International Journal for Housing Science and Its Applications
Publish August 3, 2025. Volume 46, Issue 3 Pages 741-752

https://doi.org/10.70517/ijhsa46355

A Study on Improving Japanese Writing Skills by Constructing Japanese Syntactic Analysis and Generation Technology Using Computational Methods and Artificial Intelligence Models

Tingting Xu[1] and Dongmei Shen[1,*]
[1] School of Foreign Languages, Guangzhous City University of Technology, Guangzhou, Guangdong, 510800, China
Corresponding authors: (e-mail: 15918571957@163.com).

Abstract In this paper, Japanese syntactic analysis and text embellishment techniques are designed to improve students' Japanese writing skills. Since Japanese dependency parsing is an important part of Japanese syntactic analysis. In this regard, this paper adopts the SVM model to generate a classifier using the labeled corpus as a way to determine whether there is a dependency relationship between two text sections. In order to improve the parsing accuracy of the SVM model, this paper proposes a Japanese dependency parsing method based on NN-LSVM pruning of a large-scale training corpus for dependency parsing on the basis of SVM. After that, a text touch-up technique based on syntactic structure is designed, which introduces a contrastive representation learning method and pushes the model to deeply understand the modeling relationship between semantic and syntactic structural information by adjusting the loss function to further mine more appropriate syntactic structures and expressions in order to improve the effect of the text touch-up technique. After verifying that the two techniques are feasible, this paper designs a practical task for teaching Japanese writing. During the practice, the Japanese writing scores of the experimental class using the NN-LSVM model and the language generation model designed in this paper for writing tutoring improved significantly ($P<0.05$), and there was no significant change in the control class. It shows that the technique in this paper can have the effect of promoting students' Japanese writing ability.

Index Terms Dependency, NN-LSVM, Textual embellishment technique, Japanese writing

I. Introduction

As a branch of artificial intelligence, natural language processing has been increasingly emphasized by workers in many disciplines, and its application prospect is very broad [1], [2]. At present, linguistic information processing technology has been widely used in many practical systems [3]. In today's information-exploding society, natural language processing serves as a high-level and important aspect of linguistic information processing technology [4], [5].

In the field of natural language processing, syntactic analysis belongs to the semantic parsing model, which is a layer above the lexical analysis part, so syntactic analysis is one of the fundamental works in the study of natural language understanding of Japanese, and it is a prerequisite for semantic analysis of given sentences and texts [6]-[9]. Japanese syntactic analysis is one of the core elements of Japanese natural language understanding and machine translation [10]. The main task of syntactic analysis is to generate a syntactic tree of phrases, given a sentence, with the grammatical features of the language as the main source of knowledge, and to specify the relationship between the parts of the sentence through the form of the tree, which is in essence a process of disambiguation [11]-[14]. The improvement of the performance of syntactic analysis will have an important role in promoting applications such as information retrieval, information extraction, and machine translation [15], [16].

Based on the syntactic analysis of artificial intelligence, the application of generative technology is important for improving Japanese writing ability [17], [18]. Japanese composition is an important part of learning Japanese, which can improve language expression, increase vocabulary, and improve the use of grammar [19], [20]. However, writing is still a difficult task for many learners [21]. Artificial intelligence generation technology is based on the application of natural language processing, machine learning and deep learning and other technology bases, through the analysis and learning of a large number of texts, it can generate articles that conform to grammatical norms and are clear in logic, and as an effective auxiliary tool, the generation technology can provide students with personalized writing guidance and assistance to improve their Japanese writing ability [22]-[25].

Dependency parsing in Japanese is an important part of the field of Japanese natural language processing, and in this paper, we investigate the parsing of Japanese dependencies and establish an effective parsing model. Firstly, an SVM-based Japanese dependency parsing model is established, then a group-block-based text-section step-by-step application algorithm is used to control the parsing process, and finally an NN-LSVM-based Japanese dependency parsing model is established by improving the SVM model using LSVM and NN methods. In order to realize the embellishment of Japanese text, a syntactic structure control method based on contrastive representation learning is introduced. The two techniques are used in Japanese writing training, and controlled experiments are conducted to verify the effectiveness of the techniques in improving Japanese writing ability.

II. Realization of Japanese syntactic analysis techniques

II. A.Japanese Syntactic Analysis

Syntactic analysis means automatically identifying the syntactic units contained in a sentence and the relationships between these syntactic units based on a given grammar. Syntactic analysis is a key component of natural language understanding and is the basis for further semantic analysis of natural language [26]. Research on syntactic analysis is broadly divided into 2 approaches: rule-based approaches and statistical-based approaches.

The rule-based approach is a knowledge-based rationalist approach, based on linguistic theory, emphasizing the linguist's knowledge of linguistic phenomena, and adopting non-ambiguous forms of rules to describe or explain ambiguous behaviors or ambiguous properties.

Statistically based syntactic analysis must somehow characterize the formal and grammatical rules of a language, and this characterization must be obtainable by training on the results of a known syntactic analysis, which is the syntactic analysis model. Statistical syntactic analysis based on treebanks is the mainstream technique of modern syntactic analysis. The purpose of constructing a statistical syntactic analysis model is to evaluate a number of possible syntactic analysis results (usually represented in the form of syntactic trees) in a probabilistic form and to directly choose the most probable result among these possible analysis results. A statistically based syntactic analysis model is in essence a probabilistic evaluation function for evaluating syntactic analysis results, i.e., for any input sentence s and its syntactic analysis result t, a conditional probability $P(T \mid s)$ is given, and from this, the syntactic analysis result which is considered by this syntactic analysis model to be the most probable one is identified [27], i.e., it is found $\tilde{t} = \dfrac{\arg\max P(T \mid s)}{t}$, and the sample space of the syntactic analysis problem is $S \times T$ (where: S is the set of all sentences and T is the set of all syntactic analysis results).

According to the characteristics of the Japanese language, a language like Japanese cannot use strict sentence construction rules. Instead, the richness of adjuncts and the syntactic and semantic information provided by the adjuncts suggest that it is more appropriate to use the dependency analysis method in Japanese syntactic analysis. The necessary elements in Japanese syntactic analysis are: dependency conditions, types of modifying relations, priority conditions, and basic priority.

(1) Dependency condition - consists of a pair of stanzas in which there is a dependency relationship.
(2) Type of modifying relationship - the type that constitutes the dependency relationship.
(3) Priority condition - proximal priority or remote priority.
(4) Basic Priority - Priority is assigned to matching pairs of sections.

II. B.NN-LSVM-based parsing of Japanese language dependencies

II. B. 1) LSVM model

Although SVM has been proven to be a very effective machine learning model, the optimization and classification speeds are still not quite satisfactory for large-scale sample sets, especially in the case of many support vectors. Therefore, in this paper, a pruning method is used to remove these samples in order to reduce the training set size, simplify the classification hyperplane, and improve the parsing accuracy and speed [28]. The specific steps are as follows:

(1) Randomly draw a small-scale sample set S from the large-scale sample set L, and then train with the small-scale sample set S to obtain the initial classifier. The size of the small-scale sample set is determined based on two conditions:

1) Ensure that it is not costly to train using it.
2) Ensure that the classifier trained using it has a certain classification accuracy.

(2) Prune the large-scale sample set L with the initial classifier, and then train it with the approximately reduced set to obtain the final classifier. This is done by setting the classification hyperplane of the initial classifier to be H. For any sample s of L, let the distance between s and H (greater than or equal to zero) be d. If $1-\varepsilon < d < 1+\varepsilon$,

this sample is retained, otherwise it is deleted, where $0 < \varepsilon < 1$ is a threshold that can be adjusted. The adjustment of the threshold has two functions:

1) To control the size of the approximate reduction set.

2) Influence the classification accuracy of the final classifier. In practice, the main purpose is to adjust the threshold value to get the near-optimal classifier relative to the threshold value, which is not difficult to adjust manually. For convenience, the support vector machine that adopts the above culling strategy is called LSVM.

The above censoring strategy is shown in Fig. 1. Where H denotes the classification hyperplane of the initial classifier $H+$ and $H-$ denote the hyperplanes where the two types of support vectors are located. The pruning principle is to keep the samples whose distance from $H+$ and $H-$ is less than ε, and delete the other samples. In simple terms, this means leaving the samples that are closer to the support vectors of the initial classifier. This pruning strategy captures the essence of support vector machines, i.e., the classifier is only related to the support vector and not to the other vectors (samples). By employing this pruning strategy, the samples left behind will be the ones that greatly help the classification, while the deleted samples will not help the classification or even be counterproductive (e.g., leading to over-learning instead of decreasing the accuracy of the classifier).

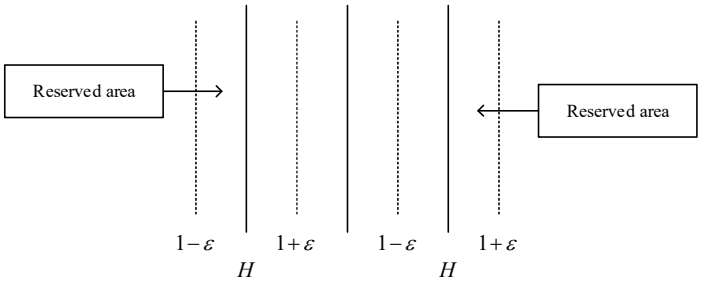

Figure 1: Pruning training samples with LSVM

II. B. 2) NN-SVM modeling

Although the goal of support vector machines is to achieve strong generalization ability, over-learning problems may occur with respect to specific sample sets. For example, when the two sample sets are heavily overlapped, the decision surface of SVM may reduce its generalization ability due to excessive complexity.

In this paper, we propose another improved SVM (NN-SVM): it first prunes the training set by deciding the trade-offs between each sample and its nearest-neighbor class label according to its similarities and differences, and then trains the SVM to obtain a classifier. This approach is very parsimonious.

In this paper, we adopt the following strategy to prune the training set:

First find out the nearest neighbor of each point and then for each point if the point belongs to the same class as its nearest neighbor keep the point and if the point belongs to a different class than its nearest neighbor remove the point. Euclidean distance is used as the distance between two vectors i.e. set:

$$\begin{aligned} x_i &= (x_i^1, x_i^2, ..., x_i^n) \\ x_j &= (x_j^1, x_j^2, ..., x_j^n) \end{aligned} \tag{1}$$

Then the distance between x_i and x_j is defined as:

$$D(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_i^k - x_j^k)^2} \tag{2}$$

The nearest neighbor of a sample is the sample that is closest to it under the above definition.

Below we give the algorithm for implementing the above method.

Given a training set $(x_1, y_1), (x_2, y_2), ..., (x_m, y_m), x_i \in R^n, y_i \in \{1, -1\}, i = 1, ...m$. Represent the training set as a matrix:

$$TR_{m\times(n+1)} = [XY] \tag{3}$$

where $X = \begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_m \end{bmatrix}$, $Y = \begin{bmatrix} y_1 \\ . \\ . \\ . \\ y_m \end{bmatrix}$.

The pruning algorithm is as follows.

(1) Find the nearest neighbor of each vector:

1) Find the distance of each point from each of the other points, defined as ∞ from itself.

2) Find the shortest distance and the corresponding point (nearest neighbor).

(2) Determine whether the class label of each vector agrees with its nearest neighbor, marked 1 and -1 respectively.

(3) Delete the vectors that do not agree with the class label of their nearest neighbors.

The pruned training set can be obtained after the above 3 steps.

The above method of pruning the training set using nearest neighbors first and then training with SVM to get the classifier is called NN-SVM.Compared with SVM, NN-SVM has the following advantages:

(1) Classification correctness is expected to increase

Due to pruning the training set, the classification boundary of NN-SVM is simplified compared to the overly complex classification boundary of SVM, and thus its generalization ability may be stronger and the classification correct rate may be higher. The experimental results confirm this idea. It can be seen that NN-SVM is an effective way to solve the problem of over-learning and weakened generalization ability of the classifier due to the serious overlapping of the two classes.

(2) Shorter time used for classification

After the training set is pruned, the support vectors of the classifier are greatly reduced, and the time used for classification is proportional to the number of support vectors, so the classification time is greatly saved.

(3) Can be used for larger training sets

Since the pruning process makes the larger training set smaller, NN-SVM can be applied to a larger training set under the same hardware conditions.

II. B. 3) NN-LSVM

Both the LSVM model and the NN model start from simplifying the size of the training set and deleting bad samples to achieve the purpose of improving the accuracy of Japanese dependency parsing and increasing the parsing speed.

In this paper, we combine these two deletion strategies with the SVM statistical model to propose an NN-LSVM model for Japanese dependency parsing.

The NN-LSVM model is specified as follows:

(1) A training set S is obtained based on the preprocessed corpus using a block-based text-segment step-by-step application algorithm.

(2) Randomly select a small portion (one-eighth of the samples in the experiments of this paper) of samples in S to form a small training set S1.

(3) Use SVM to train S1 to obtain a classifier C1.

(4) Bring each sample from the initial training set S into the classifier C1 and find the distance d from each sample to the hyperplane of C1. Have a threshold ε for $0<\varepsilon<1$, if $1-\varepsilon<d<1+\varepsilon$ then keep this sample, otherwise delete this sample. After deletion the training set S is reduced to S'. The SVM model is used to train S' to obtain the classifier C'. Do an open test with the classifier C' to get the parsing accuracy.

(5) Adjust the size of ε and repeat the process of (4) to get S' with the highest parsing accuracy.

(6) Use the method of NN to remove some more bad samples in S' to get the reduced set S'.

(7) Use the training sample set S'' of SVM to get the final classifier C''.

(8) Dependency parsing using the classifier C''.

II. C.Experimental results and analysis

II. C. 1) Experimental corpus

The Asahi Shimbun, a text corpus from Kyoto University in Japan, was used for the experiments in this paper. Part of the Asahi Shimbun corpus of January 10, 2023, with a total of 7,958 sentences (the number of sentences is used in most Japanese dependency studies) and 77,705 text sections, was used as the training corpus, and 1,220 sentences of January 9 with a total of 12,206 text sections were used as the test corpus. These corpora have been processed for word separation, stanza separation, lexical annotation, morphological discrimination, and so on, and also for determining juxtaposition and dependency, which makes them suitable for dependency parsing, and they have been used in most of the studies of Japanese dependency relations.

The commonly used dependency correctness rate and sentence correctness rate were used to evaluate the analysis results and are defined as follows:

$$\text{Dependency accuracy} = \frac{\text{The number of correctly identified dependencies}}{\text{Number of all dependencies}} \tag{4}$$

$$\text{Sentence accuracy} = \frac{\text{Completely analyze the number of correct sentences}}{\text{Number of sentences in the test set}} \tag{5}$$

II. C. 2) Experimental results

(1) Dependency resolution of Japanese language based on SVM and NN-SVM

The original model (SVM) and the improved model (NN-SVM) are used for dependency parsing respectively, and the results are shown in Table 1. As can be seen in the table, after increasing the number of training corpus, the NN-SVM model not only greatly reduces the training time, but also reduces the number of training samples that play a side effect on the classification, and improves the parsing accuracy.The dependency accuracy of SVM and NN-SVM reaches 88.11% and 89.21%, respectively.

Table 1: SVM and NN-SVM experiment were compared

Test item	SVM	SVM	NN-SVM
Quantity of training corpus (day)	1	8	8
Dependent accuracy (%)	86.39	88.11	89.21
Sentence accuracy (%)	41.26	46.32	47.82
Analytic velocity (s/sentence)	0.23	1.3	0.75
Support vector number	5362	34965	17365
Training sample number	15063	122156	97632
Training time(min)	4	901	124

(2) Analysis of Japanese language dependency based on NN-LSVM

The NN-SVM model analyzed above has already had a good improvement compared to the original model, so this paper adds the LSVM model to it to form the NN-LSVM model.

Firstly, the threshold value ε is set to 0.1 to 1.0, and when the judgment is made by using NN-SVM, if the distance of the test vector from the classification hyperplane is greater than ε, then the judgment result of NN-SVM will be taken as the final result. When the distance is less than ε, the LSVM algorithm is used to make a judgment and the result of its judgment is taken as the final classification result of this vector.

The dimension of the vector for SVM test is 33, so in this paper, we set the distance parameter d from 0 to 33, so for the same threshold value ε, we need to conduct 34 experiments for different values of d.

Experimental results for different values of d when $\varepsilon = 0.1$ and $\varepsilon = 0.2$.

The results of parsing Japanese dependency based on the NN-LSVM model are shown in Table 2. It can be seen that when $\varepsilon = 0.1$ and the value of d is between 27 and 33, the SVM fusion LSVM algorithm achieves a good result, and the text section parsing accuracy reaches 89.61%. When $\varepsilon = 0.2$, both text-section parsing accuracy and sentence parsing accuracy decreased.

Table 2: Analysis of Japanese dependency relationship based on NN-LSVM

d	$\varepsilon = 0.1$		$\varepsilon = 0.2$	
	Dependent accuracy	Sentence accuracy	Dependent accuracy	Sentence accuracy
$d = 1$	89.07%	47.13%	89.10%	47.33%
$d = 5$	89.32%	47.30%	89.13%	47.34%
$d = 10$	89.38%	47.33%	89.14%	47.41%
$d = 20$	89.49%	47.42%	89.32%	47.44%
$d = 25$	89.58%	47.56%	89.35%	47.48%
$d = 26$	89.60%	47.61%	89.36%	47.59%
$d = 27$	89.61%	48.55%	89.55%	48.01%
$d = 28$	89.56%	48.32%	89.52%	47.97%
$d = 29$	89.56%	48.32%	89.52%	47.97%
$d = 33$	89.56%	48.32%	89.52%	47.97%

This indicates that the scope of the LSVM algorithm is closer to the classification hyperplane, and the LSVM algorithm is able to utilize most of the classes of the surrounding support vectors to better determine the classes of the tested vectors, thus improving the final parsing accuracy.

The experiment proves that after pruning the training set using the NN-LSVM method, not only the size of the training set is reduced, but also the samples that have little, no, or even bad influence on the classification are deleted, and basically the good samples that play a decisive role in the classification are retained. As a result, the training time and the amount of memory required for training are shortened, and the parsing accuracy and parsing speed are improved, which makes the Japanese dependency parsing model proposed in this paper possible for practical application.

III. Text touch-up techniques based on syntactic structure control

III. A. Methods of text touch-up

The previous section parses Japanese dependencies based on NN-LSVM, and this chapter constructs a language generation model in order to improve students' Japanese writing skills. In order to enable the model to model the logical relationship between semantics and syntax of Japanese writing-oriented texts at a deeper level, so that the generation results of the model have a better quality, this paper introduces the contrastive representation learning technique, and implements two kinds of loss functions, namely, the contrastive learning loss of semantic content and the contrastive learning loss of syntactic structure, by designing and implementing them, which are used to improve the quality of the textual representations.

In this paper, we use an overall model structure of a semantic information encoder, a syntactic structural information encoder and a decoder. Denote these three components as E_{sem}, E_{syn} and D, respectively. For a given Japanese writing text Y_i, this paper generates a sentence X_i with different syntactic structure but the same semantic content as it by back-translation, which is defined in this paper as an output pair (X_i, Y_i), and since the goal of this paper is to generate a target sentence that has the same meaning as X_i with the same meaning as X_i but with similar syntax as Y_i, so, in this paper, we firstly encode its input part X_i through the semantic encoder E_{sem} to get its semantic encoding C_{xi}, and encode its output part Y_i through the semantic encoder E_{sem} and syntactic encoder E_{syn} to obtain its semantic encoding C_{yi} and syntactic encoding S_{yi} respectively. And at the same time, this topic labels the syntactic structure reconstructed by the model based on the semantic content as Z_i, so this paper also needs to encode the syntactic structure of Z_i to obtain S_{zi}.

The model of the textual touch-up method designed in this paper that introduces contrastive representation learning is shown in Fig. 2.

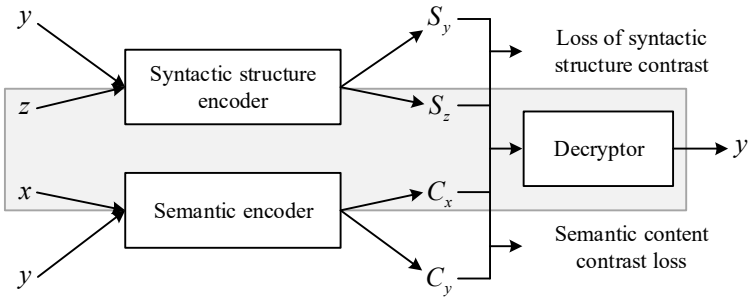

Figure 2: Comparison indicates learning methods

First, as shown in Eq. (6), this paper uses the negative log-likelihood loss (NLL) as the underlying optimization objective:

$$L_i^{nll} = -\frac{1}{|Y_i|}\sum_{t=1}^{|Y_i|} I(y_t)^T \log P_t \tag{6}$$

Second, this paper designs a semantic content contrastive learning loss (CCL), considering that X_i and Y_i should have similar semantic content, and their semantic content feature codes should be close to each other in the space of semantic features. Contrastive learning aims to minimize the distance between positive examples and maximize the distance between negative examples, which helps to establish this relationship of close proximity between similar semantics in the feature space; therefore, in this paper, we design a semantic content contrastive

learning loss to encode sentences with similar semantic information closer together, and encode sentences that do not have similar semantic information farther apart.

Formally, for n input-output pairs in a batch, this paper simply selects the corresponding semantic codes of X_i and Y_i as the positive examples, and the other non-corresponding semantic codes as the negative examples, which gives the semantic content comparison learning loss function of X_i and Y_i, as in shown in Equation (7) and Equation (8):

$$L_{X_i}^{ccl} = -\log \frac{\exp(\frac{c_{X_i} \cdot c_{Y_i}}{\tau})}{\exp(\frac{c_{X_i} \cdot c_{Y_i}}{\tau}) + \sum_{j \neq i} \exp(\frac{c_{X_i} \cdot c_{T_j}}{\tau})} \tag{7}$$

$$L_{Y_i}^{ccl} = -\log \frac{\exp(\frac{c_{Y_i} \cdot c_{X_i}}{\tau})}{\exp(\frac{c_{Y_i} \cdot c_{X_i}}{\tau}) + \sum_{j \neq i} \exp(\frac{c_{Y_i} \cdot c_{T_j}}{\tau})} \tag{8}$$

The overall semantic content comparison learning loss function is then the sum of the two terms, as shown in Equation (9):

$$L^{ccl} = \sum_{i=1}^{n} (L_{X_i}^{ccl} + L_{Y_i}^{ccl}) \tag{9}$$

Meanwhile, this paper designs a syntactic structure contrast learning loss (SCL), which is similar to the semantic content contrast learning, considering that the target syntax Z_i generated in this paper should have a high degree of similarity with Y_i and its syntactic structure feature encoding should be close to each other in the space of syntactic features. Therefore, in this paper, we design a grammatical structure comparison learning loss to bring the sentence encodings with similar grammatical structure information closer to each other, while pulling the sentence encodings without similar grammatical structure information farther away from each other.

Therefore, formally, this paper also selects the grammatical encodings of selected corresponding Z_i should and Y_i as positive examples, and the other non-corresponding grammatical encodings as negative examples, which gives the grammatical structure comparison learning loss function of Z_i should and Y_i as shown in Eq. (10), Eq. (11) are shown:

$$L_{Z_i}^{scl} = -\log \frac{\exp(\frac{s_{Z_i} \cdot s_{Y_i}}{\tau})}{\exp(\frac{s_{Z_i} \cdot s_{Y_i}}{\tau}) + \sum_{j \neq i} \exp(\frac{s_{Z_i} \cdot s_{T_j}}{\tau})} \tag{10}$$

$$L_{Y_i}^{scl} = -\log \frac{\exp(\frac{s_{Y_i} \cdot s_{Z_i}}{\tau})}{\exp(\frac{s_{Y_i} \cdot s_{Z_i}}{\tau}) + \sum_{j \neq i} \exp(\frac{s_{Y_i} \cdot s_{T_j}}{\tau})} \tag{11}$$

The overall grammatical structure comparison learning loss function is then the sum of the two terms, as shown in Equation (12):

$$L^{scl} = \sum_{i=1}^{n} (L_{Y_i}^{scl} + L_{Z_i}^{scl}) \tag{12}$$

Finally, as shown in Eq. (13), the overall loss function can be expressed as:

$$L = \sum_{i=1}^{n} L_i^{nll} + w_1 L^{ccl} + w_2 L^{scl} \tag{13}$$

where w_1 and w_2 denote the weights corresponding to the loss functions of the two contrastive representation learning designed in this paper, respectively.

III. B. Performance experiments

III. B. 1) Experimental data set

The datasets used in this experiment include CLUE (Japanese Language Understanding Evaluation Benchmark), CGED (Japanese Grammatical Error Diagnosis), and the Student Japanese Writing (SJW) dataset obtained by constructing.The corpus in CGED is derived from the test results of the Japanese Proficiency Examination (JPE).

This dataset contains different types of grammatical errors and ideally reflects the error situations faced in everyday writing.CLUE is the largest language understanding corpus for Japanese, in which the Japanese Wikipedia dataset has been selected and multilingual characters have been mixed in order to represent the complex semantic environment. The constructed SJW dataset is collected from the daily writing training of Japanese language majors in A university. For model training, 200,000 sentences were randomly selected from the SJW dataset for training in this paper. In the main evaluation experiment, 1,000 samples were randomly selected from each of the above datasets as test data. In addition, for these 1000 sentences selected from the CLUE corpus, the corresponding error sentences were constructed using a strategy similar to the one used to generate error sentences for policy network training.

III. B. 2) Performance analysis

Here, several open-source Chinese detection and error correction tools are introduced to examine the error correction performance of different methods.Founder detection tool, a classical technique for Chinese proofreading, is used to detect the number of errors in corrected sentences. The number of errors reflects the error correction capability of the method. Natural language processing tools from Baidu's artificial intelligence platform were also used in the experiments, where the language perplexity (PPL) calculation can assess the fluency of a sentence based on each word in the sentence, and the DNN score can assess the likelihood of each word in the sentence. Wherein the normalized performance values in said graphical results are calculated by ratioing the PPL and DNN score values with the original sentence.BLEU is a classical evaluation matrix that is widely used in the evaluation of translation tasks. It reflects the ability of a textual error correction method to restore a sentence to its original form by comparing the similarity between the corrected sentence and the original sentence as a reference.

In the error correction performance analysis, a difficulty model called variable length run-on scenario is also considered, in which the two length values of the incorrect sentence and the ideal correct sentence are different. However, some of the existing text correction methods can only perform correction in fixed-length scenarios. Therefore, the following methods were adopted as a baseline for comparing the state-of-the-art methods including ELECTRA, ERNIE, MacBERT, and TtT as the state-of-the-art methods in variable-length correction.

The performance test results on SJW, CLUE and CGED datasets are shown in Fig. 3 to Fig. 5, respectively. The embellishment method designed in this paper obtains better performance compared to the baseline on various evaluation metrics. This represents its ability to efficiently detect semantic errors in sentences and correct them with appropriate semantics.

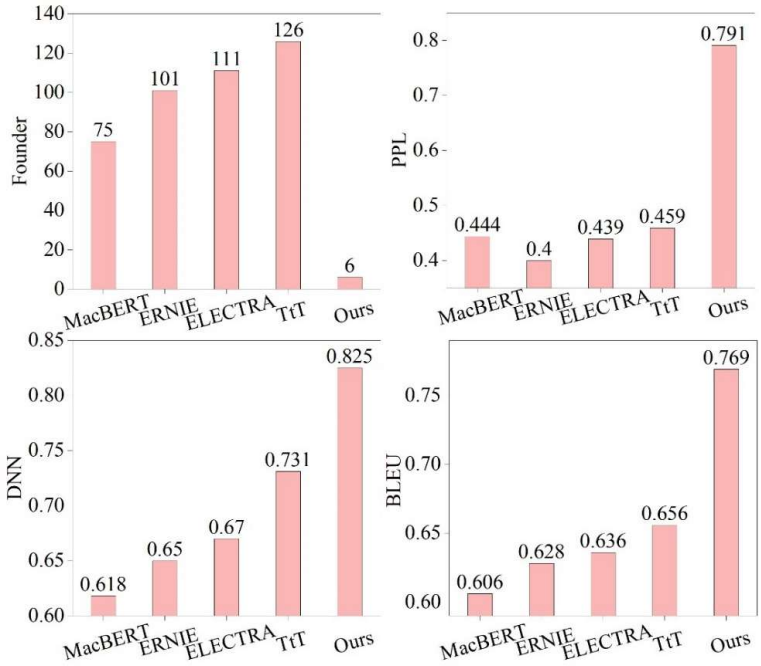

Figure 3: Performance test results on the SJW data set

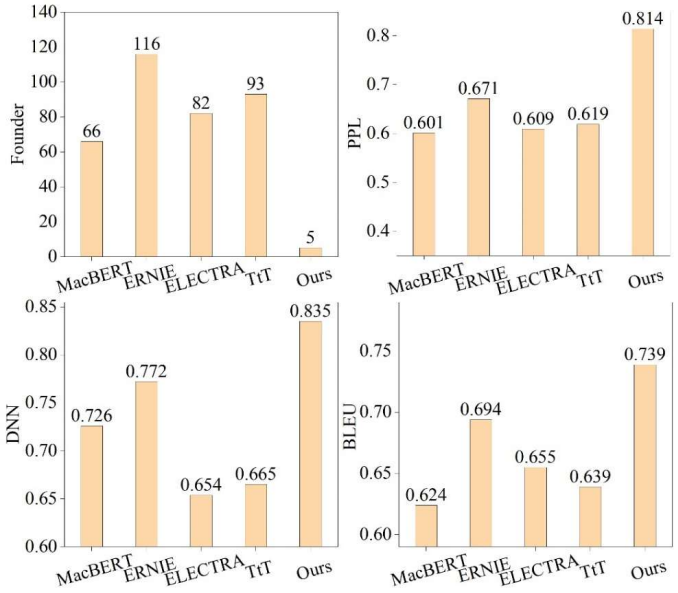

Figure 4: Performance test results on the CLUE data set

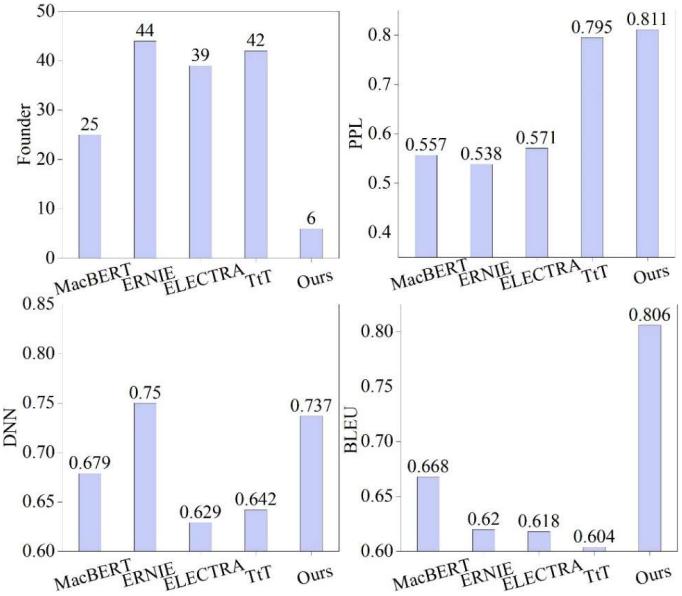

Figure 5: Performance test results on the CGED data set

IV. Japanese Writing Classroom Teaching Practices

IV. A. Research methodology

We randomly selected the sophomore class (1) of Japanese majors in school A as an experimental class (42 students, of whom 23 were male and 19 were female) and the sophomore class (2) as a control class (42 students, of whom 22 were male and 20 were female).

The pre-test was conducted in April 2023 using the topic “spring trip” as the pre-test question, and the post-test was conducted using the Japanese essay “Vacation Arrangement” from the Japanese final examination paper of July 2023 as the post-test material. The pre-test and post-test materials were corrected by the same two teachers according to the same grading criteria.

Teachers read each essay carefully and judged the quality of all essays on the basis of word choice, grammar, organization, sentence structure, and imagination. The two Japanese teachers involved in grading the essays utilized the above scoring method, and the scores of the pre and post-tested Japanese essays ranged from 0 to 10 points. The scores of these two teachers' essay ratings were averaged and used as the students' Japanese essay

grades. Before participating in the scoring of the essays, the two teachers were trained until they reached proficiency in using the method. The consistency of these two teachers' scores was 0.81.

IV. B. Experimental procedures

(1) Pre-test

A pre-test was conducted on the topic of “spring trip”. Students from both classes will be asked to describe their spring trip in Japanese. The score is 10 points out of a possible 800 words.

(2) Japanese Writing Exercise

(1) Experimental class: when the students in the experimental class conduct daily writing training, they first analyze the writing syntax based on the NN-LSVM model designed in this paper. Then the teacher uses the language generation model to touch up the students' texts, and after touching up, the logic between sentences, related terms, etc. are adjusted again, thus forming a comparison of the students' original texts and different texts modified by the model. Combined with the results of the students' syntactic analysis, the teacher gives his/her opinion on the modification.

(2) Control class: the teacher of the control class still adopts the daily writing training method. By reading each student's writing, the teacher gives revision opinions.

(3) Post-test

After a three-month training period, a Japanese final exam was conducted. The essay topic was “Vacation Arrangement,” which required about 800 words out of 10 points. The score of the essay on the test paper is used as the post-test grade.

IV. C. Experimental results

(1) T-test of Japanese writing pre and post-test scores of experimental and control classes

T-tests were conducted on the pre-test and post-test scores of Japanese writing of students in the experimental class and the control class to examine the effects of the syntactic analysis and language generation model designed in this paper on the subjects' Japanese writing scores as a whole, and the results are shown in Table 3. There is no significant difference between the pre-test scores of the experimental class and the control class ($P>0.05$), which indicates that there is not much difference in the level of Japanese writing between these two classes, which makes them suitable for a controlled experiment. And there is a significant difference ($P<0.05$) in the post-test scores, which indicates that the Japanese composition level of the two classes has widened the gap.

Table 3: Independent sample T test of (1) class and (2) class

	Laboratory class		Cross-reference class		DF	T	P
	M	SD	M	SD			
Pretest	4.26	2.13	4.28	2.64	1.26	-1.36	0.55
Posttest	5.66	1.63	4.31	2.61	2.56	-0.62	0.03

(2) T-test of pre and post-test scores of the experimental class and pre and post-test scores of the control class

T-tests were conducted on the pre and post-test scores of Japanese writing in the experimental class and the control class to test the effect of the techniques in this paper on the Japanese writing scores of the students in the experimental class. The results of the tests are shown in Table 4. There is a significant difference between the pre-test scores and post-test scores of the experimental class ($p<0.05$), while there is no significant difference between the pre-test scores and post-test scores of the control class ($p>0.05$). This indicates that the Japanese syntactic analysis and generation technique proposed in this paper has a greater contribution to the Japanese composition level of the students in the experimental class.

Table 4: Independent sample T test of pretest and posttest

	Pretest		Posttest		DF	T	P
	M	SD	M	SD			
Laboratory class	4.26	2.13	5.66	1.63	33	-5.71	0.000
Cross-reference class	4.28	2.64	4.31	2.61	52	-0.855	0.43

V. Conclusion

The study establishes an NN-LSVM-based Japanese dependency parsing model and a language generation model for improving students' Japanese writing.

(1) The NN-LSVM model's text-segment parsing accuracy can reach up to 89.61%, which has higher parsing accuracy compared with the SVM model and the NN-SVM model, and is more suitable for Japanese syntactic analysis.

(2) On the SJW, CLUE, and CGED datasets, the textual touch-up technique method proposed in this paper obtains better performance compared to the baseline method in various evaluation metrics. It shows that the method in this paper is able to better preserve the meaning of the original sentences in the process of embellishment, while improving the professionalism and readability of the syntactic structure of their expressions.

(3) The students' performance was significantly improved ($P<0.05$) when the two methods were applied simultaneously in the Japanese writing training process.

In conclusion, the two techniques proposed in this paper are more efficient in improving the quality of Japanese compositions and enhancing students' Japanese writing ability.

References

[1] Fanni, S. C., Febi, M., Aghakhanyan, G., & Neri, E. (2023). Natural language processing. In Introduction to artificial intelligence (pp. 87-99). Cham: Springer International Publishing.

[2] Goar, V., Yadav, N. S., & Yadav, P. S. (2023). Conversational AI for natural language processing: An review of ChatGPT. International Journal on Recent and Innovation Trends in Computing and Communication, 11(3s), 109-17.

[3] Juhn, Y., & Liu, H. (2020). Artificial intelligence approaches using natural language processing to advance EHR-based clinical research. Journal of Allergy and Clinical Immunology, 145(2), 463-469.

[4] Ali, A. A. S., & Shandilya, V. K. (2021). AI-Natural Language Processing (NLP). International Journal for Research in Applied Science and Engineering Technology, 9, 135-140.

[5] Xu, Q., Feng, Z., Gong, C., Wu, X., Zhao, H., Ye, Z., ... & Wei, C. (2024). Applications of explainable AI in natural language processing. Global Academic Frontiers, 2(3), 51-64.

[6] Sidorov, G., Velasquez, F., Stamatatos, E., Gelbukh, A., & Chanona-Hernández, L. (2014). Syntactic n-grams as machine learning features for natural language processing. Expert Systems with Applications, 41(3), 853-860.

[7] Cereda, P. R. M., Miura, N. K., & Neto, J. J. (2018). Syntactic analysis of natural language sentences based on rewriting systems and adaptivity. Procedia computer science, 130, 1102-1107.

[8] Zhang, X., Mao, R., & Cambria, E. (2023). A survey on syntactic processing techniques. Artificial Intelligence Review, 56(6), 5645-5728.

[9] Guetterman, T. C., Chang, T., DeJonckheere, M., Basu, T., Scruggs, E., & Vydiswaran, V. V. (2018). Augmenting qualitative text analysis with natural language processing: methodological study. Journal of medical Internet research, 20(6), e231.

[10] Choi, Y., Nguyen, M. D., & Kerr Jr, T. N. (2021). Syntactic and semantic information extraction from NPP procedures utilizing natural language processing integrated with rules. Nuclear Engineering and Technology, 53(3), 866-878.

[11] Vani, K., & Gupta, D. (2018). Unmasking text plagiarism using syntactic-semantic based natural language processing techniques: Comparisons, analysis and challenges. Information Processing & Management, 54(3), 408-432.

[12] Šubert, M., Novotný, M., Tykalová, T., Srpová, B., Friedová, L., Uher, T., ... & Rusz, J. (2023). Lexical and syntactic deficits analyzed via automated natural language processing: the new monitoring tool in multiple sclerosis. Therapeutic advances in neurological disorders, 16, 17562864231180719.

[13] Tong, X., Yu, L., & Deacon, S. H. (2024). A meta-analysis of the relation between syntactic skills and reading comprehension: a cross-linguistic and developmental investigation. Review of educational research, 00346543241228185.

[14] Ciampelli, S., de Boer, J. N., Voppel, A. E., Corona Hernandez, H., Brederoo, S. G., van Dellen, E., ... & Sommer, I. E. (2023). Syntactic network analysis in schizophrenia-spectrum disorders. Schizophrenia bulletin, 49(Supplement_2), S172-S182.

[15] Xolmurodovna, M. S. (2024). SYNTACTIC ANALYSIS AND THEIR TYPES IN ENGLISH LINGUISTICS. JOURNAL OF INTERNATIONAL SCIENTIFIC RESEARCH, 1(2), 163-166.

[16] Pandey, S., Pandey, S. K., & Miller, L. (2017). Measuring innovativeness of public organizations: Using natural language processing techniques in computer-aided textual analysis. International Public Management Journal, 20(1), 78-107.

[17] Sjöberg-Hawke, C. (2024, May). How can we motivate and engage our students to develop their technical writing skills?. In CHALMERS CONFERENCE ON TEACHING AND LEARNING 2024 (p. 29).

[18] Little, C. W., Clark, J. C., Tani, N. E., & Connor, C. M. (2018). Improving writing skills through technology‐based instruction: A meta‐analysis. Review of Education, 6(2), 183-201.

[19] Rose, H. (2019). Unique challenges of learning to write in the Japanese writing system. L2 writing beyond English, 66.

[20] Mulvey, B. (2016). Writing instruction: What is being taught in Japanese high schools, why, and why it matters. The Language Teacher, 40(3), 3-8.

[21] Joyce, T., & Masuda, H. (2018). Introduction to the multi-script Japanese writing system and word processing. Writing systems, reading processes, and cross-linguistic influences: Reflections from the Chinese, Japanese and Korean languages, 7, 179-199.

[22] Spence, L. K., & Kite, Y. (2018). Beliefs and practices of writing instruction in Japanese elementary schools. Language, Culture and Curriculum, 31(1), 56-69.

[23] Malik, A. R., Pratiwi, Y., Andajani, K., Numertayasa, I. W., Suharti, S., & Darwis, A. (2023). Exploring artificial intelligence in academic essay: higher education student's perspective. International Journal of Educational Research Open, 5, 100296.

[24] Waltzer, T., Pilegard, C., & Heyman, G. D. (2024). Can you spot the bot? Identifying AI-generated writing in college essays. International Journal for Educational Integrity, 20(1), 11.

[25] Schmohl, T., Watanabe, A., Fröhlich, N., & Herzberg, D. (2020, June). How artificial intelligence can improve the Academic Writing of students. In Conference Proceedings. The Future of Education 2020.
[26] Ryu Kitajima. (2016). Does the Advanced Proficiency Evaluated in Oral‐Like Written Text Support Syntactic Parsing in a Written Academic Text Among L2 Japanese Learners. Foreign Language Annals,49(3),573-595.
[27] Daisuke Kawahara & Sadao Kurohashi. (2014). A Fully-Lexicalized Probabilistic Model for Japanese Syntactic and Case Structure Analysis. Information and Media Technologies,9(4),695-711.
[28] KHILLARE S.A.,DHOKRAT A.V. & MAHENDER C.N. (2015). ANALYSIS OF QUESTION ANSWERING AND PARSING TECHNIQUES IN NATURAL LANGUAGE PROCESSING: A REVIEW. Advances in Computational Research,7(1),225-228.