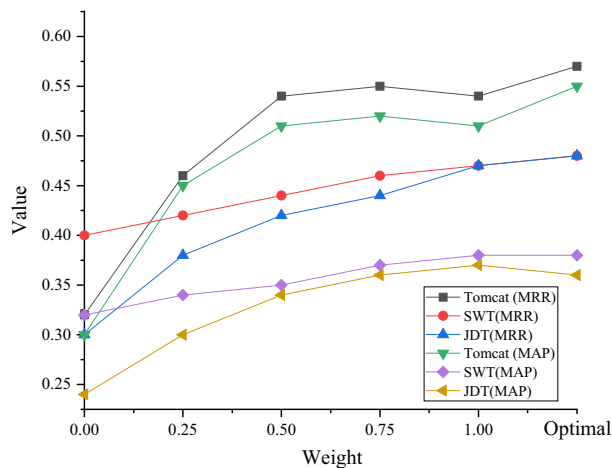
Under different  $w_2$  weights, the changes of MRR and MAP are shown in Fig. 13.

In Fig. 13, the abscissa is the five  $w_2$  values set in this test to calculate the weight comprehensive score, and the last one is the weight combination calculated by the PSO algorithm. Figure 13 shows that the MRR and MAP of the weight combination of the PSO algorithm are basically high.

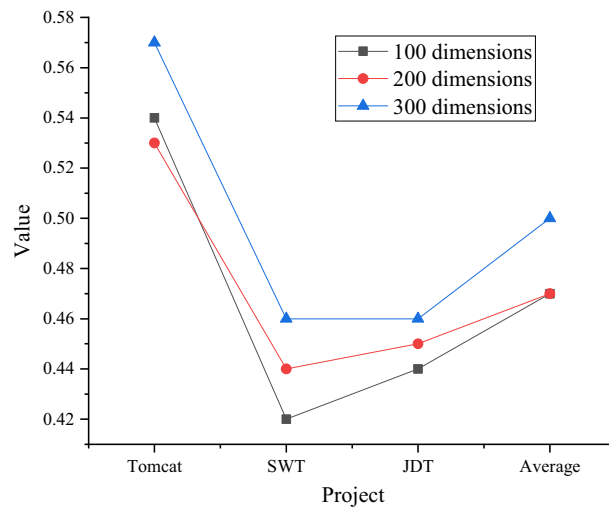
The embedding effects of Glove words of different dimensions are shown in Figs. 14 and 15.

Figure 14 shows that the proposed method works best in the 300-dimensional word embedding model. The average MRR is high, which is an improvement of 0.03 compared with 100-dimensional and 200-dimensional word. From an average point of view, the average metric values for the 100-dimension and 200-dimension in the graph are the same. The MRR of this method in the 300-dimensional word embedding model reaches 0.5.

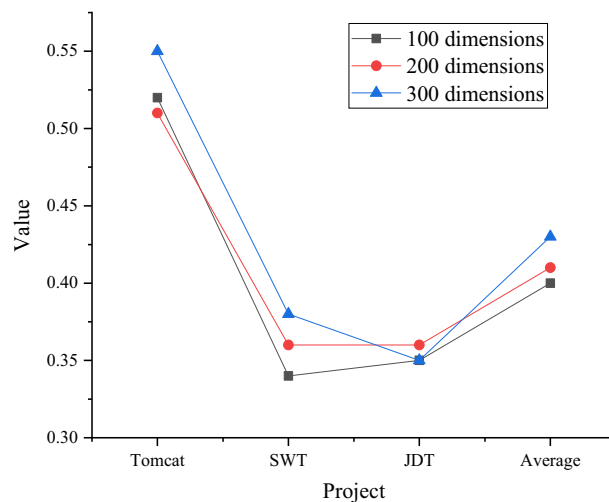
Figure 15 shows that the MAP of the proposed method also shows a relatively better effect in the 300-dimensional word embedding model. The MAP of 300-dimension word is 0.03 and 0.02 higher than that of 100-dimension and 200-dimension, respectively. The average MAP of the 200-dimension model is higher than that of the 100-dimension model, and the MAP of the proposed method in the 300-dimension word embedding model



**Figure 13.** Indicator changes under different  $w_2$ .



**Figure 14.** MRR of embedded words of different dimensions.



**Figure 15.** MAP of embedded words of different dimensions.

reaches 0.43. In Figs. 14 and 15, the proposed method is greatly affected by the dimension of the word embedding model, and the 300-dimensional word embedding model has the best effect.

## Conclusions

The statistical translation model software is designed and optimized based on the application of improved deep learning techniques and meta-heuristic algorithms in the statistical translation model. A translation model is designed by combining RNN and word vector generation algorithm. Combined with Chinese and English subtitle data, the model is trained and its performance is compared. In addition, a software defect location method combining sentence embedding technique and PSO algorithm is proposed. The method uses Glove and SIF to calculate the semantic similarity between software defect reports and source code files. In addition, a question-and-answer interaction module is designed for humanoid translation robot, and a personalized hybrid recommendation mechanism is built based on blockchain technology. Finally, the performance of the model is tested. The results show that the English subtitle translation model and software defect location model have good performance. This work provides a reference for the further development of machine learning and data mining. Applying the results of this work to the humanoid translation robot is expected to achieve a good educational effect in college film majors. Nevertheless, the research still has some shortcomings. The performance of the translation model does not exceed the Moses algorithm, and there is a lexical gap between the software defect report in the software defect location model and the source code file. In addition, the question-and-answer module of humanoid robot is designed, but the algorithm performance of this part is not evaluated. There is also a lack of case studies applying blockchain technology. Therefore, the RNN-based word embedding algorithm will be improved in the future to extract keywords from the text. It should also increase the algorithm performance

evaluation for humanoid robot interactive question and answer module and increase the application of block-chain technology. After further improvement, we hope to get a more feasible improvement scheme of humanoid translation robot (Supplementary Information).

### Data availability

All data generated or analysed during this study are included in this published article [and its supplementary information files].

Received: 15 August 2022; Accepted: 6 March 2023

Published online: 09 March 2023

### References

1. Stahlberg, F. Neural machine translation: A review. *J. Artif. Intell. Res.* **69**, 343–418 (2020).
2. Fan, A. *et al.* Beyond English-centric multilingual machine translation. *J. Mach. Learn. Res.* **22**(107), 1–48 (2021).
3. Lee, S. M. The impact of using machine translation on EFL students' writing. *Comput. Assist. Lang. Learn.* **33**(3), 157–175 (2021).
4. Freitag, M. *et al.* Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Trans. Assoc. Comput. Linguist.* **9**, 1460–1474 (2021).
5. Liu, Y. *et al.* Multilingual denoising pre-training for neural machine translation. *Trans. Assoc. Comput. Linguist.* **8**, 726–742 (2021).
6. Pham, M., Crego, J. M. & Yvon, F. Revisiting multi-domain machine translation. *Trans. Assoc. Comput. Linguist.* **9**, 17–35 (2021).
7. Farkas, A. & Németh, R. How to measure gender bias in machine translation: Real-world oriented machine translators, multiple reference points. *Soc. Sci. Hum. Open.* **5**(1), 100239 (2022).
8. Park, C., Park, K., Moon, H., Eo, S. & Lim, H. A study on performance improvement considering the balance between corpus in neural machine translation. *J. Korea Conver. Soc.* **12**(5), 23–29 (2021).
9. Rajan, K., Zielesny, A. & Steinbeck, C. STOUT: SMILES to IUPAC names using neural machine translation. *J. Cheminform.* **13**(1), 1–14 (2021).
10. Luo, Y. *et al.* Natural language to visualization by neural machine translation. *IEEE Trans. Vis. Comput. Graph.* **28**(1), 217–226 (2021).
11. Song, L., Gildea, D., Zhang, Y., Wang, Z. & Su, J. Semantic neural machine translation using AMR. *Trans. Assoc. Comput. Linguist.* **7**, 19–31 (2019).
12. Zhou, L., Zhang, J. & Zong, C. Synchronous bidirectional neural machine translation. *Trans. Assoc. Comput. Linguist.* **7**, 91–105 (2019).
13. Brynjólfsson, E., Hui, X. & Liu, M. Does machine translation affect international trade? Evidence from a large digital platform. *Manag. Sci.* **65**(12), 5449–5460 (2019).
14. Prates, M. O., Avelar, P. H. & Lamb, L. C. Assessing gender bias in machine translation: A case study with google translate. *Neural Comput. Appl.* **32**(10), 6363–6381 (2020).
15. Popel, M. *et al.* Transforming machine translation: A deep learning system reaches news translation quality comparable to human professionals. *Nat. Commun.* **11**(1), 1–15 (2020).
16. Gete, H. & Etchegoyhen, T. Making the most of comparable corpora in neural machine translation: A case study. *Lang. Resour. Evaluat.* **16**, 1–29 (2022).
17. Wu, X. *et al.* A study of BERT for context-aware neural machine translation. *Mach. Learn.* **22**, 1–19 (2022).
18. Li, J. Critical thinking of a translator: Expanding the practice of using and editing machine translation. *Think. Skills Creativ.* **43**, 100995 (2022).
19. Khoong, E. C. & Rodriguez, J. A. A research agenda for using machine translation in clinical medicine. *J. Gen. Intern. Med.* **25**, 1–3 (2022).
20. Aust, J., Shankland, S., Pons, D., Mukundan, R. & Mitrovic, A. Automated defect detection and decision-support in gas turbine blade inspection. *Aerospace* **8**(2), 30 (2021).
21. Pérez, F., Lapeña, R., Marcén, A. C. & Cetina, C. Topic modeling for feature location in software models: Studying both code generation and interpreted models. *Inform. Softw. Technol.* **140**, 106676 (2021).
22. Zhu, K., Ying, S., Zhang, N. & Zhu, D. Software defect prediction based on enhanced metaheuristic feature selection optimization and a hybrid deep neural network. *J. Syst. Softw.* **180**, 111026 (2021).
23. Ballarín, M., Marcén, A. C., Pelechano, V. & Cetina, C. On the influence of model fragment properties on a machine learning-based approach for feature location. *Inf. Softw. Technol.* **129**, 106430 (2021).
24. Zhao, Z. Review of non-destructive testing methods for defect detection of ceramics. *Ceram. Int.* **47**(4), 4389–4397 (2021).

### Acknowledgements

Fujian Educational and Scientific Research Project: Application of AI in Translation Practice (JAT210538).

### Author contributions

F.Y. and J.D. contributed to conception and design of the study. F.Y. wrote the first draft of the manuscript. J.D. wrote sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

### Competing interests

The authors declare no competing interests.

### Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-023-31053-5>.

**Correspondence** and requests for materials should be addressed to F.Y.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

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





**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2023