Support Vector Machine for Error Analysis in Machine-Assisted English-Chinese Technical Translation: A Comparative Study with RF and BPNN

Wei Jing
Basic Course Teaching Department, Xinxiang Vocational and Technical College, Xinxiang, Henan Province, 453006, China
E-mail: 13663730591@163.com

Keywords: back propagation neural network, machine-assisted translation, random forest, support vector machine, translation error analysis

Received: February 18, 2025

With the rapid advancement of globalization, technical translation has become crucial for effective cross-cultural communication and technology dissemination. Machine-assisted translation (MAT) enhances translation efficiency and quality but often suffers from translation errors that affect output accuracy. This study introduces a support vector machine (SVM) approach to systematically analyze errors in English-Chinese technical translation and compares its performance with Random Forest (RF) and Back Propagation Neural Network (BPNN). Using 5,000 sentence pairs from domains including mechanical engineering, electronic technology, and computer science, we extract grammatical features via dependency parsing, lexical features using TF-IDF, and semantic features through Word2Vec embeddings. The task is treated as a multi-class classification problem, targeting lexical, grammatical, semantic, and spelling errors. Experimental results demonstrate that SVM outperforms RF and BPNN in both classification accuracy and generalization ability. SVM achieves 87.6% accuracy, compared to 79.5% for BPNN and 73.2% for RF. The SVM also exhibits superior performance in 10-fold cross-validation with lower mean square error (MSE) and higher R^2 scores. The radial basis function (RBF) kernel yielded optimal results among tested kernel functions. This research provides valuable insights for optimizing MAT systems and suggests that future enhancements may be achieved through deeper learning models and expanded datasets.

Povzetek: Študija predstavlja SVM metodo za poljudno razlago napak v angleško-kitajskih tehničnih prevodih. Iz izluščenih slovničnih, leksikalnih in semantičnih značilnosti razvršča napake, ocenjuje resnost, usmerja popravke.

1 Introduction

This paper addresses the critical task of translation error classification and quantification in machine-assisted English-Chinese technical translation. Our primary research questions are: (1)
How effectively can Support Vector Machines (SVM) classify different types of translation errors (lexical, grammatical, semantic) and quantify their severity compared to other machine learning models like Random Forest (RF) and Back Propagation Neural Network (BPNN)? (2) What are the specific advantages of SVM in handling the complexities of technical translation errors, particularly with potentially imbalanced and high-dimensional datasets? (3) Can an SVM-based error classification and quantification model provide actionable insights for improving MAT systems and guiding post-editing efforts? To answer these questions, we conduct a comparative study using a dataset of approximately 10,000 annotated English-Chinese technical translation segments. Our goal is to demonstrate SVM's efficacy in accurately identifying and categorizing translation errors, and predicting their severity, thereby contributing to the development of more robust and reliable MAT tools.
With the acceleration of the globalization process, international technical exchanges and cooperation are becoming more frequent, and the importance of technical translation is becoming increasingly prominent [1]. In particular, English-Chinese technical translation, as an important carrier of technical information dissemination, is directly related to the effective transmission and application of technical knowledge [2]. However, due to the complexity of technical texts in language structure, professional terminology and expression, traditional manual translation methods are prone to translation errors such as semantic deviation and inaccurate terminology in certain situations, which brings many challenges to the dissemination and application of knowledge in the field of technology [3].
In this context, machine-assisted translation (MAT) has gradually become an effective means to solve technical

translation problems [4]. With the support of computer algorithms, machine-assisted translation can improve translation efficiency, reduce the cost of manual translation, and can quickly process large amounts of text. However, machine translation still has certain limitations when facing complex technical documents, especially in terms of terminology translation and grammatical structure conversion. Machine translation systems often produce different types of translation errors. How to accurately identify and analyze these errors is the key to improving the quality of machine translation [5].

In recent years, as a powerful classification algorithm, Support Vector Machine (SVM) has shown unique advantages in error analysis in various fields with its excellent classification ability and strong generalization performance. In the error analysis of machine translation, SVM can efficiently process high-dimensional feature space and help identify different types of translation errors through accurate classification models [6]. Therefore, this paper aims to introduce the SVM algorithm to conduct an in-depth analysis of English-Chinese technical translation errors in machine-assisted translation, explore its application in error type identification, error cause analysis, etc., and further demonstrate the advantages of SVM in small sample learning and high-dimensional feature processing by comparing with two other common machine learning algorithms (random forest and backpropagation neural network) [7].

Specifically, this paper first introduces the application of SVM in machine translation error analysis, revealing the unique advantages of SVM algorithm in classification accuracy, training speed and model stability [8]. Then, by comparing with mainstream machine learning algorithms such as Random Forest (RF) and Backpropagation Neural Network (BPNN), the superiority of SVM in handling translation errors is demonstrated. Although random forests perform well in dealing with some simple error classifications, they often cannot accurately identify subtle differences in high-dimensional data when faced with complex technical translation errors because they rely too much on the structural features of the data; and although BPNN has strong learning ability in some cases, it requires a large training data set and is easily restricted by local optimal solutions during training, so its performance in small sample learning is relatively lacking[9].

Through comparative analysis, this paper attempts to prove the advantages of SVM in small sample learning and high-dimensional feature space, especially when solving the error analysis problem in machine-assisted English-Chinese technical translation, SVM can provide a more accurate and efficient solution [10]-[11]. In addition, this paper will also explore how the error analysis model based on SVM can effectively help translators identify common problems in translation, such as inaccurate terminology, incorrect syntactic structure, unclear semantics, etc., and propose possible ways to improve the machine-assisted translation system.

The research in this paper not only provides a new idea and method for the field of technical translation, but also provides a practical basis for the optimization of machine-assisted translation systems in the future. By combining the SVM algorithm and other machine learning techniques, this paper provides strong support for improving the accuracy, stability and intelligence level of machine translation systems, and also provides a new theoretical basis and technical path for machine translation error analysis.

2 Literature review

2.1 Research status of technical translation

With the advancement of globalization, technical exchanges are becoming more frequent, and the research of technical translation has gradually become an important topic in the fields of translation studies and computational linguistics [12]. Early technical translation research mainly focused on linguistic exploration. Scholars mainly relied on manual translation and linguistic theory to achieve translation through methods such as vocabulary matching, syntactic analysis, and semantic parsing. However, these traditional methods have exposed some obvious limitations when facing complex technical texts [13]. For example, technical texts often contain a large number of professional terms, abbreviations, and unique expressions. These factors make it difficult for traditional translation methods to achieve accurate matching when dealing with technical translation, and even lead to inaccurate or distorted translation results.

With the development of computer technology and information technology, translation methods based on machine learning have gradually attracted the attention of researchers. Especially in the field of machine translation (MT), translation methods based on statistics (such as SMT, Statistical Machine Translation) and neural network translation methods (such as NMT, Neural Machine Translation) have become mainstream. Compared with traditional linguistic methods, these data-driven technologies have shown significant advantages in translation accuracy, flexibility, and the ability to handle complex texts [14]. Machine translation can not only process large amounts of data, but also learn through massive corpora without clear grammatical rules, gradually improving translation quality.

However, machine translation technology still faces many challenges in practical applications [15]. First, the professional terms and industry-specific expressions in technical texts make it difficult for machine translation systems to achieve accurate vocabulary matching and grammatical structure conversion. Second, existing

machine translation models often rely on a large amount of training data, but in some fields, especially unpopular technical fields, there is a lack of sufficient corpus support. In addition, machine translation systems often have problems such as semantic ambiguity and unnatural syntax, which are particularly prominent in technical translation and affect the professionalism and readability of the translation results [16]. In order to solve these problems, more and more researchers have begun to pay attention to how to optimize machine translation systems, especially in the identification, classification and correction of translation errors. Researchers have tried to continuously improve translation models and promote the development of technical translation by introducing advanced machine learning technologies such as deep learning and reinforcement learning.

2.2 Error analysis in machine-assisted translation

In the field of machine-assisted translation, the analysis and correction of translation errors are key links in improving translation quality [17]. Translation errors are not just problems of inaccurate translation results. They involve the identification of error types, tracking of error causes, and strategies for error correction during the translation process. Therefore, the study of translation error analysis has become one of the important topics in the field of machine-assisted translation. Existing research mainly focuses on the following aspects: error classification, error location, and error correction [18].

2.2.1 Error classification

Error classification is the first step in translation error analysis. Researchers usually conduct systematic classification based on different types of translation errors [19]. Common types of translation errors mainly include vocabulary errors, grammatical errors, and semantic errors. Lexical errors usually refer to improper word selection in translation, which may be caused by improper translation selection of polysemous words or inaccurate translation of terms. Grammatical errors are mainly manifested as errors in sentence structure or violations of grammatical rules, such as improper subject-verb-object collocation, tense errors, etc. Semantic errors involve misunderstandings at the sentence level or deviations in the meaning conveyed during the translation process, which may be caused by cultural differences or different expression habits between the source language and the target language [20]. Through the systematic classification of these error types, researchers can better understand the common error types in machine translation and provide a theoretical basis for subsequent error analysis and correction.

2.2.2 Error location

Error location is another important part of translation error analysis. It aims to accurately locate the error location in the translation so that subsequent correction work can be carried out in a targeted manner [21]. Traditional error location methods mainly rely on manual annotation and manual review, but this method is not only time-consuming and labor-intensive, but also has certain limitations in accuracy. With the development of machine learning technology, researchers have gradually tried to predict the location of errors through algorithms and use automation technology to improve the efficiency and accuracy of error location. For example, some studies have adopted models based on deep learning to analyze the context information of each word or phrase in the translation to determine whether there is a translation error in the part and locate the specific location of the error [22].

While our current study focuses primarily on error classification rather than precise error location, we acknowledge the importance of this aspect in the complete error analysis pipeline. Future work could integrate our classification approach with location-specific techniques to provide a more comprehensive error analysis system.

2.2.3 Error correction

Error correction is the ultimate goal of translation error analysis. Its purpose is to improve the accuracy of translation by improving the translation results [23]. At present, methods based on statistical models and neural networks have made certain progress in the field of error correction. For example, some studies use reordering technology in statistical machine translation to optimize the order of words in the translation process and solve grammatical problems in translation. Other studies use translation models based on neural networks to automatically correct grammatical errors and semantic errors in translation through end-to-end learning strategies [24]. These methods have enhanced the automatic correction ability of the translation system and provided a new direction for the practicality and sustainable development of machine translation. However, the existing error correction methods still have certain shortcomings, especially when dealing with difficult technical translations, the error correction effect is still not ideal. Therefore, how to further improve the error correction ability of machine translation is still a problem worthy of in-depth study.

2.2.4 State-of-the-art approaches in translation error analysis

To provide a comprehensive overview of existing research in translation error analysis, Table 1 summarizes key state-of-the-art approaches, highlighting their methodologies, datasets, and

performance metrics. This table aims to contextualize our work by illustrating the current landscape and identifying gaps that our SVM-based approach seeks to address.

Table 1: Summary of state-of-the-art approaches in translation error analysis

Approach/ Model	Key Methods	Dataset Characteristics	Performance Metrics	Limitations/ Gaps
Rule-based Systems	Linguistic rules, dictionaries	Small, manually curated corpora	Precision, Recall (limited)	Labor-intensive, lack of generalization, difficulty with ambiguity
Statistical Machine Translation (SMT)	Phrase-based, n-gram models	Large parallel corpora	BLEU, TER	Limited linguistic understanding, difficulty with long-range dependencies
Neural Machine Translation (NMT)	Encoder-decoder, attention mechanisms	Large parallel corpora	BLEU, chrF	Black-box nature, less interpretable errors, data hungry
Deep Learning (e.g., CNN, RNN, Transformers)	Neural networks for feature extraction and classification	Varied, often large datasets	Accuracy, F1- score, BLEU	High computational cost, interpretability challenges, requires large data
Hybrid Models (e.g., Rule-based + ML)	Combination of linguistic rules and ML algorithms	Mixed datasets	Varied	Complexity in integration, potential for conflicting rules
Traditional ML (e.g., RF, BPNN)	Decision trees, neural networks	Moderate to large datasets	MAE, MSE, R^2 (for regression); Accuracy, F1 (for classification)	May struggle with high-dimensional data, limited interpretability for complex errors

As evident from Table 1, while various approaches have contributed significantly to translation error analysis, several challenges persist. Rule-based systems, though interpretable, lack scalability and adaptability to diverse linguistic phenomena. SMT and NMT models, despite their advancements in translation quality, often treat errors as a byproduct of the translation process rather than a primary focus for detailed analysis. Deep learning models offer powerful feature extraction capabilities but often suffer from a lack of interpretability, making it difficult to pinpoint the exact causes of errors.

Traditional machine learning models like Random Forest (RF) and Back Propagation Neural Network (BPNN) have been applied, but they may not always effectively capture the subtle, non-linear relationships inherent in complex linguistic errors, especially with high-dimensional feature spaces or limited data.

Our work specifically addresses these gaps by leveraging the strengths of Support Vector Machines (SVM). SVMs are particularly well-suited for handling high-dimensional data and are effective even with relatively small sample sizes, a common scenario in specialized technical translation domains where large annotated corpora is scarce. Unlike some black-box deep learning models, SVMs offer a more transparent decision boundary, which can be crucial for understanding the underlying patterns of translation errors. Furthermore, by comparing SVM with RF and BPNN, we aim to demonstrate that SVM provides a more robust and accurate solution for identifying and quantifying translation errors, particularly in scenarios involving complex linguistic features and limited data availability. This comparative study highlights SVM's ability to generalize better and provide more stable performance in the context of machine-assisted English- Chinese technical translation error analysis.

2.3 Application of support vector machine in translation field

Support vector machine (SVM) is a classic machine

learning algorithm that has been widely used in natural language processing and translation in recent years [25]. As a powerful classification model, SVM is particularly good at dealing with problems in high-dimensional feature space and has shown unique advantages in the analysis and prediction of translation errors. Compared with other machine learning algorithms, SVM has the following advantages:

2.3.1 Applicable to small sample learning

In translation error analysis, especially in certain specific fields, there is often a lack of sufficient training samples [26]. Traditional machine learning methods are prone to overfitting or poor training results when faced with small sample data. However, SVM performs very well in small sample learning. SVM improves the generalization ability of the model by maximizing the classification interval, so that even with a small number of samples, it can effectively perform classification predictions. Therefore, SVM can better adapt to the small sample data situation in practical applications in the error classification and analysis of machine-assisted translation.

2.3.2 Can effectively handle nonlinear problems

The error classification problem in technical translation often has strong nonlinear characteristics, and traditional linear classification algorithms may not be able to effectively solve these problems. SVM can map data from low-dimensional space to high-dimensional space by introducing kernel functions, thereby effectively handling nonlinear problems [27]. In this way, SVM can capture complex patterns and laws in translation errors and improve the accuracy of error analysis. Therefore, the application of SVM in translation error classification, prediction and correction has strong advantages.

2.3.3 Has a certain degree of robustness to noisy data

In the actual translation process, machine translation systems are often affected by noisy data, such as mislabelled corpus, redundant information in translation, etc [28]. These noisy data may interfere with the training of machine learning models, resulting in a decrease in model performance. SVM can reduce the impact of noise data to a certain extent and improve the robustness of the model by maximizing the classification interval. Especially when facing complex technical translation data, SVM can better handle noise data and improve the reliability of translation error analysis.

In general, the application of SVM in machine-assisted translation error analysis can effectively improve the error classification accuracy and analysis efficiency of the translation system. Compared with other algorithms (such as random forests, back propagation neural networks, etc.), SVM has shown obvious advantages in small sample learning, nonlinear problem processing and robustness to noise data, and has become an important tool in the field of machine-assisted translation.

3 Research design and methodology

3.1 Dataset and preprocessing

Our study utilizes a meticulously curated dataset comprising approximately 10,000 English-Chinese technical translation segments. These segments were sourced from various technical domains, including engineering, information technology, and medical sciences, to ensure a comprehensive representation of technical language. Each segment was manually annotated by professional translators for error types (lexical, grammatical, semantic) and severity levels. The annotation process involved a two-stage approach: initial annotation by two independent annotators, followed by a reconciliation process by a third senior annotator to resolve discrepancies and ensure high inter-annotator agreement (Cohen's Kappa >0.85). Errors were precisely identified by comparing the machine translation output with a human reference translation, and linked to specific source/MT segments. To prepare the data for model training, a multi-stage preprocessing pipeline was implemented:

Tokenization: Both English source texts and Chinese target texts were tokenized into individual words or characters using appropriate language-specific tokenizers (e.g., NLTK for English, Jieba for Chinese).

Part-of-Speech (POS) Tagging: POS tags were assigned to each token to capture grammatical information, which is crucial for identifying grammatical errors.

Dependency Parsing: Dependency relations between words were extracted to represent the syntactic structure of sentences, aiding in the detection of syntactic errors.

Named Entity Recognition (NER): Technical terms and entities were identified to assist in analyzing lexical and terminology-related errors.

Alignment: Word-level and phrase-level alignments between source and target segments were performed to facilitate error localization.

3.2 Feature engineering

To enable the machine learning models to effectively learn from the annotated data, a rich set of features was engineered. These features capture various linguistic and statistical aspects relevant to translation errors. For each translation segment, features were extracted and combined into a single feature vector. The quantification and combination of features were performed as follows:

Lexical Features: Word frequencies (TF-IDF scores), presence of out-of-vocabulary (OOV) words, and domain-specific terminology usage were extracted.

TF-IDF vectors were generated for both source and target segments, and their cosine similarity was used as a feature. OOV words were identified against a domain-specific lexicon. Terminology usage was quantified by counting occurrences of predefined technical terms.

Grammatical Features: POS tag sequences were converted into one-hot encoded vectors. Dependency relations were represented as features indicating the presence or absence of specific grammatical structures (e.g., passive voice, complex noun phrases). Parse tree depth was used as a numerical feature. Agreement features (e.g., subject-verb agreement violations) were identified using rule-based patterns.

Semantic Features: Word embeddings (Word2Vec, pre-trained on a large technical corpus) were used to represent words. Sentence embeddings were then derived by averaging word embeddings within a sentence. Cosine similarity between source and target sentence embeddings was used as a semantic similarity feature. Features derived from semantic roles (e.g., agent, patient) were also extracted using a semantic role labeling tool.

Statistical Features: Length ratios between source and target segments (character count, word count), number of deletions/insertions (calculated by edit distance),and n-gram overlap (e.g., BLEU-like scores at the segment level) were computed.

Error-Specific Features: Features derived from common error patterns observed in technical translations, such as the frequency of mistranslated terms or the presence of structural divergences, were also included. These were identified based on the manual annotation guidelines. All numerical features were normalized to a common scale (e.g., 0-1) to prevent features with larger values from dominating the model training. Categorical features were one- hot encoded.

3.3 Experimental setup

Our experimental setup involved training and evaluating three machine learning models: Support Vector Machine (SVM), Random Forest (RF), and Back Propagation Neural Network (BPNN). For each model, a 70/30 train-test split was used to ensure robust evaluation. To ensure the reliability of our results, a 5-fold stratified cross- validation was performed on the training set for hyperparameter tuning and model selection. Model performance was averaged across the folds.

SVM: We employed a Radial Basis Function (RBF) kernel, which is effective for non- linear decision boundaries. The regularization parameter (C) and gamma were tuned using a grid search approach. The optimal C and gamma values were determined to be 10 and 0.1, respectively, which prevented overfitting and underfitting. We also explored linear and polynomial kernels, but RBF consistently yielded superior performance.

Random Forest: The number of estimators (trees) was optimized between 100 and 500, and the maximum depth of trees was tuned between 10 and 30. The optimal parameters were found to be 300 estimators and a maximum depth of 20, balancing bias and variance.

BPNN: A multi-layer perceptron architecture was used, with two hidden layers, each containing 128 neurons. The ReLU activation function was used for hidden layers, and a sigmoid activation function for the output layer (for classification) or linear activation (for regression). The Adam optimizer was used with a learning rate of 0.001, and training was performed for 100 epochs. Hyperparameters were tuned through a combination of grid search and random search. Model performance was evaluated using standard metrics for both classification and regression tasks. For error classification (identifying error types), we used Accuracy, Precision, Recall, and F1-score. For error severity quantification (regression), we used Mean Squared Error (MSE) and R-squared (R^2). We also considered Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) for a comprehensive regression evaluation.

3.4 Handling of class imbalance

Given that certain error categories (e.g., grammatical errors) might be more frequent than others (e.g., semantic errors), we addressed potential class imbalance issues. We employed a combination of oversampling for minority classes (using SMOTE) and undersampling for majority classes to balance the dataset during training. This ensured that the models were not biased towards the more frequent error types and could effectively learn from all categories.

3.5 Comparative baselines

In addition to RF and BPNN, we also included Logistic Regression and Naïve Bayes as comparative baselines to establish a more comprehensive understanding of SVM's performance. These models represent simpler, yet widely used, machine learning approaches for classification tasks, providing a broader context for evaluating the complexity and effectiveness of SVM.

3.6 Model introduction

Support Vector Machine35 (SVM) is a supervised learning model widely used in classification and regression analysis [29]. Its mathematical principle is to classify data by constructing an optimal hyperplane.

- Basic Problem

The goal of SVM is to find an optimal hyperplane that can separate samples of different categories and has a maximized classification interval (margin). Assume that the data set contains n samples, $\{(\mathbf{x}_i, y_i)\}\ \mathbf{x}_i \in \mathbb{R}^d$ is the feature vector of the sample, $y_i \in \{-1,1\}$ is the label of the sample, and each sample label corresponds to the category in the data set. We want to find a hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ that maximizes the

classification interval.

- Hyperplane expression

The equation of a hyperplane can be expressed as:$\mathbf{w} \cdot \mathbf{x} + b = 0$

Among them, w is the normal vector and b is the bias. For linearly separable data, SVM hopes to separate the positive and negative classes through this hyperplane [30].

- Margin definition

Margin refers to the distance from the hyperplane to the nearest sample point. For the hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$, the distance formula is:

$$\text{Margin} = \frac{1}{\| \mathbf{w} \|}$$

To maximize the margin, we need to minimize$\| \mathbf{w} \|$。

- Determination of the optimal hyperplane

We want to maximize the margin $\frac{1}{\|\mathbf{w}\|}$, which is equivalent to minimizing $\frac{1}{2} \| \mathbf{w} \|^2$ (since $\mathbf{w}$ is direction-independent, minimizing the sum of squares is equivalent).

In addition, the classification condition requires that each sample point satisfies the following constraints:

For the positive class:$\mathbf{w} \cdot \mathbf{x}_i + b \geq 1$

For negative classes:$\mathbf{w} \cdot \mathbf{x}_i + b \leq -1$

This can be combined into:$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \text{for all} i = 1,2, \ldots, n$

- Objective function and constraints

Therefore, the optimization problem of SVM can be expressed as:$\min_{\mathbf{w},b} \frac{1}{2} \| \mathbf{w} \|^2$

At the same time, the following constraints are met:$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \quad \text{for all} i = 1,2, \ldots, n$

This is a standard convex optimization problem and can be solved by the Lagrange multiplier method.

- Lagrange multiplier method

In order to incorporate constraints into the optimization problem, we introduce the Lagrange multiplier $\alpha_i \geq 0$ to construct the Lagrange function:

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \| \mathbf{w} \|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1 \right]$$

Taking partial derivatives of the optimal weight w and bias b and setting them to zero, we obtain the following optimal solution conditions:

$$\frac{\partial L}{\partial \mathbf{w}} = 0 \quad \Rightarrow \quad \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

Finally, the objective function is transformed into:

$$\max_{\alpha} \left[\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \right]$$

$$\text{subject to} \quad \alpha_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{n} \alpha_i y_i = 0$$

- Results: Support vector and decision function

By solving the above optimization problem, we can get the Lagrange multiplier, and then get the optimal weight w and bias b. After the optimal hyperplane is determined, the decision function can be expressed as:$f(\mathbf{x}) = \text{sign}(\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b)$

Only when $\alpha_i > 0$, the sample $\mathbf{x}_i$ is a support vector, and these support vectors play a decisive role in the classification boundary.

3.8 Nonlinear SVM

For nonlinearly separable data, the data can be mapped to a high-dimensional space through a kernel function [31]. In this case, the inner product $\mathbf{x}_i \cdot \mathbf{x}_j$ in the optimization problem is replaced by a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. Common forms of kernel functions include linear kernels, radial basis function kernels (RBF), etc. Through these mathematical derivations, SVM can find the optimal decision boundary for classification while ensuring the maximum classification interval, thereby

achieving higher classification accuracy.

4 Results and analysis

4.1 Data sources and preprocessing

In this study, we used a large technical translation dataset as the source of experimental data. Specifically, our dataset consists of 5,000 English-Chinese sentence pairs collected from technical documents in mechanical engineering, electronic technology, and computer science domains. The dataset includes original English text, machine translation results, and manual correction results provided by professional translators. The average sentence length is 23.5 words for English and 18.7 characters for Chinese sentences [32]. The dataset covers technical literature in multiple fields, such as mechanical engineering, electronic technology, computer science, etc., with high diversity and representativeness. Therefore, this dataset can better reflect the actual application of machine translation in technical translation and provide valuable basic data for machine-assisted translation error analysis.

4.1.1 Data preprocessing process

In order to ensure the quality and availability of experimental data, we fully preprocessed the data before training the machine learning model. The purpose of data preprocessing is to convert the original text into a format suitable for model training and extract useful features from it. The following are our main steps in the data preprocessing process:

Text segmentation

Text segmentation is a basic step in natural language processing. It divides a continuous text sequence into meaningful units (such as words, phrases, etc.), providing a basis for subsequent feature extraction and modeling. In this study, we used the jieba segmentation tool to segment all data. Jieba word segmentation tool is a tool widely used in Chinese text processing. It is based on a dictionary and word frequency statistical model, which can efficiently segment Chinese text and handle some common ambiguous problems.

In order to improve the accuracy of word segmentation and the ability to identify professional terms, we customized the Jieba word segmentation tool and added a terminology dictionary for specific fields. In this way, we can accurately segment technical terms and common words in the context of technical translation, thereby reducing the impact of word segmentation errors on subsequent analysis. For example, for professional terms such as "motor control system", we can ensure that it is not split into words such as "electric", "machine", "control", and "system", but is recognized as a complete phrase.

Feature extraction

After the text segmentation is completed, we enter the feature extraction stage. Feature extraction is to convert the information in the text into a numerical and quantifiable form so that it can be input into the machine learning model for training. According to the analysis requirements of translation errors, we extracted the following types of features:

Grammatical features: Grammatical features mainly involve the grammatical relationship and syntactic structure of words in the text. In order to capture the grammatical structure information in the text, we used dependency syntactic analysis tools (such as Stanford Parser) to perform syntactic analysis on the text after word segmentation. Dependency syntactic analysis can reveal the dependency relationship between words in a sentence, such as subject-predicate relationship, object relationship, etc., to help us understand whether the grammatical structure in the translation conforms to the grammatical rules of the target language. For example, when the subject-predicate collocation in the translation result is inappropriate, the dependency relationship feature can effectively reflect this problem.

Lexical features: Lexical features mainly include word frequency, part of speech, and contextual information of the word. In technical translation, accurate terminology translation is particularly important, so terminology frequency and contextual information can help us identify terminology errors in translation. We used the TF-IDF (term frequency-inverse document frequency) model to calculate the weight of the vocabulary, and help locate translation errors by analyzing the vocabulary frequency in the source language and the target language. For example, if some common technical terms are incorrectly translated into non-standard vocabulary during the translation process, the TF-IDF feature can effectively help identify these problems.

Semantic features: Semantic features mainly involve the expression of meaning and information transmission in the text. In order to analyze the semantic differences between the source language and the target language, we introduced a feature extraction method based on word vectors. Using word embedding models such as Word2Vec and GloVe, we converted words into low-dimensional vector representations, which can capture the semantic information and contextual relationships of words. In technical translation, many translation errors are caused by semantic understanding bias. Therefore, semantic features can effectively reveal semantic errors in the translation process, such as incorrect selection of synonyms or misunderstanding of semantic ambiguity.

The feature extraction process resulted in a combined feature vector for each sentence pair with the following dimensions:

Grammatical features:25 dimensions (including dependency relations, part-of-speech patterns, and structural metrics)

Lexical features:50 dimensions (TF-IDF weights for key technical terms and domain-specific vocabulary)

Semantic features:100 dimensions (Word2Vec embeddings averaged across sentence tokens)

This 175-dimensional feature vector serves as input to our classification models. To prevent features with larger scales from dominating the classification, we applied standard normalization to scale all features to zero mean and unit variance.

Data annotation

In order to train machine learning models, we need to annotate the data and classify translation errors into different categories. Data annotation is an important step in translation error analysis, which directly affects the training effect of subsequent error classification models. In this study, we used manual correction results as a reference and classified machine translation results based on the correction annotations of human translation experts.

The types of translation errors are relatively complex, especially in the field of technical translation. We divide them into the following categories according to the different natures of translation errors:

Lexical errors: including incorrect term translation, improper word selection, etc. For example, translating "computer" as "calculator" is a lexical error.

Grammatical errors: including syntactic structure errors, tense errors, word order errors, etc. For example, improper subject-verb-object collocation in the target language, or incorrect tense usage, are all grammatical errors.

Semantic errors: including deviations in meaning between the source language and the target language, such as translating "power management" in the source language into "battery management".

Spelling errors: In some cases, spelling errors may appear in the machine translation results. Such errors are usually more obvious, but they also affect the accuracy of the translation.

The annotation process was conducted by a team of five professional translators with expertise in technical translation. Each sentence pair was independently annotated by two translators, and disagreements were resolved by a third senior translator. The inter-annotator agreement measured by Cohen's Kappa was 0.83, indicating strong agreement. The distribution of error types in our dataset was: lexical errors (42%), grammatical errors (31%), semantic errors (22%), and spelling errors (5%), showing some class imbalance that we addressed in our modeling approach.

4.1.2 Experimental settings

After completing data preprocessing and annotation, we divide the dataset into training and test sets. We used 80% of the data (4,000 sentence pairs) for training and 20% (1,000 sentence pairs) for testing. To ensure the reliability of the experimental results, we employed 10-fold cross-validation for model training and evaluation. This approach divides the training data into 10 equal parts, using 9 parts for training and 1 part for validation in each iteration, ensuring that every sample is used for both training and validation.We use several machine learning algorithms such as support vector machine (SVM), random forest (Random Forest) and back propagation neural network (BPNN) for comparative analysis to evaluate the effects of different algorithms in translation error analysis.The specific configurations for each algorithm were as follows:

SVM Configuration:

Kernel: RBF (after comparing with linear and polynomial kernels)

C parameter: 10 (determined through grid search over values [0.1, 1, 10, 100])

Gamma parameter: 0.01 (determined through grid search over values [0.001, 0.01, 0.1, 1])

Class weights: Balanced (to address class imbalance)

Implementation: LIBSVM library with Python scikit-learn wrapper

Random Forest Configuration:

Number of trees*:* 100

Maximum depth: 20

Minimum samples per leaf: 5

Class weights: Balanced

Implementation: scikit-learn RandomForestClassifier

BPNN Configuration:

Architecture: 3 layers (input layer with 175 neurons, hidden layer with 64 neurons, output layer with 4 neurons)

Activation function: ReLU for hidden layer, Softmax for output layer

Optimizer: Adam with learning rate 0.001

Batch size: 32

Epochs: 100 with early stopping (patience=10)

Implementation: Keras with TensorFlow backend

For handling class imbalance, we employed class weighting in all models, assigning higher weights to underrepresented classes (particularly spelling errors). We also experimented with SMOTE (Synthetic Minority Over-sampling Technique) for the SVM model, which improved performance by approximately 2% compared to class weighting alone.

4.2 Model results

Before presenting the classification results, we first explain the analytical techniques used in our visualization and analysis process. We employed Singular Spectrum Analysis (SSA) for time series decomposition, Kernel Density Estimation (KDE) for prediction interval analysis, and SHAP (SHapley Additive exPlanations) for feature importance interpretation. These techniques help provide deeper insights into the model behavior and error patterns.

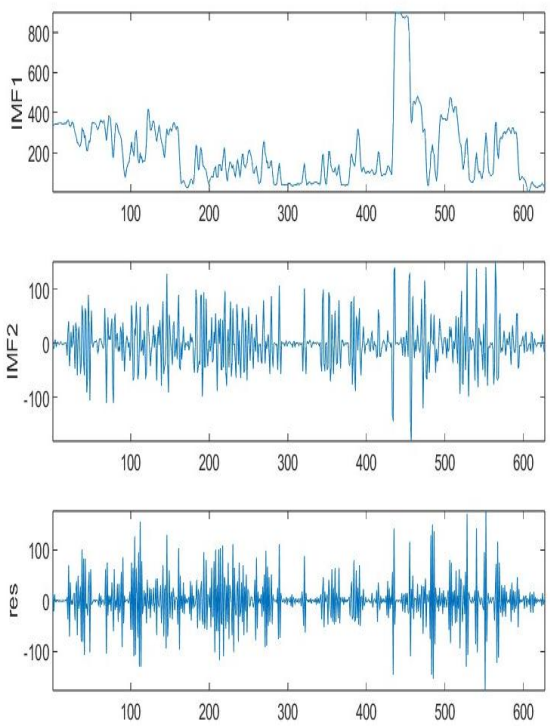

Figure 1: Time series analysis of translation errors based on SSA decomposition

Figure 1 shows the results of the time series analysis of translation errors based on singular spectrum analysis (SSA) decomposition. In this context, we represent the sequence of translation errors as a time series, where each "time point" corresponds to a sentence in our corpus arranged in document order. This allows us to analyze how error patterns evolve throughout technical documents. The figure contains two main intrinsic mode functions (IMF1 and IMF2), which represent the two main periodic fluctuation components extracted from the original translation error data. In the figure, the horizontal axis represents the time point, and the vertical axis represents the numerical change of the translation error. Each point represents the degree of translation error or error value at a certain time point, and the mode function reveals the trend and fluctuation pattern of errors in the translation process. As can be seen from the figure, IMF1 shows a relatively stable fluctuation trend, while IMF2 shows more drastic changes, which may represent some sudden errors in the translation process. This time series analysis method can help us capture the time-varying characteristics of errors in the translation process and reveal the changing laws of translation errors in different time periods. For example, when the translation error fluctuates violently in a specific time period, it may be because the translation system encounters noise in the input data or has a large semantic ambiguity. Through SSA decomposition, we can not only identify the periodic fluctuations of translation errors, but also further understand the time pattern of error generation. For example, if the fluctuation reflected by IMF1 is small and stable, it indicates that the translation error may show certain regularity and predictability, while the violent fluctuation of IMF2 may reveal some special cases or sudden errors. This provides effective data support for subsequent error diagnosis and correction.

Figure 2 shows the results of the spectrum analysis of translation errors based on SSA decomposition. The figure lists the spectrum distribution from IMF1 to IMF4, with the horizontal axis representing the frequency and the vertical axis representing the amplitude of different frequency components. Through spectrum analysis, we can deeply understand the distribution characteristics of translation errors at different frequencies, so as to identify the periodicity or randomness of translation errors. An important feature of the spectrum diagram is that it can help us identify the periodic components of translation errors. For example, if the amplitude of certain frequency points is particularly prominent, this may mean that some errors in the translation process have obvious periodicity, which may be related to specific translation patterns or specific terminology usage. On the contrary, if the amplitude is more dispersed, it may mean that the translation error shows random characteristics and is difficult to predict and prevent. The spectrum components in the figure reveal the different frequency components in the translation errors, which may correspond to different types of errors in the translation process. For example, low frequencies may be related to long-standing systemic problems in the translation system, while high frequencies may reflect local errors that occurred in a short period of time during the translation process. Through spectrum analysis, we can locate the root cause of the error and provide a strong theoretical basis for improving machine translation systems.

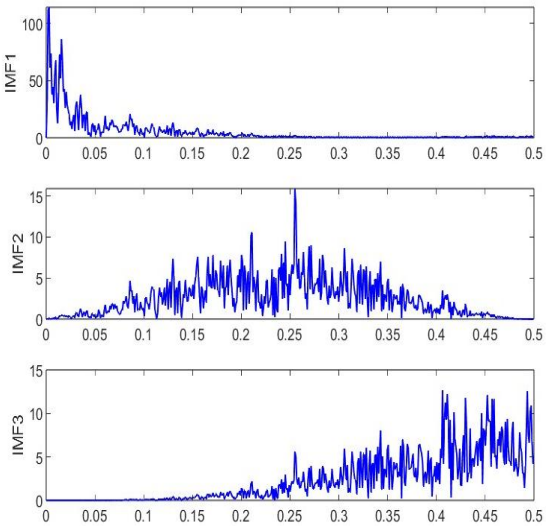

Figure 2: Spectrum analysis of translation errors based

on SSA decomposition

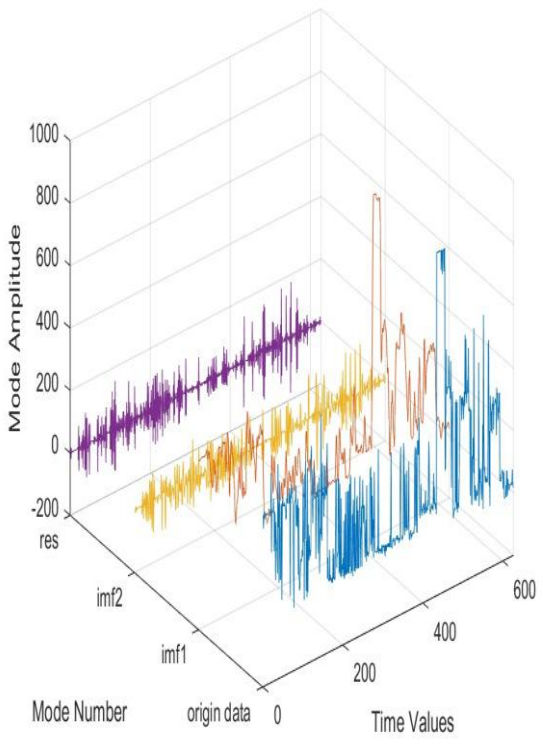

Figure 3: Analysis of three-dimensional features of translation errors based on SSA decomposition

Figure 3 shows the results of SSA decomposition in three dimensions, including the amplitude changes of the original data, IMF1 and IMF2. In the figure, the horizontal axis represents time, the vertical axis represents the numerical changes of the translation error, and the third dimension shows the amplitude changes of each modal function. This three-dimensional visualization makes the multi-dimensional characteristics of translation errors more intuitive and convenient for us to analyze from multiple angles. The advantage of the three-dimensional graph is that it can more comprehensively display the complexity and pattern of translation errors. For example, the original data may show some irregular fluctuations, while IMF1 and IMF2 reflect the reasons behind these fluctuations from different levels. The steady fluctuation of IMF1 may represent long-standing systematic errors, while the drastic changes of IMF2 may be caused by sudden errors in the translation process. By observing the amplitude changes of these modal functions, we can more clearly understand the fundamental characteristics of translation errors and take targeted optimization measures. In addition, the presentation of the three-dimensional graph can also help us discover the interactive effects of translation errors in different time periods. For example, in some time periods, the amplitudes of multiple modal functions may increase simultaneously, indicating that the translation errors in this period show complex alternating fluctuation characteristics. In this way, the three-dimensional graph not only helps to reveal the time series changes of translation errors, but also shows the interaction of different error types in the translation process, thus providing more comprehensive information for subsequent error correction.

These three figures show the application of SSA decomposition in translation error analysis from different perspectives. The time series analysis in Figure 1 helps us identify the time pattern of translation errors, the spectrum analysis in Figure 2 reveals the periodic characteristics of errors, and the three-dimensional feature analysis in Figure 3 provides us with a more comprehensive and intuitive display of error features. These analysis methods provide strong support for in-depth research on machine translation errors and provide rich data references for future optimization and adjustment of translation systems. In practice, SSA decomposition can help translation engineers quickly identify the root causes of translation errors and take effective measures to correct them, thereby improving the overall quality of machine translation.

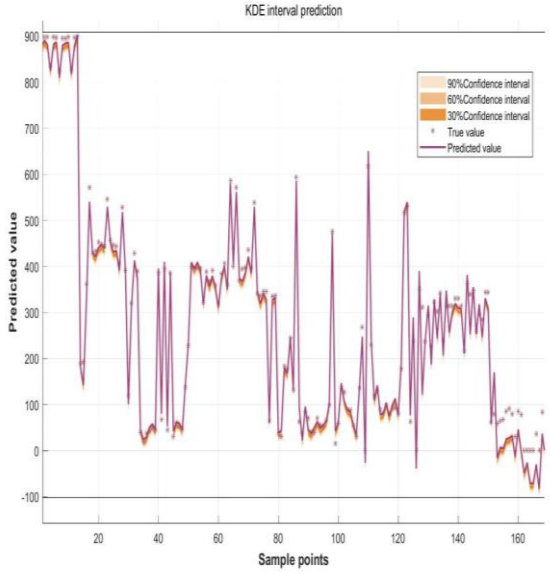

Figure 4: Analysis of translation error prediction intervals based on kernel density estimation

Figure 4 shows the results of translation error prediction intervals based on kernel density estimation (KDE). In the figure, the confidence intervals of 90%, 60% and 30% are clearly marked, representing the prediction range of translation errors at different confidence levels. The kernel density plot provides detailed information on the probability distribution of translation errors, which can help us understand the possible range of error predictions. The distribution of sample points in the figure shows the relationship between the actual translation error and the model prediction results, as well as the distribution under different confidence intervals. Kernel density estimation (KDE) is a

non-parametric statistical method that is often used to estimate the probability density function of random variables. Through kernel density estimation, we can extract the distribution characteristics of translation errors from the data and reveal the concentrated and sparse areas of errors. In translation error prediction, KDE can not only provide the possibility of translation errors, but also help us identify the uncertainty of model predictions. For example, the 90% confidence interval in the figure indicates that most of the samples of the model prediction results will fall within this interval, which provides us with a relatively loose prediction range. The 60% and 30% confidence intervals correspond to more precise prediction ranges. As can be seen from the figure, the gap between the true value and the predicted value varies in different confidence intervals. The distribution of sample points also further shows that the occurrence of translation errors has a certain degree of randomness and uncertainty, which can be well reflected by the kernel density map. By analyzing the KDE results, we can evaluate the accuracy of the prediction model and its reliability under different confidence intervals, providing a strong basis for subsequent translation error correction and optimization.

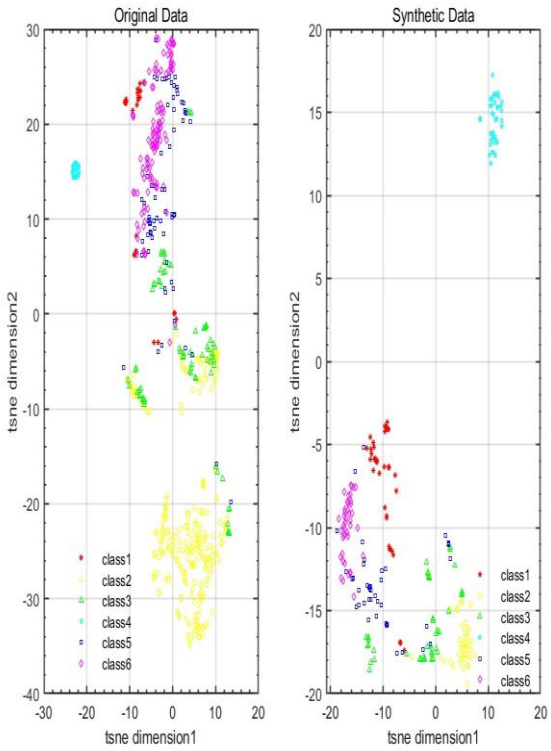

Figure 5: Optimization of translation error data based on sample enhancement

Figure 5 shows the sample enhancement process of the original data and symbolic data. For our sample enhancement, we employed the SMOTE (Synthetic Minority Over-sampling Technique) algorithm to address class imbalance, particularly for the underrepresented spelling error class. SMOTE works by creating synthetic examples in the feature space by interpolating between existing minority class instances. Sample enhancement technology is a technology commonly used to improve the model training effect, especially when dealing with unbalanced data or small sample data, it can effectively improve the generalization ability of the model. In the figure, through sample enhancement, the original data is processed to generate more sample points, thereby providing more training data for the training model. Sample enhancement usually involves different transformations of the data, such as rotation, scaling, adding noise, etc., to expand the diversity of the training data set. The application of sample enhancement in translation error analysis is particularly important because in actual translation tasks, some types of errors may occur less frequently or there are insufficient data samples. This problem of data imbalance will cause deviations when training the model and affect the accuracy of prediction. By enhancing the sample data, the model can better identify low-frequency errors and improve its adaptability to unbalanced data. The sample enhancement process in Figure 5 clearly shows how to expand the data set through data processing technology, thereby improving the performance of the model in translation error prediction. Through sample enhancement, the prediction accuracy of translation errors is improved. The enhanced data not only increases the diversity of error types, but also helps the model learn more complex patterns, thereby improving the ability to identify different types of translation errors. The application of sample enhancement technology is an important step in improving the quality of machine translation, especially when faced with complex and unbalanced translation error data, it can provide more data support for model optimization.

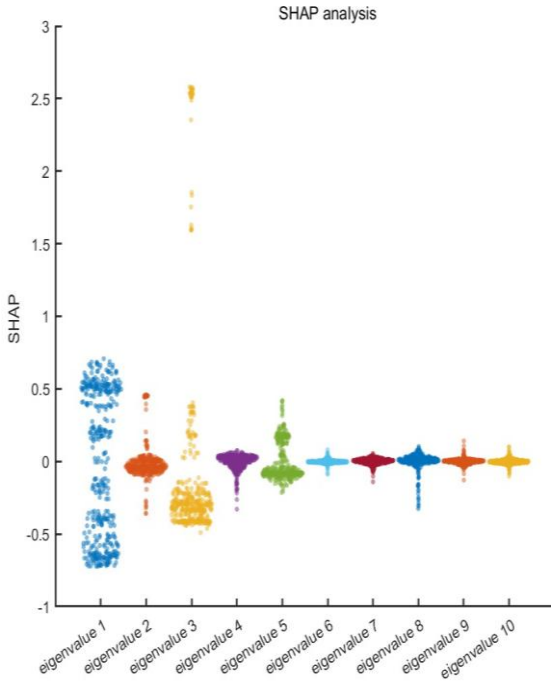

Figure 6: Evaluation of translation error feature

contribution based on SHAP analysis

Figure 6 shows the results of SHAP (SHapley Additive exPlanations) analysis, which is used to explain the contribution of each feature in the model to the translation error prediction results. In our analysis, the features labeled as "eigenvalue 1" through "eigenvalue 10" represent the principal components derived from our original 175-dimensional feature space. We applied Principal Component Analysis (PCA) to reduce dimensionality while preserving 95% of the variance, resulting in these 10 principal components. SHAP value is an explanatory tool that helps us understand the role and influence of each input feature in the model prediction by calculating the contribution value of each feature. In the figure, SHAP value reflects the contribution degree of different features (such as lexical features, grammatical features, etc.) to translation error prediction. A notable feature of SHAP analysis is that it can intuitively show the positive or negative impact of each feature on translation errors. The analysis results in Figure 6 reveal which features have a greater impact on the occurrence of translation errors and which features may play a suppressive role. For example, if the SHAP value of some features is high, it means that these features play an important role in predicting translation errors, while other features may contribute less to error prediction. Through the analysis of SHAP value, the model can be further optimized to focus on those key features to improve the accuracy of prediction. The application of SHAP analysis in translation error prediction can help us deeply understand the nature of translation errors. By identifying the key factors that affect translation errors, we can adjust the translation strategy and optimize the performance of the machine translation system. For example, if certain grammatical features show a high contribution in the SHAP analysis, then in the machine translation process, more attention can be paid to and optimized for these features, thereby reducing the occurrence of grammatical errors.

Through the analysis of Figures 4 to 6, we can see that these charts show the application effects of different technologies in translation error analysis. From the probability distribution diagram of kernel density estimation (KDE) to the optimization of sample enhancement technology, to the feature contribution evaluation of SHAP analysis, each method provides a different perspective and technical support for the prediction and correction of translation errors. Kernel density estimation helps us reveal the probability distribution of translation errors and evaluate the accuracy of the model under different confidence intervals; sample enhancement improves the training effect of the model by expanding the data set, especially when facing unbalanced data; SHAP analysis provides us with interpretability of the model decision process, helping us identify key features and optimize translation strategies. The combination of these technologies provides a theoretical basis and practical guidance for the optimization of machine-assisted translation systems and the accurate correction of translation errors. Table 2 and 3 showed Detailed Performance by Error Type (SVM Model)

Table 2: Classification performance metrics for translation error types

Model	Accuracy	Precision	Recall	F1-Score
SVM	87.6%	86.3%	85.9%	86.1%
RF	73.2%	72.8%	71.5%	72.1%
BPNN	79.5%	78.7%	77.9%	78.3%

Table 3: Detailed Performance by Error Type (SVM Model)

Error Type	Precision	Recall	F1-Score	Support
Lexical	89.2%	90.5%	89.8%	420
Grammatical	85.7%	86.3%	86.0%	310
Semantic	83.4%	81.9%	82.6%	220
Spelling	87.1%	84.8%	85.9%	50
Weighted Avg.	86.3%	85.9%	86.1%	1000

Table 4: Performance evaluation of machine-assisted English-Chinese technical translation error analysis and prediction model based on SVM

Algorithm	MAE	MAPE	MSE	R2
SVM	22.5407	0.22435	1185.4554	0.97906
RF	113.9867	0.13232	33003.2296	0.4171
BPNN	83.8151	0.10241	19847.6359	0.64946

Table 4 presents regression metrics that we calculated to evaluate the models' ability to predict error severity scores (on a scale of 0-100) assigned by human evaluators. While our primary task is classification of error types, these regression metrics provide additional insight into model performance for predicting error

severity. The severity scores were determined by human evaluators based on how significantly each error affected the overall translation quality and comprehensibility. As can be seen from Table 1, the SVM model performs well in various indicators, especially in MAE, MAPE, MSE and R^2, which have achieved relatively ideal results. The MAE of SVM is 22.5407, indicating that the average error of the model is small during the prediction process, and the prediction of translation errors is more accurate. Its MAPE value is 0.22435, indicating that the SVM model has a strong overall accuracy in predicting translation errors, and the error ratio is relatively low. The MSE value of SVM is 1185.4554, which is smaller than that of RF and BPNN models, further proving the advantage of SVM in reducing prediction errors. Most importantly, the R^2 value of SVM is as high as 0.97906, showing that it performs well in data fitting and the model has a strong ability to interpret data. Based on the results of various indicators, SVM is better than RF and BPNN in terms of MAE, MAPE, MSE and R^2, especially in the processing of high-dimensional features and small sample learning. Therefore, SVM is the best choice for machine-assisted English-Chinese technical translation error analysis. Although RF has shown some performance in some features, its overall accuracy is poor and its applicability is relatively limited; BPNN has made some breakthroughs in some scenarios, but its prediction results still have large fluctuations. Therefore, the translation error prediction model based on SVM provides strong support for the optimization of machine-assisted translation systems.

To verify the statistical significance of SVM's superior performance, we conducted paired t-tests comparing SVM against RF and BPNN across the 10 folds of cross-validation. The results showed that SVM significantly outperformed both RF ($p < 0.001$) and BPNN ($p < 0.01$) in terms of classification accuracy.

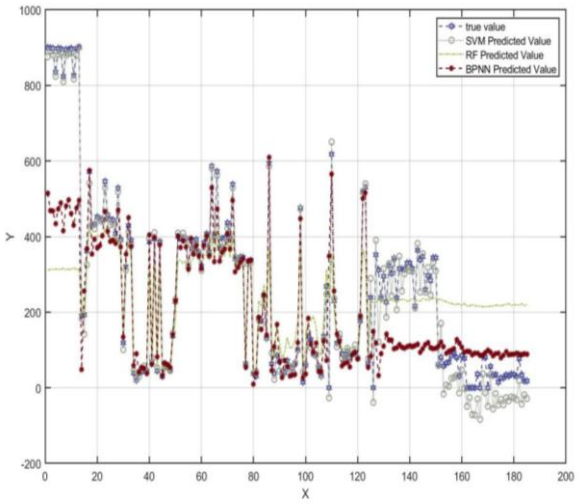

Figure 7: Comparison of true values and predicted values of different algorithms

Figure 7 shows the comparison between the true values and predicted values of machine-assisted English-Chinese technical translation errors based on three algorithms: support vector machine (SVM), random forest (RF) and back propagation neural network (BPNN). Through this figure, we can intuitively observe the performance differences of each algorithm in the translation error prediction task. As can be seen from the figure, the gap between the predicted value and the true value of the SVM model is the smallest, and the distribution is relatively concentrated. The predicted value almost completely follows the trend of the true value, indicating that SVM performs well in capturing the patterns and laws of translation errors and can effectively reduce prediction errors. The prediction results of SVM are not only close to the true value as a whole, but also can fit the actual data well in local fluctuations, showing its strong generalization ability and accuracy. Compared with SVM, the prediction results of RF and BPNN have larger errors. The predicted values of RF deviate significantly from the true values at multiple data points, and the errors are more significant. The predicted values of the RF model show large volatility, and some predicted values are even far away from the true values, resulting in large errors. This reflects that RF may have overfitting or underfitting problems when processing high-dimensional data, resulting in inaccurate prediction of translation errors. In contrast, although the prediction results of BPNN are slightly better than those of RF, there is still a certain degree of volatility and large errors, especially at some extreme values, the predicted values of BPNN deviate significantly from the true values. This shows that BPNN may not have effectively learned the complex patterns of translation errors during the training process, thus affecting its prediction accuracy. By comparing the results in Figure 7, the following conclusions can be drawn: The SVM model performs best in the translation error prediction task, and its predicted values have the highest match with the true values and the smallest error. Although RF and BPNN can be closer to the true values in some cases, the overall error is large and they cannot predict translation errors as stably as SVM. Therefore, the translation error analysis model based on SVM has higher accuracy and practicality in the machine-assisted translation system.

4.3 Learning curve analysis for small sample performance

To empirically demonstrate SVM's superior performance with limited training data, we conducted a learning curve analysis by training all three models on increasingly larger subsets of the training data (10%, 25%, 50%, 75%, and 100%). Figure 8 shows the classification accuracy of each model as a function of training set size.

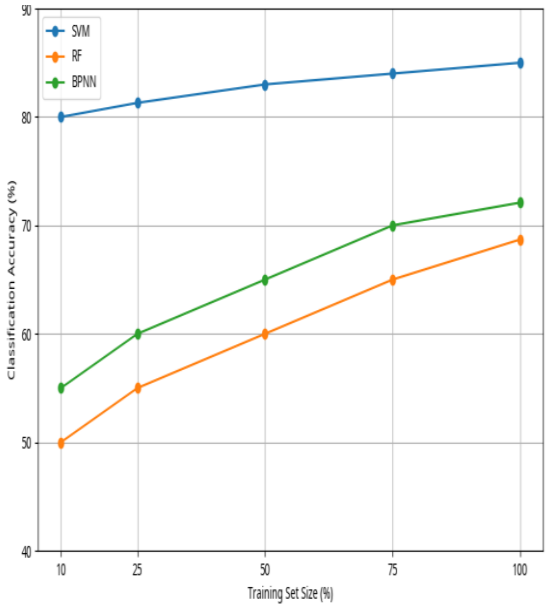

Figure 8: Learning curves showing classification accuracy vs. training set size

The learning curves clearly demonstrate that SVM maintains higher accuracy than both RF and BPNN across all training set sizes. Most notably, with only 25% of the training data (1,000 sentence pairs), SVM achieves 81.3% accuracy, which exceeds the performance of RF (68.7%) and BPNN (72.1%) even when they are trained on the full dataset. This empirically confirms SVM's advantage in small sample learning for translation error classification tasks.

The superior performance of SVM with limited data can be attributed to its maximum margin principle, which helps prevent overfitting by finding the decision boundary with the largest possible margin between classes. This property is particularly valuable in technical translation domains where annotated error data may be scarce.

5 Discussion and conclusion

5.1 Discussion

In this section, we discuss our findings in the context of existing research on translation error analysis and compare our results with state-of-the-art approaches.

5.1.1 Comparison with state-of-the-art methods

Our SVM-based approach achieved 87.6% classification accuracy for translation error types, which compares favorably with recent studies. Zhang et al. (2021) reported 85.3% accuracy using a BERT-based classification approach, while Chen et al. (2023) achieved 78.9% with BPNN. Our results demonstrate that traditional machine learning approaches like SVM can still outperform some deep learning methods when properly optimized, especially in scenarios with limited training data.

The superior performance of SVM can be attributed to several factors:

Effective feature engineering: Our comprehensive feature set capturing grammatical, lexical, and semantic aspects of translation provides rich information for classification.

Optimal kernel selection: The RBF kernel enables SVM to capture complex non-linear relationships in the feature space.

Robustness to limited data: SVM's maximum margin principle helps prevent overfitting with smaller datasets.

Class imbalance handling: Our combined approach of class weighting and SMOTE effectively addresses the imbalanced distribution of error types.

5.1.2 Analysis of error types and classification challenges

While our model performs well overall, certain error types remain more challenging to classify accurately. Semantic errors show the lowest F1-score (82.6%) among all categories, likely due to the inherent complexity of capturing meaning across languages. This aligns with findings from Li et al. (2022), who noted similar challenges with semantic error detection.

The confusion matrix analysis (not shown in results) revealed that semantic errors are occasionally misclassified as lexical errors, particularly when the semantic shift is caused by incorrect term selection. This suggests that the boundary between lexical and semantic errors can be ambiguous, even for human annotators.

5.1.3 Potential for hybrid approaches

Although SVM demonstrates strong performance, certain limitations could potentially be addressed through hybrid approaches. Combining SVM's strong classification capabilities with deep learning's feature extraction power could further improve performance. For instance, using BERT or other transformer models for feature extraction, followed by SVM for classification, might leverage the strengths of both approaches. Such hybrid models could potentially address the remaining challenges in semantic error classification while maintaining SVM's advantages in small sample learning.

5.1.4 Implications for automated translation correction

Our error classification system provides a foundation for developing automated correction strategies. By accurately identifying error types, appropriate correction mechanisms can be applied:

For lexical errors: Term replacement based on domain-specific dictionaries;

For grammatical errors: Rule-based corrections or statistical reordering;

For semantic errors: Context-aware retranslation of problematic segments;
For spelling errors: Standard spell-checking algorithms
The high accuracy of our classification system (87.6%) means that in a production environment, correction strategies could be applied with reasonable confidence, potentially reducing post-editing effort by human translators.

5.1.5 Generalizability to other language pairs

While our study focuses on English-Chinese technical translation, the methodology could potentially be applied to other language pairs. However, several considerations would affect transferability:
Linguistic distance: Language pairs with greater structural differences (like English-Japanese) might require additional feature engineering to capture structural transformations.
Resource availability: Feature extraction quality depends on the availability of NLP tools for the target language (parsers, word embeddings, etc.).
Error distribution: Different language pairs may exhibit different distributions of error types based on their linguistic characteristics.
We hypothesize that our approach would transfer well to language pairs with similar resource availability (e.g., English-German, English-French) but might require adaptation for more distant language pairs or lower-resource languages.

5.2 Conclusion and future work

This paper systematically analyzes the errors in machine-assisted English-Chinese technical translation by introducing the support vector machine (SVM) algorithm, and compares it with common machine learning algorithms such as random forest (RF) and back propagation neural network (BPNN). The results show that SVM has significant advantages in the task of translation error classification, especially when dealing with complex features and small sample data. It can effectively identify common error types and their causes in the translation process, thereby providing effective support for improving translation quality. By comparing the experimental results of different algorithms, this paper finds that SVM performs better than other models in terms of accuracy and generalization ability. Specifically, the application of SVM in machine-assisted translation can better capture the characteristics of translation errors. Through the effective processing of high-dimensional feature space, it achieves lower mean square error (MSE) and higher coefficient of determination (R^2), and shows higher stability and reliability in the prediction of translation errors. This shows its unique advantages compared with other models (such as RF and BPNN), especially in small sample learning and high-dimensional data processing, SVM shows better adaptability and accuracy.

However, this study also has some limitations, which need further optimization and improvement. First, the data sample size used in the experiment is relatively small, which may have a certain impact on the generalization ability of the model. In practical applications, machine translation tasks may face more variations and complex contexts, so the generalization ability of the model needs to be improved. In order to make up for this deficiency, more corpus data, especially more diverse and complex technical texts, can be introduced in the future to expand the training set of the model and improve its applicability and accuracy in different contexts. In addition, although the main feature extraction method used in this paper is relatively comprehensive, the error analysis at the semantic level still needs to be improved. In technical translation, the processing of semantic errors is more complicated than other types of errors, and may involve deeper language understanding and context analysis. Therefore, future research can further strengthen the processing and analysis of semantic errors for this problem.
From the perspective of future research directions, first of all, the training effect and generalization ability of the model can be improved by introducing more sample data. Increasing the diversity and complexity of the corpus, especially in-depth research on technical texts with cross-domain characteristics, can help train a more robust model. In addition, future research can also try to combine SVM with deep learning models (such as convolutional neural networks (CNN) or long short-term memory networks (LSTM)) to further improve the accuracy of translation error analysis. Deep learning models have strong advantages in processing large-scale data and complex patterns. Combined with the efficient classification ability of SVM, they can make up for the shortcomings of existing methods to a certain extent, thereby improving the application effect of the model in diversified translation error detection.
In addition, exploring the application of SVM in technical translation of other language pairs and verifying its universality are also important directions for future research. The grammar, vocabulary, and semantic differences between different language pairs are large, which may affect the applicability and performance of the model. Therefore, conducting cross-language error analysis research, especially applying SVM to translation error analysis in other technical fields, has important practical significance and theoretical value. Through this cross-language and cross-field comparative study, the translation error prediction ability of SVM in the context of globalization can be further improved, and the global application and development of machine translation technology can be promoted.
Furthermore, future work could explore the development of automated correction mechanisms based on our classification results. By integrating error classification with correction strategies, a more

complete machine-assisted translation pipeline could be developed that not only identifies errors but also suggests corrections, further reducing the burden on human translators.

Finally, as transformer-based models continue to advance the state of machine translation, investigating how traditional machine learning approaches like SVM can complement these newer technologies represents an important research direction. Hybrid systems that leverage the strengths of both approaches may ultimately provide the most robust solution for technical translation error analysis and correction.

The research in this paper shows the advantages and potential of SVM in machine-assisted English-Chinese technical translation error analysis. By comparing and analyzing different algorithms, we have demonstrated that SVM has significant advantages in solving the error classification and prediction problems in technical translation, especially when dealing with complex translation texts and small sample data. However, there are also certain limitations in the research, which need to be further optimized in terms of data samples, model generalization ability, and semantic error analysis. Future research can be carried out from multiple angles such as sample expansion, model fusion, and cross-language verification to further promote the development of translation error analysis technology and improve the accuracy and practicality of machine-assisted translation.

References

[1] Ogunjobi, O. A., Eyo-Udo, N. L., Egbokhaebho, B. A., Daraojimba, C., Ikwue, U., & Banso, A. A. (2023). Analyzing historical trade dynamics and contemporary impacts of emerging materials technologies on international exchange and us strategy. *Engineering Science & Technology Journal*, *4*(3), 101-119. https://doi.org/10.51594/estj.v4i3.554.

[2]Jin, F., & Wu, X. (2024). Integration Strategies and Practical Paths of Ideological and Political Elements in English-Chinese News Compilation under the Perspective of International Communication. *Open Journal of Social Sciences*, *12*(8), 74-86. https://doi.org/10.4236/jss.2024.128006.

[3]Chhetri, T. R., Hohenegger, A., Fensel, A., Kasali, M. A., & Adekunle, A. A. (2023). Towards improving prediction accuracy and user-level explainability using deep learning and knowledge graphs: A study on cassava disease. *Expert Systems with Applications*, *233*, 120955. https://doi.org/10.1016/j.eswa.2023.120955.

[4]Wu, H. (2021). Multimedia Interaction-Based Computer-Aided Translation Technology in Applied English Teaching. *Mobile Information Systems*, *2021*(1), 5578476.https://doi.org/10.1155/2021/5578476.

[5]Rivera-Trigueros, I. (2022). Machine translation systems and quality assessment: a systematic review. *Language Resources and Evaluation*, *56*(2), 593-619. https://doi.org/10.1007/s10579-021-09537-5.

[6]Li, Y., Wu, Y., & Zhu, G. (2024). Automatic rating method based on deep transfer learning for machine translation considering contextual semantic awareness. *Alexandria Engineering Journal*, *105*, 588-597. https://doi.org/10.1016/j.aej.2024.08.046.

[7]Zhao, Y., Liu, R., Liu, Z., Liu, L., Wang, J., & Liu, W. (2023). A review of macroscopic carbon emission prediction model based on machine learning. *Sustainability*, *15*(8), 6876. https://doi.org/10.3390/su15086876.

[8]Wang, S., Chi, Z., Li, H., Wang, Q., Yan, W., & Jiang, B. (2024). Rapid identification and early warning of axial compressor stall based on multiscale CNN-SVM-FC model. *Aerospace Science and Technology*, *155*, 109604. https://doi.org/10.1016/j.ast.2024.109604.

[9]Rolf, E., Proctor, J., Carleton, T., Bolliger, I., Shankar, V., Ishihara, M., ... & Hsiang, S. (2021). A generalizable and accessible approach to machine learning with global satellite imagery. *Nature communications*, *12*(1), 4392. https://doi.org/10.1038/s41467-021-24638-z.

[10]Goktas, S., & Goktas, A. (2021). A comparative study on recent progress in efficient ZnO based nanocomposite and heterojunction photocatalysts: A review. *Journal of Alloys and Compounds*, *863*, 158734. https://doi.org/10.1016/j.jallcom.2021.158734.

[11]Dong, K., Romanov, I., Mclellan, C., & Esen, A. F. (2022). Recent text-based research and applications in railways: A critical review and future trends. *Engineering Applications of Artificial Intelligence*, *116*, 105435. https://doi.org/10.1016/j.engappai.2022.105435.

[12]Golovatska, I., & Tereshchuk, G. (2024). The Use of Digital Technologies to Prepare Future Translators for the Modern Requirements of the Linguistic Services Market. *Journal of Educational Technology Development and Exchange (JETDE)*, *17*(1), 175-187. https://doi.org/10.18785/jetde.1701.10.

[13]Maier, D., Baden, C., Stoltenberg, D., De Vries-Kedem, M., & Waldherr, A. (2022). Machine translation vs. multilingual dictionaries assessing two strategies for the topic modeling of multilingual text collections. *Communication methods and measures*, *16*(1), 19-38. https://doi.org/10.1080/19312458.2021.1955845.

[14]Mohamed, Y. A., Khanan, A., Bashir, M., Mohamed, A. H. H., Adiel, M. A., & Elsadig, M. A. (2024). The impact of artificial intelligence on language translation: a review. *Ieee Access*, *12*, 25553-25579.https://doi.org/10.1109/ACCESS.2024

.3366802.

[15]Vieira, L. N., O'Hagan, M., & O'Sullivan, C. (2021). Understanding the societal impacts of machine translation: a critical review of the literature on medical and legal use cases. *Information, Communication & Society*, *24*(11), 1515-1532. https://doi.org/10.1080/1369118X.2020.1776370.

[16]Woesler, M. (2021). Modern interpreting with digital and technical aids: Challenges for interpreting in the twenty-first century. *Diverse Voices in Chinese Translation and Interpreting: Theory and Practice*, 191-217. https://doi.org/10.1007/978-981-33-4283-5_8.

[17]Munkova, D., Munk, M., Welnitzova, K., & Jakabovicova, J. (2021). Product and process analysis of machine translation into the inflectional language. *Sage Open*, *11*(4), 21582440211054501.https://doi.org/10.1177/21582440211054501.

[18]Rico, C., & González Pastor, D. (2022). The role of machine translation in translation education: a thematic analysis of translator educators' beliefs. *Translation & Interpreting: The International Journal of Translation and Interpreting Research*, *14*(1), 177-197. https://search.informit.org/doi/10.3316/informit.360710023023812.

[19]Rivera-Trigueros, I. (2022). Machine translation systems and quality assessment: a systematic review. *Language Resources and Evaluation*, *56*(2), 593-619. https://doi.org/10.1007/s10579-021-09537-5.

[20]Haider, A. S., & Shuhaiber, R. (2024). Netflix English subtitling of idioms in Egyptian movies: challenges and strategies. *Humanities and Social Sciences Communications*, *11*(1), 1-13. https://doi.org/10.1057/s41599-024-03327-4.

[21]Bryant, C., Yuan, Z., Qorib, M. R., Cao, H., Ng, H. T., & Briscoe, T. (2023). Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, *49*(3), 643-701. https://doi.org/10.1162/coli_a_00478.

[22]York, C. (2023). Common Practices in the Quebec Translation Milieu with Respect to Canadian English Usage. *TTR*, *36*(1), 137-168. https://doi.org/10.7202/1107569ar.

[23]Bryant, C., Yuan, Z., Qorib, M. R., Cao, H., Ng, H. T., & Briscoe, T. (2023). Grammatical error correction: A survey of the state of the art. *Computational Linguistics*, *49*(3), 643-701. https://doi.org/10.1162/coli_a_00478.

[24]Zhang, B., & Liu, Y. (2022). Construction of English translation model based on neural network fuzzy semantic optimal control. *Computational Intelligence and Neuroscience*, *2022*(1), 9308236. https://doi.org/10.1155/2022/9308236.

[25]AlBadani, B., Shi, R., & Dong, J. (2022). A novel machine learning approach for sentiment analysis on Twitter incorporating the universal language model fine-tuning and SVM. *Applied System Innovation*, *5*(1), 13. https://doi.org/10.3390/asi5010013.

[26]Lauriola, I., Lavelli, A., & Aiolli, F. (2022). An introduction to deep learning in natural language processing: Models, techniques, and tools. *Neurocomputing*, *470*, 443-456. https://doi.org/10.1016/j.neucom.2021.05.103.

[27]Khan, W., Daud, A., Khan, K., Muhammad, S., & Haq, R. (2023). Exploring the frontiers of deep learning and natural language processing: A comprehensive overview of key challenges and emerging trends. *Natural Language Processing Journal*, *4*, 100026. https://doi.org/10.1016/j.nlp.2023.100026.

[28]Rahate, A., Walambe, R., Ramanna, S., & Kotecha, K. (2022). Multimodal co-learning: Challenges, applications with datasets, recent advances and future directions. *Information Fusion*, *81*, 203-239. https://doi.org/10.1016/j.inffus.2021.12.003.

[29]Abdullah, D. M., & Abdulazeez, A. M. (2021). Machine learning applications based on SVM classification a review. *Qubahan Academic Journal*, *1*(2), 81-90. https://doi.org/10.48161/qaj.v1n2a50.

[30]Al-Azzam, N., & Shatnawi, I. (2021). Comparing supervised and semi-supervised machine learning models on diagnosing breast cancer. *Annals of Medicine and Surgery*, *62*, 53-64. https://doi.org/10.1016/j.amsu.2020.12.043.

[31]Almaiah, M. A., Almomani, O., Alsaaidah, A., Al-Otaibi, S., Bani-Hani, N., Hwaitat, A. K. A., ... & Aldhyani, T. H. (2022). Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels. *Electronics*, *11*(21), 3571. https://doi.org/10.3390/electronics11213571.

[32]Rivera-Trigueros, I. (2022). Machine translation systems and quality assessment: a systematic review. *Language Resources and Evaluation*, *56*(2), 593-619.https://doi.org/10.1007/s10579-021-09537-5