Enhancing Machine Translation of English Complex Sentences Using Refined Gradient CNN on Large-Scale Corpora

SiYuan Li
Department of Business and Foreign Languages, Hunan International Business Vocational College, Changsha, Hunan, 410200, China
E-mail: lisiyuan198084@gmail.com

Keywords: optimization, machine translation algorithm, english complex long sentence, refined gradient - convolutional neural network

Received: June 30, 2025

Optimization of long and complicated sentences in English. Translating complex, lengthy statements from one language to another is the job of computer systems called machine translation algorithms (MTAs). A machine translation assistant (MTA) that trains on a big data corpus is one that makes use of a diverse and extensive collection of textual resources to improve translation quality. Translating complex and lengthy English sentences poses significant challenges for machine translation (MT) systems, especially when preserving semantic accuracy. It introduces the Refined Gradient-CNN (RG-CNN) model as a post-processing refinement mechanism to enhance phrase-level translation accuracy. The model is trained on a specially curated "Parallel Corpus" dataset comprising 1,563 English sentence pairs, including complex originals and their simplified counterparts. The RG-CNN employs gradient-enhanced convolution and bidirectional recurrent layers to capture and refine syntactic structures. The model is implemented using Python 3.11. Experimental results demonstrate the model's superior performance. It achieved BLEU scores of 73.1% (corpus) and 70.1% (local), significantly outperforming. Likewise, RG-CNN reported a reduced WER of 0.3% (corpus) and 0.10% (local) compared to baseline models. Accuracy and recall were also improved to 97.51% and 98.43%, respectively, outperforming the baseline model. These results affirm RG-CNN's ability to optimize complex sentence translation, reduce ambiguities, and advance MT systems across diverse linguistic domains

Povzetek: Model Refined Gradient-CNN (RG-CNN) je predlagan za izboljšanje strojnega prevajanja dolgih in kompleksnih angleških stavkov, zlasti za fraze. Model je treniran na obsežnem korpusu (1.563 parov) in optimira prevode kompleksnih besednih struktur.

1 Introduction

The English complex long sentence machine translation method is designed to be more efficient and accurate with a lot of important parameters. The method ought to be context-aware and aware of the original sentence's meaning so that it can generate translations that are very faithful to the original text. It should be capable of dealing with idiomatic expressions, cultural references, and metaphors to generate translations that are faithful to the original text. In order to effectively address the computational requirements for analyzing compound, lengthy words, the approach can take advantage of parallel processing methods and distributed computation models to offer optimal efficiency [1]. The research ensures that even sentences that are long and complex get translated outputs within a reasonable timeframe. Optimization and analysis of the translated output can be facilitated by incorporating a human translator or linguist feedback loop. The computer can learn from human experience through repeated processes and thus can translate difficult, lengthy texts with ease and efficiency [2]. With large language models, machine learning, high-performance NLP techniques, context and semantics considered, efficiency boost, and a human subject-matter expert feedback loop, all are included in the improvement of the English complex long sentence machine translation. All these in consideration, the algorithm's translation capability for complex, long texts may be significantly enhanced [3]. The English complex sentence machine translation technique from a large corpus of data needs to be trained to improve the quality, accuracy, and fluency of the translation. The following are a few key considerations: Choose a large, broad-based, and comprehensive corpus that includes a range of topics, genres, and styles [4]. Align the source and target sentences in the corpus to produce aligned sentence pairs for generating the translation. It can learn English phrase-to-phrase mappings and their respective translations through a critical phase while training a supervised machine translation system. Apply parallel

processing techniques and distributed computing platforms for efficient management of the enormous processing involved in training over an enormous data corpus. It is easy to scale up and accelerate the training [5]. Employ the most advanced neural network architectures, for instance, transformer architectures, which have made significant jumps in machine translation tasks. The models are better at handling complex sentence structure and long-distance relations. For pre-training the machine translation model, employ the pre-trained language models like BERT or GPT. Fine-tune the model on the huge data corpus [6], particularly for the translation of very hard, long sentences. Transfer learning helps the model learn to identify universal language use patterns, and fine-tuning helps it learn to follow the specific translation task that is being performed. Generate artificial sentence variations or paraphrases to improve the training set. The algorithm becomes stronger and more immune to complex, long sentences by being exposed to more varied sentence forms and language variations. Employ the automated translation system and collect user ratings of translations. Employ the feedback to improve the quality of the translations over time, using repeated usage in the training process. The research enables the algorithm to capture user preferences and personal translation challenges from complex, lengthy sentences [7]. Use standard measuring devices at regular intervals to test how well the improved algorithm is performing. Compare the algorithm with other cutting-edge machine translation systems to evaluate its performance and what it needs to improve on. The English compound long sentence MTA can be translated better, more fluently, and contextually using the power of big data corpora and implementing the optimization techniques. It generates more accurate, efficient, and effective translations of long compound sentences [8]. Large and diverse quantities of text data referred to as a "big data corpus" undergo processing in machine translation and other NLP tasks, training, testing, and also updating the processes. "Big data" thus addresses the sheer volume, variety, and velocity of data that can be processed and analyzed [9]. The size and variation of the corpus are also significant factors that decide the level at which it can be successfully used for training. With a large corpus, there can be complete utilization of different grammatical patterns, lexis, idiomatic expressions, and language variation. The variation of the corpus ensures that the algorithm is trained in multiple domains and styles, thereby ensuring that it is capable of handling mixed styles and subject matters. Data collection and cleaning must be performed cautiously when creating a large corpus of data. It is ensured that text data covers a wide range of language and context variations by obtaining it from various sources [10]. Preprocessing methods, such as tokenization, phrase breaking, and part-of-speech tagging, are employed in attempting to process data to train and analyze. The various techniques are able to train the machine translation models, such as statistical machine translation (SMT) and neural machine translation (NMT), upon the creation of the big data corpus. The models can learn more complex phrase structures, language patterns, and correspondences because they have larger training data. Big data corpora have helped machine translation advance [11]. Big data analysis enables researchers and developers to train and better develop models, which enhance translation quality, manage complex sentence structures better, are natural-sounding, and possess greater awareness of context. MTAS should work much better when there is a massive amount of data that is high in quality. To ensure the corpus is representative, precise, and unbiased, or noise-free, and does not adversely affect the translation outcome, meticulous data selection, preprocessing, and curation are needed. Lastly, a large quantity of data as a corpus allows machine translation systems to learn from heterogeneous linguistic data, leading to more efficient and robust translation abilities for practically any level of sentence complexity and linguistic diversity [12].

Key contributions:

- The application of a certain machine translation algorithm to process lengthy, complicated words, designing the Refined Gradient-CNN model, applying a huge training set, and optimization methods used to enhance word translation accuracy.
- This Research aims to overcome the difficulties in translating complex phrase patterns and enhance the general effectiveness of machine translation systems.

2 Related work

Research in many areas must include a literature survey, commonly called a literature review or systematic review. It entails a thorough review and analysis of the research body, shown in Table 1.

Table 1: Literature survey

References	Objectives	Summary of Findings	Limitations
[13]	Research suggests that the segmentation of lengthy phrases is made possible by the hierarchical network of ideas technique, which has been enhanced.	The study evaluated the features of professional literature and discussed a translation optimization approach for professional literature, combining statistics, which significantly increases.	Limited focus on structural segmentation; lacks handling of deep syntactic variations in English

[14]	Research improved to build researchers used the multi-objective optimization technique. The study also employs parallel corpora and monolingual corpora routes with an emphasis on node distribution and data flow analysis.	The study focused on the neural machine translation model's probabilistic structure, which allows researchers to draw conclusions about data-related regularization items and apply them.	Lacks real-time learning adaptation; limited performance on unseen sentence structures
[15]	The study improved the framework to optimize a computer-assisted translation system to increase the accuracy and reliability of automatic translation of long-character English with memory-assisted English.	It showed that the newly suggested computer-assisted translation system can improve translation quality and intelligently translate memory-assisted long-character English with high data recall rates, accuracy, and dependability.	Relies on memory-based context, which is insufficient for unseen phrase structures
[16]	The Research introduced by employing a word corpus, the word alignment optimization approach enhances word alignment performance in the transformer system.	The overview objective of the goal showed that, in comparison to the earlier methods, the suggested technique lowers the average alignment error rate.	Focused only on word alignment; doesn't address phrase-level semantic coherence.
[17]	The Research suggested that a model for calculating language-semantic correlation that uses the best fuzzy semantics for English lengthy sentences should be developed.	The study evaluated the process of fuzzy semantic selection achieved using a machine learning neural network adaptive learning technique.	Limited grammatical handling; doesn't scale well for professional or complex sentence contexts.
[18]	The study suggested language combinations and collected and cleaned texts from diverse sources to form four parallel corpora, which were used to build the translation system.	The Research focused on creating human and automated assessments of the resulting models.	Data-centric approach; lacks structural model improvements for long or technical texts
[19]	The overall objective of the goal was to explore the two different NMT algorithms, Bidirectional Long Short-Term Memory (LSTM) and Transformer-based NMT, used for the Bangla-to-English language pair.	The Research investigated to show the viable direction for Research to improve Bangla-English NMT.	Focused on a specific language pair; not generalizable to English complex phrase structure.
[20]	The study analyzed that well-known translator like Google Translate do quite well when translating between English, French, or Spanish. Still, studies make trivial mistakes when translating recently introduced languages like Bengali, Arabic, etc.	The Research examines English, which has been the base or source language for the vast majority of NLP research projects that have been discovered so far. The study had several regularly spoken potential languages that still need to be explored.	Not optimized for general MT performance; lacks contextual learning layers
[21]	The overall objective of the goal is to briefly present the voice recognition neural network technique. The machine translation method was then put through simulation studies and contrasted with two additional machine translation techniques.	The study showed that the backpropagation (BP) neural network recognized speech more quickly than artificial recognition and with a reduced word mistake rate.	Focused on speech input; not optimized for written complex language translation
[22]	To enhance Punjabi-to-English NMT translation by addressing out-of-vocabulary (OOV) words and multi-word expressions (MWEs).	Incorporating MWEs and word embeddings improved translation fluency and adequacy, achieving BLEU scores of 15.45, 43.32, and 34.5 on small, medium, and large test sets, respectively.	Limited to Punjabi-English pair; does not generalize to other low-resource or morphologically rich languages.
[23]	$BLEU_n$-based evaluation, residual comparison, Google Translate and European Commission's Translation tool (EC) tool	NMT showed higher translation quality than SMT across all $BLEU_n$ scores	Focus limited to English–Slovak; no deep feature extraction or semantic ranking; lacks domain-independent generalization

3 Methodology

3.1 Problem statement

To improve the quality of translation of lengthy and complicated English sentences, especially for low-resource language pairings, while existing methods like fuzzy semantic selection approaches [17] encourage phrase segmentation and semantic correlation, but cannot be particularly effective at learning deep contextual relationships. Similarly, structural alignment for long-sentence translation, especially minority languages [20], was a challenge for Transformer-based models [19] and Bidirectional LSTM models [19]. Furthermore, conventional approaches have significant word error rates and are ineffective at managing contextual memory [15]. The proposed Refined Gradient-CNN model overcomes all of the above limitations by incorporating contextual memory encoding and layered semantic mapping, which improves translation accuracy for complex phrase structures.

3.2 Experimental procedure

This section provided a thorough explanation of how the steps of the suggested design in Figure 1 were created, covered its creation process, and covered its key components. This analysis has four parts: Information gathering is generally the main goal of the first stage. The second part included MTA for long English sentences. The most significant information is found in the third part, which describes the work performed to develop the Refined Gradient-CNN model and compile the essential knowledge. The efficiency of each current and previous design is presented in the fourth section. It is judged by contrasting the pertinent factors.

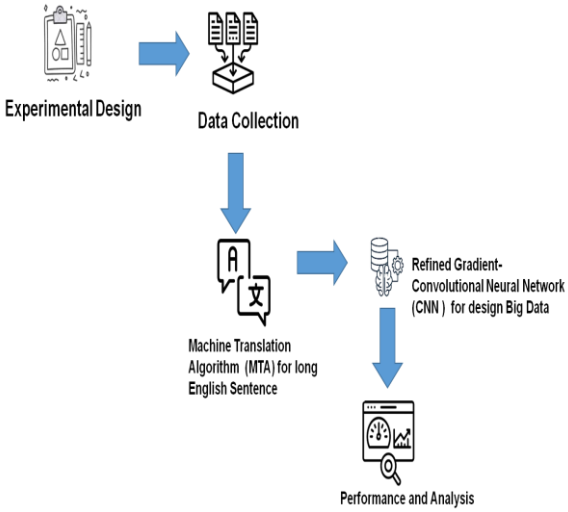

Figure 1: Methodological design

A. Data collection

To optimize sentences and enhance translation was validated it using the 1,563 English sentence pairings from the "Parallel Corpus" dataset. Each record includes meta-data such as readability ratings, difficulty levels, and domains, as well as the original complicated text and its simplified or optimized translation. The aim of decreasing language complexity, enhancing semantic retention in AI, education, and NLP applications, and enhancing phrase translation quality at the phrase level are all complemented by this dataset. For model development and evaluation, the dataset was partitioned into 70% for training (1,094 samples), 15% for validation (235 samples), and 15% for testing (234 samples).This structured split enables robust model performance assessment across varied difficulty levels and domain contexts, making it a reliable benchmarK due to its rich linguistic annotations and domain diversity. It can serve as a thorough benchmark for model performance testing on a range of difficulty and context aspects due to its extensive domain coverage and strong language annotation.

Source: https://www.kaggle.com/datasets/ziya07/parallel-corpus-data/data

B. MTA for english complex long sentence

Sentences that are distinguished by their length and complicated structure are referred to as being optimized for English complex, lengthy sentences. Many clauses, sub-clauses, phrases, modifiers, and dependent connections may be found in the sentences. Complex, lengthy phrases may be difficult to understand, comprehend, and translate because of their complex syntax and potential for ambiguity. The rearranging module intends to fit more closely the short phrases' translation with the language order following the combination by rearranging the short phrases generated through segmentation. Figure 2 depicts the upgraded intelligent MTA's flow.

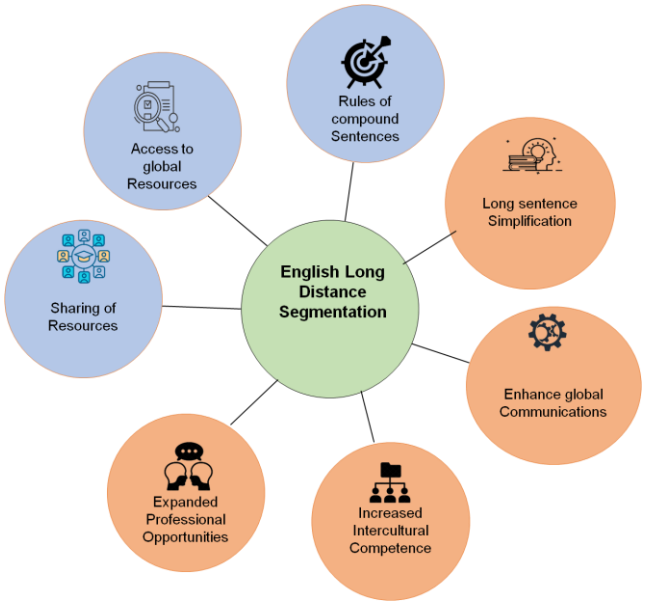

Figure 2: Translation process of MTA for long English sentences

The sentence segmentation module is designed to split lengthy English sentences into shorter, manageable segments. This is accomplished by predicting the likelihood of each word being a segmentation point using a maximum entropy (MaxEnt) classifier. The MaxEnt approach is particularly suitable here because it models conditional probabilities flexibly without assuming independence among features.

$$o\left(v\middle|u(z)\right) = \frac{\exp\left(\sum_j z_j h_j(v,u(z))\right)}{\left(\sum_v exp(\sum_j z_j h_j(v',u(z))\right)} \quad (1)$$

Where $o\left(v\middle|u(z)\right)$, the likelihood that a word will$z$ is a segmenting term in the lengthy statement, $u(z)$ is the background knowledge.

After segmentation, the reordering module rearranges the segmented short sentences to reflect the original logical flow. This is again modeled using a maximum entropy classifier, which estimates the likelihood of a correct sequence based on context and neighboring sentence features. Equation (2) is the appropriate computation and reads as follows:

$$o(p|D_n^t, D_s^n) = \frac{\exp\left(\sum_j z_j h_j(p, D_n^t, D_s^n)\right)}{\left(\sum_p exp\left(\sum_j z_j h_j(D_n^t, D_s^n)\right)\right)} \tag{2}$$

The encoder receives the short English sentences that have been segmented and re-ordered. The original text is encoded by the encoder using an LSTM model, and the resulting computation is given by Equation (3-7):

$$e_s = \sigma(a_e + X_{eu_s} + Z_{eg_{s-1}}) \tag{3}$$

$$t_s = e_s t_{s-1} + h_s \sigma(a + X_{u_s} + Z_{eg_{s-1}}) \tag{4}$$

$$h_s = \sigma(a_h + X_h u_s + Z_{eg_{s-1}}) \tag{5}$$

$$h_s = tang(X_h) r_s) \tag{6}$$

$$r_s = \sigma(a_r + X_r u_s + Z_r g_{s-1}) \tag{7}$$

C. English long-distance segmentation

English long-distance segmentation refers to breaking up a long sentence into sub-clauses or segments in order to better understand and analyze it. It is commonly applied in linguistics, machine translation, and natural language processing (NLP) for splitting complicated sentences and grasping the syntactic structure of the sentence. The dataset was chosen due to its relevance for tasks involving the reduction of linguistic complexity and the preservation of semantic meaning, particularly within applications related to artificial intelligence, education, and natural language processing (NLP). Its diverse domain representation and robust linguistic annotations make it a reliable benchmark for evaluating model performance across varying levels of sentence complexity and contextual nuance. Various clauses, words, and sub-clauses within a sentence are detected and separated from each other according to their grammatical relationship and dependencies via long-distance segmentation. The intention is to produce substantial sentences that can be learned separately or in conjunction with other, longer sentences. Find the sentence's main clauses or independent sections. The foundation elements convey complete ideas and may be utilized alone as independent sentences. Identify any subordinate or dependent sentences that provide the direct clauses with explanation, background, and information. The fines typically begin with relative pronouns such as "who," "which," or "that" and subordinating conjunctions such as "although," "because," or "if." The sentence needs to be dissected into modifiers and relevant phrases. These include noun phrases, verb phrases, adverbial phrases, and adjectival words. Chunks help identify interrelations and roles among sentence constituents. Establish relationships and interdependence between the various constituents. Establish the verb-object relationships, subject-verb concordances, and other syntactic relations that contribute to the general sentence form. Large sentences become richer in analysis and interpretation for linguists, NLP programmers, and machine translation algorithms when they are divided into smaller constituents. The proposed approach provides a better understanding of the syntactic form and semantic connections of the text and is more convenient for translation or further study with greater accuracy. Long-distance segmentation is particularly useful in complex languages such as English, which possess complicated sentence forms with multiple clauses and modifiers. The lengthy phrases can be broken down into smaller segments to lessen confusion and enhance the interpretation and comprehension of the entire message. By the decomposition of huge words into smaller pieces or clauses, long-distance segmentation is important in the study of language, NLP processing, and machine translation. It ensures that sentence structures can be examined more systematically and explicitly, allowing for free, correct comprehension, translation, and analysis of complex language phenomena.

D. Refined gradient-convolutional neural network /9RG- CNN) design for big data

There are several factors to consider and methods to use when developing huge amounts of data. To improve RG - -CNN design, particularly for large data situations, there are differences between a regular CNN and the proposed Refined Gradient-CNN (RG-CNN). To handle complicated phrase structures and huge datasets more effectively, the RG-CNN combines gradient-based refinement, batch normalization, dropout, ReLU variations, enhanced memory handling for big data, and sophisticated pooling methods. Take into consideration the following key strategies: Big data typically encompasses a very large number of input samples. The task is to develop an RG - CNN model that is scalable to attack it. This might involve employing parallel computing platforms, splitting the job between numerous computers, and enhancing memory management to deal with large datasets. Big data typically must split the training task between numerous computer nodes or clusters. The training process can be segmented based on techniques such as model parallelism and data parallelism, thereby providing faster convergence and efficient use of resources. Batch normalization is one of the techniques used to surmount the challenge of training RG-CNN and other deep neural networks with large sets of data. It scales and normalizes the activations of every layer of a network to help with faster and convergent training. Both the overall performance and the generalization of the RG-CNN model can be improved by

batch normalization. Overfitting of massive data must be avoided using regularization methods. The task can employ regularization and dropout in order to control the complexity of the model and induce more generalization. Regularization improves the ability of the model to deal with natural noise and variance present in large data. The performance of CNN on extremely large data might be significantly affected by choosing the appropriate activation functions. Rectified Linear Units (ReLU), which reduce the vanishing gradient problem and the training speed, have proven to be useful. In order to detect more intricate patterns, their variations, such as Leaky ReLU or Parametric ReLU, can be used. For handling enormous data, transfer learning can be used. Starting points include pre-trained CNN models on huge datasets like ImageNet. The huge data may be fine-tuned better to fit the CNN to the particular job using the information gained from these models. The problem of little labeled data may be solved by transfer learning, which will enhance RG-CNN performance. Table 2 displays the RG-CNN model hyperparameters.

Table 2: RG-CNN model hyperparameters and configurations for effective training and convergence.

Parameter	**Value / Range**	**Description**
Optimizer	Adam / AdamW	Adaptive learning for sparse data
Learning Rate	1e-4 to 5e-4	Tuned using learning rate scheduler (e.g., ReduceLROnPlateau)
Batch Size	64–128	Based on GPU memory
Epochs	10–30	Early stopping based on BLEU validation
Dropout Rate	0.3–0.5	To prevent overfitting
Activation Function	ReLU / LeakyReLU	For non-linearity in CNN layers
Max Sequence Length	100–150 tokens	For padding and positional encoding
Embedding Dimension	512 / 768 (or match transformer encoder)	Use with pre-trained embeddings
Kernel Size (CNN)	3 × 3 / 5 × 5	For capturing n-gram features
Pooling	MaxPooling	To retain the most relevant features
Gradient Clipping	1	Prevent exploding gradients
Scheduler	Warm-up + Cosine Annealing / ReduceLROnPlateau	Smoother convergence

3.3 Convolutional Neural Network

A CNN is made up of five parts: input data, a convolutional layer, a pooling layer, FC overlay, and an output vector. CNNs come in a variety of layer combinations. The CNN's structure, which was used in this experiment, is shown in Figure 3.

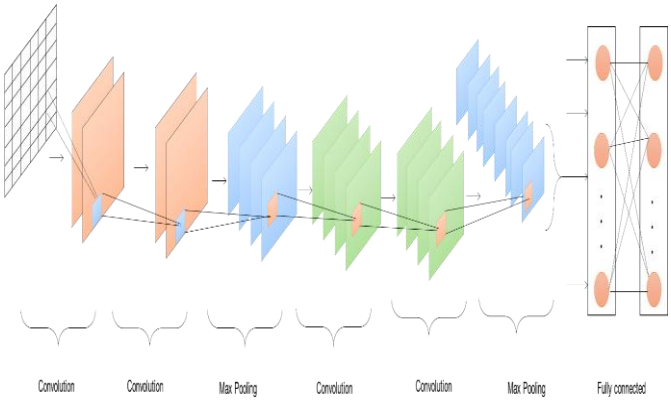

Figure 3: Structure of Convolutional Neural Network

Finding intriguing patterns in the data is the goal of the convolutional gradient task. Each of the many layers' convolutional kernels has a frequency and a divergence coefficient. It is assumed that u_j is the weight parameter, a_j is the divergence amount, and V_{j-1} is the input to convolution layers j while the inversion kernel j is active. One such expression for the convolution operation of Equation (8) is:

$$V_j = e\left(u_j \otimes V_{j-1} + a_j\right) \tag{8}$$

Where $V_j$the output result of the convolution kernel $j$represents the convolution operation, and $e(x)$ represents the activation function.

The input data is swept repeatedly by the convolutional network, which then extracts the distinctive information. In addition, the multilayer's operational amplifier is changed to$ReLU$. The Linear transfer function is simpler to derive than the exponential transfer function and other training algorithms, allowing for faster model training and better protection against gradient disappearing. It is possible to write $ReLU$, which is represented in Equation (9) as:

$$ReLU\left(V_j\right) = \begin{cases} V_j\left(V_j > 0\right) \\ 0\ \left(V_j > 0\right) \end{cases} \tag{9}$$

The pooling layer's main function is to reduce down-sampling data redundancy, which also aids in

achieving invariance and reducing CNN complexity. The two most popular ways to finish pooling are pooling layer and max pooling. If the study uses averaged pooling, the outcome is the computing area's arithmetic mean, but if the study uses max pooling, the outcome is the computation area's highest value. Max pooling was used for this investigation because it preserves important data better than average pooling. Equation (10) in mathematics provides the maximum pooling:

$$R_i = max(O_i^0, O_i^1, O_i^2, O_i^3, \ldots . O_i^s,) \tag{10}$$

Where R_i is the return outcome of the pooled region i, Max is the maximum pooling procedure, and O_i^0 is the pooling area i is the element s.

A CNN's "classifiers" are its layers. Its main objective is to reorganize the data that the convolutional and pooling layers retrieved and weighted from the hidden-layer space. A similar dropout method is implemented in the layer to randomly eliminate neurons to prevent over-fitting.

Let's determine the backpropagation updates for a network's convolutional layers. The output feature map is created by convolving the feature maps from the preceding layer using learnable kernels and then processing them via the activation function. Convolutions with numerous input maps may be combined in each output map. Equation (11) is often shown as,

$$U_i^k = f\left(\sum_{j \in N_i} U_i^k * K_{ji}^k + a_i^k\right) \tag{11}$$

3.4 Computing the gradients

A down-sampling layer's map weights are all set to the same valueβ, to determine the value of, we only scaled the result of the prior procedure by β. For each map, we may do the same δ. Calculation again iEquation (12–15) represents the pairing of the map from the layer of convolution and the associated map from the sub-samples layer:

$$\delta_i^k = \beta_i^{k+1}\left((e'(x_i^k) \circ up(\delta_i^{k+1})\right) \tag{12}$$

$$up(u) \equiv u \otimes 1_{m \times m} \tag{13}$$

$$\frac{\partial F}{\partial a_i} = \sum_{x,y}(\delta_i^k)_{xy} \tag{14}$$

$$\frac{\partial F}{\partial \iota_{ji}^k} = rot180(conv2(u_j^{k-1}, rot180\left(\delta_j^{k-1}, rot180\left(\delta_j^k\right), valid'\right)) \tag{15}$$

The input maps are produced as down-sampled copies using a sub-sampling layer. There will be exactly N export maps if there are N intake maps, albeit the final maps will be smaller. More formally, they are calculated as Equations (16),

$$u_i^k = e(\beta_i^k down\left(u_j^{k-1}\right) + a_j^k \tag{16}$$

To identify that the patch in the input map is related to a specific pixel in the output map, and to calculate the gradient of a kernel. Applying a delta recursion that resembles Equation (17-20) in this case requires determining which area in the sensitivity map of the present layer corresponds to a particular pixel in the sensitivity map of the following layer. The weights, being the weights of the (rotated) convolution kernel, are increased by the relationship between the input patch and the output pixel. Convolution is once again used to do this effectively:

$$\delta_j^k = e'(u_j^k) \circ conv(\delta_j^{k+1}, rot180\left(l_i^{k+1}\right), 'full') \tag{17}$$

$$\frac{\partial F}{\partial a_i} = \sum_{x,y}(\delta_j^k)_{xy} \tag{18}$$

$$c_i^k = down(u_i^{k-1}) \tag{19}$$

$$\frac{\partial F}{\partial a_i} = \sum_{x,y}(\delta_i^k \circ x_i^k)_{xy} \tag{20}$$

3.5 CNN algorithm

The network weights are updated by the CNN algorithm (1) by using a method known as backpropagation, depending on the error between the predicted and actual results. The CNN can learn and develop its capacity to identify patterns and objects in pictures due to this iterative process of forward propagation (feeding data through the network), backpropagation, and object detection. In various computer vision applications, such as picture classification, object recognition, and image segmentation, CNN methods have shown outstanding performance. We have been used to various tasks, including autonomous driving, picture analysis in medicine, and face identification.

Algorithm 1: Convolutional Neural Network

Function CNN (input_data);
// Convolutional layers
For each convolutional layer:
Convolution=apply convolution (input_data, weights)// Apply convo:

Activation=apply_activation (convolution) // Apply activation function
Pooling=apply_pooling (activation)// Apply pooling operation (e.g)
// fully connected layers
Flattened=flatten (pooling) //Flatten the pooled feature maps into a
For each fully connected layer:
Weights=initialize_weights () //Flatten the polled feature maps into a
Bias =initialize bias () // Initialize bias for the fully (
Linear_transform = apply_linear_transform (flattened, weights, bias)
Activation=apply_activation (linear_transform) //Apply activation
//Output layer
Out_weights -= initialize_output_weights () //Initialize weight for
Output_bias = initialize-output-bias () // Initialize bias for the output
Output= apply_linear_transform (activation, output_weights, output_bias)
Predicated_class = classify_output (output)// Classify the output to dete
Return predicted_class

4 Result and discussion

Results are always advised to reference the most recent literature survey for the most up-to-date information on these themes, since the exact outcomes and improvements in the translation of complicated, lengthy phrases might differ based on the research and development efforts in the area.

To improve the accuracy of translating complex English sentences in real time, the proposed RG-CNN model was built using Python 3.11. Table 3 demonstrates the simulation setup.

Table 3: Simulation setup

Component	Recommended Specification
GPU	NVIDIA A100 / RTX 3090 / Tesla V100
RAM	32–64 GB
Storage	SSD (1 TB recommended for large corpora like WMT)
Framework	PyTorch / TensorFlow
Distributed Training	Optional with Horovod / DDP (for WMT-scale corpora)

4.1 English translation design using big data

The results further emphasize the efficiency and rationale of the two training approaches, particularly the bidirectional training strategy. This strategy, when combined with TransE, demonstrates its effectiveness through enhanced entity prediction using a Multilayer Perceptron (MLP) layer. The MLP receives inputs from the TransE pre-trained embeddings and refines them, enhancing the expressive capability of the overall model. Table 4 and Figure 4 illustrate the outcomes. The proposed RG-CNN model, which incorporates both bidirectional training and the TransE+MLP architecture, achieves significantly higher precision values: P@1 = 84%, P@5 = 88%, and P@10 = 98%. These metrics indicate that the model reliably ranks the correct simplified sentence within the top predicted candidates. Furthermore, the use of a Recurrent Neural Network (RNN) within RG-CNN effectively captures bidirectional dependencies in the data, contributing to performance that surpasses even the TransE+MLP configuration. This validates the architectural choice and demonstrates the robustness of the proposed model.

Table 4: Numerical outcomes of the Training strategy based on the algorithm

Test sets	Percentage (%)		
	P@1	P@5	P@10
TeansE	25	30	35
TeansE+ MLP	40	45	50
FB15K	82	86	95
PP1	70	75	80
Refined Gradient - CNN [Proposed]	**84**	**88**	**98**

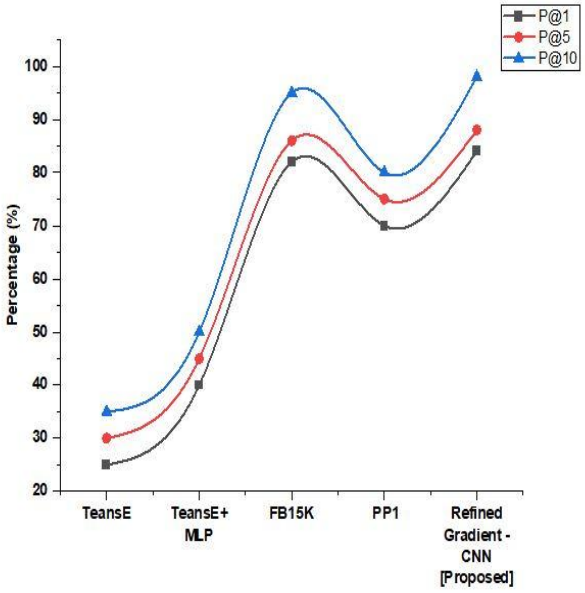

Figure 4: Comparison of Training strategy based on the algorithm

The suggested Refined Gradient-CNN model outperforms the Improved Long Short-Term Memory (LSTM) [24] and Hierarchical Network of Concepts (HNC) models [25]. It successfully enhances machine

translation of complex language patterns by capturing complex phrase structures.

The proposed Refined Gradient-CNN model outperformed the Improved LSTM [24], 31.3% and 3.9%, respectively, in terms of BLEU scores, as shown in Table 5 and Figure 5, 73.1% for the corpus dataset and 70.1% for the local dataset. The outcomes demonstrate how well the proposed model works to translate complex phrase structures with greater n-gram overlap. This aligns with the goal of the study, which is to improve machine translation systems by raising overall translation quality and semantic integrity across a variety of datasets, especially for complex language structures.

Table 5: Comparison of BLUE score on corpus and local dataset

Methods	**BLEU (%)**	
	Corpus dataset	Local dataset
Improved LSTM [24]	31.3	3.9
Refined Gradient-CNN [Proposed]	**73.1**	**70.1**

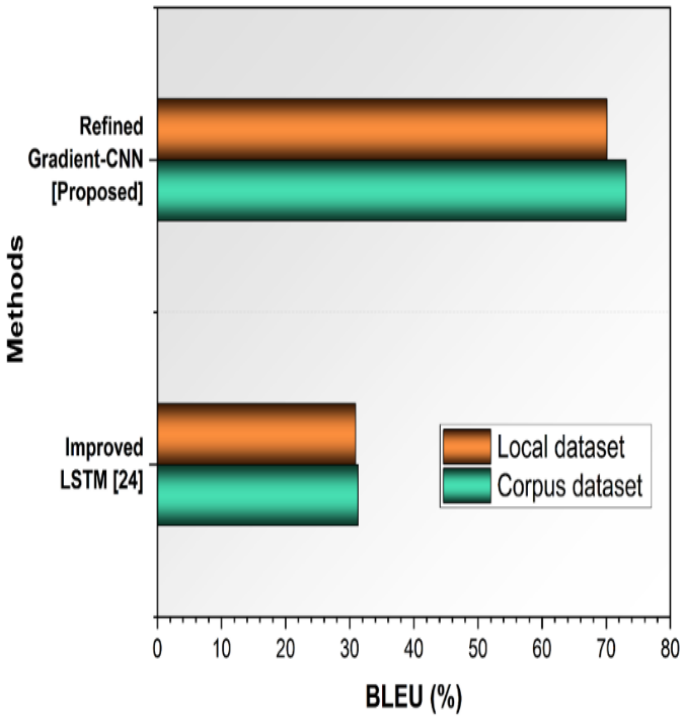

Figure 5: BLEU comparison of various models

The translation accuracy metric known as WER, which counts the number of insertions, deletions, and substitutions required to arrive at a reference translation, is shown in Table 6 and Figure 6. Higher translation precision is indicated by a lower WER. Compared to the Improved LSTM [24] 0.9% and 1.1%, the WERs recorded by the Refined Gradient-CNN were significantly lower 0.3% for the corpus and 0.10% for the local data. These findings align with the study goal of improving overall machine translation quality by showcasing the model's enhanced capacity to manage complex phrase structure.

Table 6: Comparative word error rate (%) analysis for english phrase translation accuracy

Methods	**Word Error Rate (%)**	
	Corpus dataset	Corpus dataset
Improved LSTM [24]	0.9	1.1
Refined Gradient-CNN [Proposed]	**0.3**	**0.10**

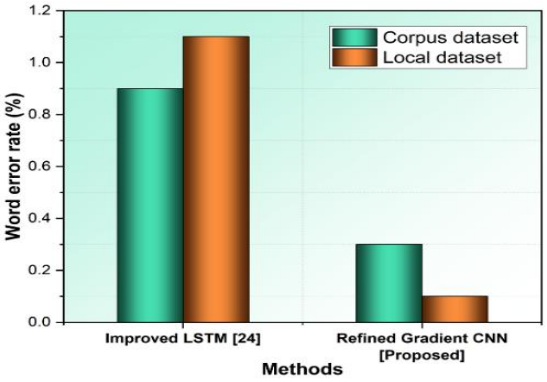

Figure 6: Word Error Rate (%) comparison of various models

Improving machine translation accuracy, the proposed Refined Gradient-CNN model significantly enhances the translation of the challenging English phrase structures. With 97.51% accuracy and 98.43% recall, the Refined Gradient-CNN outperformed the HNC technique [25], which achieved 93.38% accuracy and 94.51% recall, as shown in Table 7 and Figure 7. These improvements demonstrate the model's improved capacity to recognize and translate intricate phrase patterns more accurately. The results demonstrate that Refined Gradient-CNN improves machine translation systems' overall performance and efficiency in authentic language situations while also reducing translation ambiguities.

Table 7: Comparison of accuracy and recall between HNC and the proposed refined gradient-CNN model

Methods	**Accuracy (%)**	**Recall (%)**
HNC [25]	93.38	94.51
Refined Gradient-CNN [Proposed]	**97.51**	**98.43**

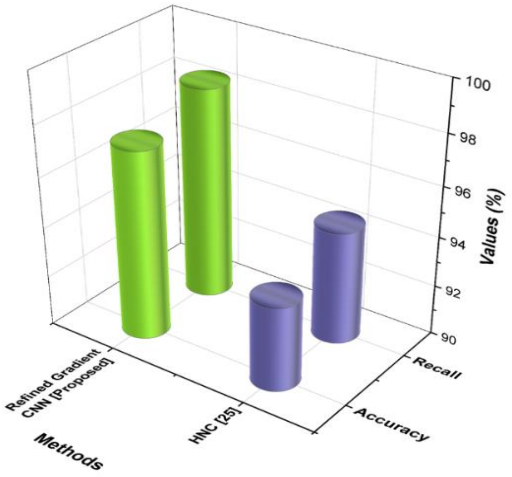

Figure 7: Performance metrics of various models in translating complex phrase patterns

5 Discussion

Enhancing the quality of machine translation at the phrase level, particularly when translating difficult and syntactically complicated English formulations. For lengthier phrases, existing models such as Improved LSTM [24] have not been adequate in terms of structural equivalence and semantic coherence. Similarly, because there are fewer contextual generalizations, the HNC model [25] is unable to interpret nested and specialized phrase patterns.

The suggested Refined Gradient-CNN model effectively overcomes these drawbacks by integrating gradient-driven refinement into the convolution process for improved recognition of complex language structure. By reducing ambiguity in translation and enhancing contextual knowledge, the approach increases the dependability of machine translation systems for technical and professional communication procedures. The design is a significant improvement over earlier models in terms of structure, recall, and generalization.

Limitation

It requires an enormous amount of processing power to train models to handle complicated, lengthy words. Large-scale models could require a lot of memory and take a long time to learn, rendering them unavailable to people or organizations with limited resources. There are many possible valid translations for a complicated statement, and the context or specific use conditions determine the preferred translation. This flexibility is hard to replicate with machine translation processes and to reasonably reflect the learning.

6 Conclusion

To achieve notable gains in semantic retention and translation quality, the English complex long sentence machine translation architecture (MTA) is optimised using Refined Gradient-CNN (RG-CNN), which was trained on a specially constructed parallel corpus of 1,563 sentence pairs enhanced with readability scores, complexity labels, and domain-specific metadata. RG-CNN effectively models structural intricacy, idiomatically controlled use, and long-distance relationships in English by fusing deep convolutional neural architecture with gradient-based smooth optimisation. The model is better able to generalise across contexts and adjust to complex language patterns when it is exposed to a heterogeneous, metadata-annotated corpus. Empirical evaluation confirms the model's excellent performance capability. With BLEU scores of 70.1% (local) and 73.1% (corpus). WER was also reduced to 0.3% (corpus) and 0.10% (local) compared to the improved LSTM [24] model. RG-CNN beat traditional models like HNC [25] in terms of classification performance, achieving 97.51% accuracy and 98.43% recall. Hyperparameter tweaking was also used to increase the model's parameter efficiency in order to get optimal convergence and significantly reduce overfitting. The RG-CNN model improved its translation ranking performance by using the Parallel Corpus dataset for bidirectional training with TransE + MLP. The model converted simple and complicated words into vectors and rated them based on their proximity to the right response. The model achieved 84 percent accuracy (P@1), 88 percent (P@5), and 98 percent (P@10), proving its effectiveness in NLP and text simplification tasks related to education. These outcomes collectively support the objective of building an efficient, high-performance model for simplifying complex English sentences across educational and NLP applications.

Future scope

The new strategy is utilizing the power of big linguistic data and optimization methods in solving the issues of translating intricate, long sentences, which will end up improving the overall performance of the machine translation system.

Acknowledgement: The Research is supported by: Construction of Training Modes for Business English Majors: A Perspective of Business Needs（ 2021HXXM175)

References

[1] Li, G., 2024. Research on Automatic Identification of Machine English Translation Errors Based on Improved GLR Algorithm. *Informatica*, *48*(6). https://doi.org/10.31449/inf.v48i6.5249

[2] Ruan, Yuexiang. 2022. “Design of Intelligent Recognition English Translation Model Based on Deep Learning.” *Journal of Mathematics* 2022: 1–10. https://doi.org/10.1155/2023/9893016.

[3] Wang, Xi. 2021. “Translation Correction of English Phrases Based on Optimized GLR Algorithm.” *Journal of Intelligent Systems* 30 (1): 868–80. https://doi.org/10.1515/jisys-2020-0132.

[4] Zhang, Qiang. 2022. "Cross-Context Accurate English Translation Method Based on the Machine Learning Model." *Mathematical Problems in Engineering* 2022: 1–11. https://doi.org/10.1155/2022/9396650.

[5] Quoc, T.N., Le Thanh, H. and Van, H.P., 2023. Khmer-Vietnamese Neural Machine Translation Improvement Using Data Augmentation Strategies. *Informatica*, *47*(3). https://doi.org/10.31449/inf.v47i3.4761

[6] Liang, J., and M. Du. 2022. "Two-Way Neural Network Chinese-English Machine Translation Model Fused with Attention Mechanism. Ding B, Editor. Scientific Programming" 2022: 1–11. https://doi.10.1155/2022/9143845

[7] Shen, Xiaoping, and Runjuan Qin. 2021. "Searching and Learning English Translation Long Text Information Based on Heterogeneous Multiprocessors and Data Mining." *Microprocessors and Microsystems* 82 (103895): 103895. https://doi.org/10.1016/j.micpro.2021.103895.

[8] Li, Xiaoyu. 2022. "The Impact of Big Data Technology on Phrase and Syntactic Coherence in English Translation." *Mathematical Problems in Engineering* 2022: 1–11. https://doi.org/10.1155/2022/1428748.

[9] Suleiman, D., Etaiwi, W. and Awajan, A., 2021. Recurrent neural network techniques: Emphasis on use in neural machine translation. *Informatica*, *45*(7). https://doi.org/10.31449/inf.v45i7.5267

[10] Rawat, R., Raj, A.S.A., Chakrawarti, R.K., Sankaran, K.S., Sarangi, S.K., Rawat, H. and Rawat, A., 2024. Enhanced Cybercrime Detection on Twitter Using Aho-Corasick Algorithm and Machine Learning Techniques. *Informatica*, *48*(18). http://dx.doi.org/10.31449/inf.v48i18.6272

[11] Farooq, Uzma, Mohd Shafry Mohd Rahim, and Adnan Abid. 2023. "A Multi-Stack RNN-Based Neural Machine Translation Model for English to Pakistan Sign Language Translation." *Neural Computing & Applications* 35 (18): 13225–38. https://doi.org/10.1007/s00521-023-08424-0.

[12] Mahata, Sainik Kumar, Avishek Garain, Dipankar Das, and Sivaji Bandyopadhyay. 2022. "Simplification of English and Bengali Sentences for Improving Quality of Machine Translation." *Neural Processing Letters* 54 (4): 3115–39. https://doi.org/10.1007/s11063-022-10755-3.

[13] Bensalah, Nouhaila, Habib Ayad, Abdellah Adib, and Abdelhamid Ibn El Farouk. 2022. "Transformer Model and Convolutional Neural Networks (CNNs) for Arabic to English Machine Translation." In *Proceedings of the 5th International Conference on Big Data and Internet of Things*, 399–410. Cham: Springer International Publishing. DOI:10.1007/978-3-031-07969-6_30

[14] Yang, Jing, and Lina Fan. 2023. "Optimization Strategy of Machine Translation Algorithm for English Long Sentences Based on Semantic Relations." In *Lecture Notes on Data Engineering and Communications Technologies*, 572–79. Singapore: Springer Nature Singapore. DOI 10.21203/rs.3.rs-5734365/v1

[15] Song, Xin. 2021. "Intelligent English Translation System Based on Evolutionary Multi-Objective Optimization Algorithm." *Journal of Intelligent & Fuzzy Systems* 40 (4): 6327–37. https://doi.org/10.3233/jifs-189469.s

[16] Cao, Qianyu, and Hanmei Hao. 2021. "A Chaotic Neural Network Model for English Machine Translation Based on Big Data Analysis." *Computational Intelligence and Neuroscience* 2021: 3274326. https://doi.org/10.1155/2021/3274326.

[17] Yu, Jinlin, and Xiuli Ma. 2022. "English Translation Model Based on Intelligent Recognition and Deep Learning." *Wireless Communications and Mobile Computing* 2022: 1–9. https://doi.org/10.1155/2022/3079775.

[18] Li, H., and W. Xiong. 2022. "Analysis of the Drawbacks of English-Chinese Intelligent Machine Translation Based on Deep Learning." In *The 2021 International Conference on Machine Learning and Big Data Analytics for IoT Security and Privacy: SPIoT-2021*, 1:104 11. Springer International Publishing. DOI:10.2478/amns-2025-0565

[19] Dong, Z. 2022. *Research on Machine Translation Method of English-Chinese Long Sentences Based on Fuzzy Semantic Optimization. Mobile Information Systems*. DOI:10.1155/2022/4863623

[20] Shao, D., and R. Ma. 2022. *English Long Sentence Segmentation and Translation Optimization of Professional Literature Based on Hierarchical Network of Concepts. Mobile Information Systems*. DOI:10.1155/2022/3090115

[21] Guo, Xiaohua. 2022. "Optimization of English Machine Translation by Deep Neural Network under Artificial Intelligence." *Computational Intelligence and Neuroscience* 2022: 2003411. https://doi.org/10.1155/2022/2003411

[22] Garg, K.D., Shekhar, S., Kumar, A., Goyal, V., Sharma, B., Chengoden, R. and Srivastava, G., 2022. Framework for handling rare word problems in neural machine translation system using multi-word expressions. *Applied Sciences*, *12*(21), p.11038. https://doi.org/10.3390/app122111038

[23] Benkova, L., Munkova, D., Benko, Ľ. and Munk, M., 2021. Evaluation of English–Slovak neural and statistical machine translation. *Applied Sciences*, *11*(7), p.2948. https://doi.org/10.3390/app11072948

[24] He, H., 2023. An intelligent algorithm for fast machine translation of long English sentences. *Journal of Intelligent Systems*, *32*(1), p.20220257. https://doi.org/10.1515/jisys-2022-0257

[25] Shao, D. and Ma, R., 2022. English long sentence segmentation and translation optimization of professional literature based on a hierarchical network of concepts. *Mobile Information Systems*, *2022*(1), p.3090115. https://doi.org/10.1155/2022/3090115