Zahra Moradi

Improving Machine Translation Output with Lightweight Preprocessing and CNN-Based Quality Estimation

Ensuring high-quality output in Neural Machine Translation (NMT) systems remains a central challenge, especially in applications with critical fluency, grammatical accuracy, and semantic fidelity. While significant advancements have been made in model architecture, less attention has been given to the role of data preprocessing and post-translation evaluation, both of which are essential for enhancing translation reliability and scalability. This study introduces a dual-stage framework that integrates linguistic preprocessing techniques and a Convolutional Neural Network (CNN)-based classifier for translation quality assessment. We use the English–Spanish Translation Dataset by Lonnie Qin to train a lightweight Seq2Seq NMT model with and without preprocessing steps, including tokenization, lowercasing, lemmatization, and normalization. Translation quality is evaluated using BLEU, METEOR, and ROUGE metrics. A 1D CNN is trained as a lightweight post-translation screening model using BLEU-derived weak labels for large-scale binary supervision and a smaller human-labeled subset for finer-grained validation. The purpose of this classifier is not merely to reproduce a fixed BLEU cutoff, but to learn reusable quality patterns that can support rapid quality flagging in settings where repeated reference-based evaluation or manual review is impractical. It is important to note that the CNN is not used as the translation generator itself, but as a lightweight post-translation quality estimation module, selected to examine whether reliable sentence-level quality screening can be achieved in resource-conscious and real-time settings without resorting to heavier transformer-based classifiers. Experimental results demonstrate that preprocessing consistently improves the translation accuracy of this lightweight model, with BLEU scores increasing by 8.4 points and METEOR by 4 points; these gains are specific to the simple Seq2Seq model studied here and are not claimed to transfer to stronger architectures. The CNN classifier achieves 88.7% accuracy in binary classification and a macro F1-score of 0.82 in multi-class evaluation. The integrated pipeline improves both the generation and validation of translations, making it suitable for real-time quality assurance and educational use cases. The proposed approach highlights the often-underestimated impact of preprocessing and the efficiency of CNNs in evaluating translation quality. Together, they form a robust, scalable, and adaptable solution for improving translation outputs. This framework offers substantial potential for deployment in educational and professional language-support settings, while domain-specific applications such as legal or medical translation require further validation on specialized corpora.

Keywords: Neural Machine Translation (NMT), Translation Quality Assessment, Convolutional Neural Networks (CNN), Deep Learning, Machine Translation Validation

1 Introduction

Machine Translation (MT) has historically been a fundamental aspect of natural language processing (NLP), aiming to transform text or speech from one language to another automatically (H. Wang et al., 2022). The domain has significantly progressed in recent decades, transitioning from early rule-based methods to contemporary statistical and neural models. The rise of Neural Machine Translation (NMT) has significantly transformed the domain, superseding conventional phrase-based models with comprehensive deep learning frameworks that can learn language representations directly from extensive datasets (Khan et al., 2023). Sequence-to-sequence (Seq2Seq) architectures with attention mechanisms, along with the more recent transformer-based models, have demonstrated impressive results in numerous high-resource language pairs (H. Wang et al., 2023). Their fluency, grasp of context, and capacity to produce human-like responses have led to their widespread use in both academic research and commercial applications.

Even with the achievements of neural models, ensuring uniform translation quality remains a significant obstacle, especially in practical applications. A concern is that the effectiveness of NMT systems relies significantly on the training data's quality, quantity, and domain-related pertinence. For example, systems developed using general-purpose datasets might handle regular conversations well but struggle when used with specialized materials like legal, medical, or technical texts (Ibraheem et al., 2017). Additionally, low-resource language pairs or informal linguistic structures worsen the problem, as models often hallucinate or produce grammatically accurate but semantically flawed translations (Stasimioti et al., 2020). Even in high-resource language pairs such as English-Spanish, minor mistakes like idiomatic discrepancies, sentiment loss, wrong verb conjugations, or syntactic misalignments can greatly diminish translation quality and user confidence (S. Chen et al., 2022).

A crucial aspect affecting the performance of NMT systems is the quality and consistency of the input data, which can be greatly enhanced by applying preprocessing methods (Fan et al., 2021). Preprocessing in NLP involves a set of actions taken on unprocessed textual data prior to its input into a model (Egger & Gokce, 2022). These processes might involve converting to lowercase, removing punctuation, tokenizing, filtering out stop words, stemming, and lemmatizing. Although these methods may seem straightforward, they are essential in decreasing linguistic noise, standardizing phrases, reducing vocabulary size, and enhancing sentence structure (Kaur & Sohal, 2024). Through the simplification and standardization of the source data, preprocessing enables the model to concentrate on uncovering significant patterns and associations between words and phrases instead of being sidetracked by spelling inconsistencies or unrelated tokens (Chai, 2023). Moreover, preprocessing enhances the alignment between sentences in the source and target languages, which is particularly crucial in parallel corpora for translation tasks (Mashtalir & Nikolenko, 2023).

Alongside enhancing the translation results, it is crucial to establish effective systems for the automatic assessment of translated text quality. Historically, the evaluation of translations has depended on human assessment, which, while precise, is subjective, costly, and lacks scalability for extensive datasets (Freitag et al., 2021). Automated metrics like BLEU (Bilingual Evaluation Understudy), METEOR, ROUGE, and TER have emerged as standard instruments for assessing MT results (Chauhan & Daniel, 2023). Nonetheless, these metrics largely depend on superficial

word overlaps and might not truly represent semantic accuracy, fluency, or contextual integrity, particularly when numerous valid translations are available. As the discipline progresses toward more context-sensitive and semantically enriched evaluation techniques, the necessity for machine learning-driven quality assessment becomes clear (Evtikhiev et al., 2023).

A promising method involves applying Convolutional Neural Networks (CNNs) to classify translation quality (Zhong et al., 2019). CNNs, while commonly linked to image analysis, have demonstrated significant effectiveness in text classification tasks because of their capability to capture local n-gram features and positional data. When used on translated texts, CNNs can recognize distinguishing features that differentiate high-quality translations from inferior ones by detecting patterns associated with word order, grammar, semantic consistency, and stylistic harmony (Mohamed et al., 2024). In contrast to manually created rule-based systems or fixed metrics, CNNs benefit from learning directly from data, adjusting to various translation styles, language pairs, and quality levels according to training instances (Shahin & Ismail, 2024). They are efficient in computation, simple to implement, and highly scalable, which makes them appropriate for incorporation into real-time MT workflows (Islam et al., 2021).

While prior research has either focused on improving translation output through increasingly complex neural architectures or evaluating translations through static or human-based methods, few studies have combined these critical aspects, enhancement and evaluation, into a unified, data-driven framework. Additionally, much of the attention in improving translation output has centered on architectural sophistication, such as larger transformer models, rather than upstream improvements in input data quality through preprocessing. This often leads to resource-intensive solutions that may be impractical for real-time or low-budget applications. At the same time, evaluation systems have lagged in terms of leveraging modern deep learning techniques that could offer adaptive and nuanced assessments of translation quality (Gao, 2025).

To tackle these shortcomings, this paper proposes a practical framework that combines preprocessing on the input side with quality evaluation on the output side using a CNN-based classifier. Importantly, the CNN is not employed as the translation model itself; translation generation is performed by a Seq2Seq model with attention, whereas the CNN is used only for post-translation quality estimation. The goal of the study is therefore not to benchmark all modern architecture families or to claim superiority over transformer-based alternatives, but to test whether a lightweight and computationally efficient evaluation module can provide reliable quality screening when paired with linguistically informed preprocessing. This design choice is motivated by deployment-oriented settings in which inference speed, implementation simplicity, and reduced computational cost remain important. Within this scope, we train a translation model with raw and preprocessed data and then evaluate translated outputs using a supervised CNN quality estimator. The specific contribution does not lie in preprocessing alone, which could equally be assessed with BLEU, METEOR, or human ratings, nor in the CNN classifier alone, but in their pairing under a single resource-conscious constraint. Both components operate on the same local lexical and syntactic regularities: preprocessing reshapes these surface patterns on the input side so that the lightweight generator produces more regular output, and the convolutional estimator screens for exactly these local patterns on the output side without requiring a gold reference at inference time. The combination therefore yields a compact generate-and-verify loop in which the same inexpensive pipeline both improves translations and flags low-quality ones, which is the aspect of

the design that is examined empirically in this work.

This study adds to the machine translation domain by providing a practical, scalable, and understandable framework that enhances translation results and automates quality evaluation. It emphasizes the frequently ignored strength of preprocessing and demonstrates how lightweight CNN-based models can serve as effective learned screening components within a translation-quality workflow. This research not only connects translation improvement and assessment but also paves the way for creating efficient, deployable systems for practical multilingual interaction.

2 Related work

This section examines the key studies that establish the theoretical and technical foundation for present research. It covers earlier progress in neural machine translation (NMT), the role and effects of preprocessing methods in both NLP and translation endeavors, advances in translation quality estimation (TQE), along with the implementation of CNNs for text classification and assessment of translation-associated data. The concluding section highlights a gap in the current literature that drives the present research (Calzolari et al., 2022; Gao & Bu, 2024)

2.1 Neural machine translation

The arrival of neural models has fundamentally changed the terrain of machine translation. The sequence-to-sequence (Seq2Seq) architecture presented by (Sutskever et al., 2014) marked one of the first innovations in neural machine translation. This architecture uses two recurrent neural networks (RNNs): a decoder produces output sequences using fixed-length context vectors generated from an encoder converting input sequences. Initially successful, the fixed-size context representation of this technique limited the handling of extended sequences (Khalil & Pipa, 2022; Scotti et al., 2023).

Bahdanau et al. (2016) developed the attention mechanism to let the decoder concentrate on particular sections of the source sequence during translation (Chorowski et al., 2015), so this is addressed. By allowing dynamic alignment between input and output tokens, attention greatly enhanced performance, lowering information loss for long and complicated sentences.

A big step forward was taken when Vaswani et al. (2017) developed the Transformer architecture. Transformers rely entirely on attention mechanisms and avoid recurrence entirely, unlike RNN-based Seq2Seq models. While enhancing translation quality, multi-head self-attention, positional encodings, and feed-forward layers accelerated training speed and parallelizable capability. Particularly in multilingual and low-resource environments, transformer models such as BERT, GPT, and T5 have since been modified and fine-tuned for different machine translating chores (Sanni, 2021; Zayyanu, 2024). Strong multilingual NMT systems spanning hundreds of language pairs have also been made possible by open-source implementations such as MarianMT and mBART.

These models are accurate; however, most NMT systems are still sensitive to vocabulary size and data quality (Costa-jussà et al., 2024; Durrani et al., 2019). For researchers or developers working in constricted surroundings, their need for large datasets and computational tools can often be restricted. Furthermore, even if architectural advancements push forward, modern NMT processes mostly rely on data-level improvements, including preprocessing (Zhou et al., 2025).

2.2 Role of preprocessing in NLP and MT

Preprocessing has traditionally been a crucial step in natural language processing activities, playing a vital part in structuring and preparing text data for analysis or model training (Eisenstein, 2019). Techniques such as converting to lowercase, breaking text into tokens, removing punctuation, eliminating stop words, stemming, and using lemmatization are often employed to normalize text, reduce vocabulary sparsity, and remove irrelevant information (Kulkarni & Shivananda, 2019).

In traditional statistical machine translation (SMT), preprocessing was crucial as it relied on phrase tables and alignment scores that were highly susceptible to minor changes. Despite the increased resilience of NMT models to anomalies, studies by Domingo et al. (2023) and Camacho-Collados and Pilehvar (2018) show that proper preprocessing can lead to improved convergence speeds, more excellent training stability, and enhanced generalization in NMT systems (Deep et al., 2021; C. Wang et al., 2018). For instance, tokenization helps break words into smaller, more manageable sections, especially in languages with intricate morphology, while converting to lowercase and removing punctuation reduces the risk of overfitting rare or exceptional tokens. Preprocessing is particularly beneficial in low-resource environments where data augmentation is limited and in specialized translation tasks where precision and consistency are crucial. Nonetheless, the importance of preprocessing in modern NMT has mostly been overlooked due to the focus on model complexity, leading to a scarcity of research that evaluates its true impact on translation quality (Brook, 2023).

2.3 Translation quality estimation (manual and automated methods)

Evaluating the quality of text produced by machine translation is a challenging but crucial job, particularly when these systems are used in critical areas. Historically, human expert evaluation has been regarded as the benchmark of excellence. Metrics like adequacy, fluency, and fidelity are evaluated on a Likert scale by reviewers who are either bilingual or experts in the domain. Although precise and detailed, manual assessments are labor-intensive, costly, and fundamentally subjective (Castilho et al., 2018).

Automated evaluation metrics have been widely embraced to facilitate scalable and uniform assessment. The BLEU score (Papineni, 2002) evaluates the overlap of n-grams between translations produced by machines and reference translations. Although widely used, BLEU has limitations regarding its sensitivity to semantic correctness and stylistic suitability. METEOR, ROUGE, and TER were developed to enhance BLEU by integrating stemming, synonym matching, and edit distance, respectively (Tamine & Goeuriot, 2021). However, all these metrics depend significantly on lexical similarity and do not reflect deeper semantic equivalence.

Quality estimation (QE) based on machine learning has recently become increasingly popular. It seeks to forecast translation quality independently of reference translations. As an illustration, the WMT QE shared tasks have promoted the creation of regression and classification models designed to predict quality scores at the sentence or word level. Some systems rely on manually crafted features such as alignment scores and language model probabilities, whereas more recent models take advantage of sentence embeddings or fine-tuned BERT encoders (Mayfield & Black, 2020). Nevertheless, these techniques frequently necessitate extensive, annotated QE datasets and their applicability across various languages and fields still raise concerns.

2.4 CNNs for text classification or translation-related tasks

Initially designed for image recognition tasks, CNNs have been effectively modified for numerous text processing uses, such as text classification, sentiment analysis, and question answering (Umer et al., 2023). In the realm of NLP, CNNs work with word embeddings or character embeddings, employing convolutional filters to identify local features like bi-grams and tri-grams throughout sequences. The power of CNNs is rooted in their capability to identify local patterns in text, rendering them exceptionally efficient for classification tasks where the meaning at the phrase level is vital.

In machine translation processes, CNNs have been used in tasks like automatic post-editing, detecting translation errors, and classifying quality. For instance, Kim et al. (2016) showed that a straightforward CNN architecture could surpass conventional techniques in sentence classification tasks. Subsequent research expanded on this by using CNNs to identify poor-quality translations through classification based on syntactic coherence and lexical consistency.

In contrast to recurrent architectures such as LSTMs and GRUs, CNNs are quicker in computation, need fewer parameters, and are simpler to parallelize. They are ideal for real-time systems requiring instant quality assessment of translated results. Although transformers have become the leading model in various NLP tasks, CNNs are still appealing for lightweight, deployable systems needing quick and precise classification abilities (Fields et al., 2024).

Although transformer-based encoders have become dominant in many NLP classification settings because of their stronger contextual modeling capacity, they also introduce greater computational and implementation overhead, especially when fine-tuning and deploying them for repeated sentence-level quality screening. For this reason, CNNs remain relevant when the task primarily requires fast detection of local lexical and syntactic quality patterns and when the research objective is lightweight deployment rather than state-of-the-art architecture benchmarking. The present study is positioned within this latter setting and therefore uses CNN as a deliberate design choice rather than as a claim that transformer-based estimators are unimportant.

2.5 Gap in literature that this paper addresses

Although the literature provides a strong basis for machine translation and quality assessment, a significant gap exists between improving translation and evaluating quality. Most research focused on enhancing translation quality often achieves this by modifying or broadening neural architectures, such as adding deeper layers, using hybrid models, or implementing multilingual pretraining while neglecting the possible advantages of improving data through preprocessing. Consequently, the significance of input normalization in enhancing model performance is not thoroughly investigated, especially in neural settings where data inconsistencies continue to impact learning greatly.

Regarding evaluation, although rule-based and metric-based systems such as BLEU and METEOR are commonly employed (Sai et al., 2022), they provide minimal understanding of contextual mistakes or stylistic decline. Deep learning methods for quality estimation, including CNNs or transformers, have demonstrated potential but are frequently regarded as individual tools instead of parts of a unified MT pipeline.

Moreover, scarcely any studies combine preprocessing-driven enhancement with CNN-based classification within a cohesive translation process. This exclusion creates a methodological void: a chance to investigate how basic preprocessing methods can enhance output quality and how CNNs can be employed for detection and automatically verify these enhancements.

The present research tackles this deficiency by introducing a two-phase system that integrates organized preprocessing methods to enhance NMT results alongside a CNN-based classifier for evaluating the quality of translated content. In this way, this paper presents a comprehensive, resource-effective, and scalable framework that improves the generation and assessment of machine translation.

3 Materials and methods

In this section, we detail the methodology adopted in our study, which involves the creation of a translation and evaluation pipeline for English-Spanish language pairs. The system integrates preprocessing techniques to improve translation quality and uses a Convolutional Neural Network (CNN) for automatic quality assessment. Our pipeline consists of five main components: the dataset, preprocessing stages, translation model architecture, evaluation metrics, and the CNN-based classification system. Each component is designed to function independently and synergistically within the workflow to produce, refine, and assess translations in a scalable and replicable manner.

3.1 Dataset description

The dataset used in this research is the publicly available English-Spanish Translation Dataset curated by Lonnie Qin and hosted on Kaggle. The version used in this study contains approximately 130,000 English-Spanish sentence pairs, with each record consisting of one English sentence and its corresponding Spanish translation. The public Kaggle listing provides the dataset name and hosting information but does not include a detailed dataset description or fully documented provenance metadata such as original source corpus, collection year, or domain annotation. Accordingly, the dataset is treated in this study as a public bilingual sentence-pair resource for English-Spanish translation, rather than as a formally characterized legal, scientific, or medical corpus.

The sentences vary in length, structure, and syntactic complexity, ranging from simple greetings such as ”How are you?” to more compound constructions like ”We are planning to meet the delegation after lunch.” The diversity of sentence structure makes the dataset suitable for training and evaluating sequence-based translation models.

Before processing, we conducted an initial exploration analysis. Sentences with null values, corrupted characters, or excessive length (e.g., more than 50 tokens) were removed during corpus cleaning in order to exclude unusually long parallel pairs that could destabilize training and increase padding inefficiency in the Seq2Seq model. We also discarded sentence pairs where the source or target was composed entirely of stop words or repeated tokens. After cleaning, the dataset was randomly shuffled to eliminate any bias related to sequential sentence order and was split into three subsets: 80% for training, 10% for validation, and 10% for testing. The training set consisted of approximately 104,000 sentence pairs, the validation set contained 13,000, and the test set also

Table 1: Sample Dataset Structure

English Sentence	Spanish Sentence	English Token Count*	Spanish Tokens Count*	English Char Count**	Spanish Char Count**
I am going to school.	Voy a la escuela.	5	4	21	17
He plays the guitar very well.	Él toca la guitarra muy bien.	6	6	30	29
She didn't like the movie.	A ella no le gustó la película.	5	7	26	31

* Token counts are reported at the word level for descriptive purposes in this table.
** Character counts were calculated on the original sentence strings, including spaces and terminal punctuation.

held 13,000 sentence pairs. This split ensured the model was exposed to a wide range of linguistic patterns during training and could be robustly validated and tested on unseen data.

Because Kaggle release does not provide explicit domain labels or detailed source provenance, the present experiments should be interpreted as demonstrating performance on a public general bilingual sentence-pair dataset, not as direct validation on specialized domains such as legal, scientific, or medical translation. Any extension of the framework to those domains would require additional evaluation on appropriately curated domain-specific corpora.

A sample of the dataset structure is presented in Table 1. To avoid ambiguity in bilingual length reporting, the table shows English and Spanish token counts separately, together with character counts calculated directly from the original sentence strings, including spaces and terminal punctuation.

3.2 Preprocessing pipeline

Preprocessing was conducted to improve the quality and uniformity of data input to the neural machine translation model (Sarkar et al., 2024). This process significantly influences how well the model can generalize and converge during training (Maharana et al., 2022). The final preprocessing pipeline implemented in the experiments consisted of data cleaning, tokenization, lowercasing, lemmatization, and punctuation/text normalization, executed using Python libraries such as spaCy, NLTK, and regex (Altinok, 2021). Stop-word removal and stemming were explored in preliminary ablation tests but were not retained in the final pipeline.

Tokenization was the first step, where both English and Spanish sentences were split into individual tokens based on linguistic rules. We used spaCy's language-specific tokenizers (en_core_web_sm for English and es_core_news_sm for Spanish) to preserve syntactic correctness and handle contractions, punctuation, and clitics effectively. Tokenized outputs were saved in new columns for further processing (Danenas & Skersys, 2022).

Lowercasing all characters helped reduce vocabulary sparsity by treating uppercase and lowercase variants of the same word as a single token. For instance, "House" and "house" would be normalized to "house." This step is beneficial for deep learning models with limited embedding dimensions.

Although controversial in machine translation, stop-word removal was explored in an ablation study. Stop words were removed from a duplicate training set to evaluate the impact on translation fluency and alignment. Stop word lists from NLTK and spaCy were used and tailored for both

languages (Spring & Johnson, 2022). In the final experimental pipeline, stop-word removal was not applied because many stop-words carry grammatical information essential for sentence integrity and translation adequacy.

Lemmatization was applied using the same spaCy language-specific models employed for tokenization (en_core_web_sm and es_core_news_sm), which derive lemmas from part-of-speech and morphological analysis rather than from surface affix stripping. It was used to normalize morphological variants of words in the final pipeline. Stemming was examined only in preliminary trials and was not retained because lemmatization preserved semantic meaning more reliably. For example, lemmatization maps the irregular forms "went" to "go" and "better" to "good," whereas a rule-based stemmer such as Porter or Snowball leaves these irregular forms unchanged and crudely truncates regular ones—for instance reducing "studies" to "studi" or Spanish "corriendo" to "corr"—producing inconsistent and sometimes non-existent word forms; lemmatization was therefore preferred. This step reduced vocabulary size, especially in Spanish, where verb conjugations and noun-adjective agreements introduce substantial morphological variation. We note that lemmatization necessarily discards some inflectional detail; in this work it was used to curb sparsity arising from rare surface variants rather than to strip all morphology, and the empirical gains reported in Sections 4.1 and 4.3 indicate that, for the lightweight Seq2Seq model studied here, the reduction in sparsity outweighed the loss of fine-grained morphological cues. As discussed in Section 5, this trade-off is model- and language-dependent and may not hold for stronger models or for morphologically richer target languages.

Punctuation removal and normalization include the elimination of extraneous punctuation, such as multiple exclamation marks or ellipses, which do not typically affect semantic content but increase input variability. We preserved sentence-final punctuation (periods, question marks) when relevant for model alignment. Text normalization also included converting multiple spaces into a single space, removing non-printable characters, and standardizing quotation marks and apostrophes (Bilal et al., 2022; Falgueras et al., 2010; Tian et al., 2018).

The implemented preprocessing pipeline used in the final experiments is summarized in Figure 1, which visually represents each transformation stage from the raw bilingual corpus to the final standardized input used for model training.

3.3 Translation architecture

The translation model adopted for this study is based on a sequence-to-sequence (Seq2Seq) architecture with an attention mechanism implemented using TensorFlow 2.0 (T. Chen, 2024; Raj, 2021). The architecture consists of two key components: an encoder, which processes the source (English) sentence, and a decoder, which generates the translated (Spanish) sentence.

The encoder comprises an embedding layer followed by a bidirectional Long Short-Term Memory (BiLSTM) layer (Hameed & Garcia-Zapirain, 2020; Naik & Jaidhar, 2022). The input sequence $X = (x_1, x_2, \ldots, x_n)$ is transformed into dense vectors through the embedding layer and then passed through the BiLSTM to capture both past and future context.

The hidden states of the encoder are used to compute the attention scores. The Bahdanau attention mechanism dynamically focused on relevant encoder states during decoding. The attention scores $\alpha_{t,i}$ are calculated using the SoftMax function:

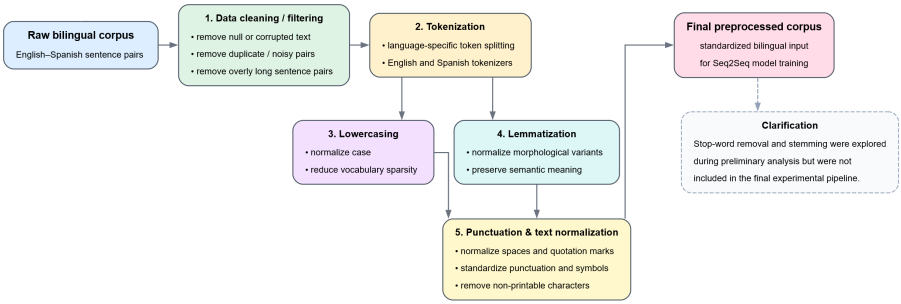

Figure 1: Preprocessing Pipeline

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{k=1}^{n} \exp(e_{t,k})} \tag{1}$$

where the alignment score $e_{t,i}$ is computed as:

$$e_{t,i} = v^{\top} \tanh(W_s\, s_{t-1} + W_h\, h_i) \tag{2}$$

with s_{t-1} being the decoder's previous hidden state and h_i being the encoder's hidden state. The context vector c_t is then derived as a weighted sum of the encoder's outputs.

The decoder also consists of an embedding layer and a unidirectional LSTM layer, followed by a dense output layer with SoftMax activation to produce probabilities for the target vocabulary.

We used the following hyperparameters for model training:

- Embedding dimension: 256
- LSTM hidden units: 512
- Dropout rate: 0.3
- Batch size: 64
- Optimizer: Adam with a learning rate of 0.001
- Loss function: Sparse categorical cross-entropy
- Epochs: 20 (with early stopping based on validation loss)

Model training was performed on an NVIDIA GPU (RTX 3080) for approximately 2.5 hours. After each epoch, model outputs were monitored on the validation set using BLEU and METEOR as complementary evaluation metrics. The best checkpoint was selected based on the validation BLEU score, whereas METEOR was retained as a supplementary diagnostic metric to verify that improvements were not limited to n-gram overlap alone.

3.4 Translation evaluation metrics

To evaluate the performance of the translation model, we adopted three automatic evaluation metrics: BLEU, METEOR, and ROUGE, and one optional human evaluation.

BLEU (Bilingual Evaluation Understudy) is a precision-based metric that measures the degree of n-gram overlap between a machine-generated translation and one or more reference translations (Datta et al., 2022). We calculated BLEU-1 through BLEU-4 using the sacrebleu library to ensure consistency and reproducibility. A smoothing function was applied to mitigate issues with short sentences.

METEOR incorporates both precision and recall, using stemming and synonym matching through WordNet to capture semantic similarity beyond surface-level matching. METEOR correlates more closely with human judgments than BLEU, particularly for languages with flexible word order (Lavie & Denkowski, 2009).

In the present study, BLEU and METEOR were used for different methodological purposes. BLEU was used as the primary checkpoint-selection metric for the translation model in order to maintain consistency with the downstream binary quality-labeling setup, which was also based on BLEU-derived thresholds. METEOR was computed in parallel as a complementary semantic-sensitive metric, allowing us to confirm that observed gains were not restricted to surface-level n-gram overlap. Accordingly, METEOR served an interpretive and corroborative role rather than a model-selection role.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation), primarily used in summarization (Akter et al., 2022), was applied here to assess how much of the reference content was successfully captured in the output. We computed ROUGE-1 and ROUGE-L scores.

Human evaluation was conducted on a sample of 500 translated sentences. Each was rated by bilingual annotators on a 5-point Likert scale for fluency (grammatical correctness and naturalness) and adequacy (semantic preservation). These scores were later averaged and used for CNN classification training as soft labels.

The three automatic metrics defined above (BLEU, METEOR, and ROUGE-L) were computed for both the raw and the preprocessed systems; the resulting comparison is reported and analysed in Section 4 (Table 4).

3.5 CNN-based translation quality assessment

To automatically evaluate the quality of translations generated by the Seq2Seq neural translation model, we implemented a CNN-based classifier as a lightweight post-translation screening module. This classifier was not intended to replace reference-based metrics in offline benchmarking when gold translations are available; rather, it was designed to reduce reliance on repeated manual inspection and to support rapid sentence-level quality flagging in deployment-oriented settings. CNNs are highly effective in capturing local patterns and n-gram level features in sequential data, making them suitable for sentence-level classification tasks in NLP (T. Chen et al., 2017; Patil et al., 2023).

In this study, CNN was selected because the objective was to evaluate a lightweight post-translation screening module that can be trained and deployed with lower computational overhead than transformer-based alternatives. Accordingly, the methodological scope of this work is a

Table 2: Representative examples of BLEU-derived binary quality labels used in CNN training.

Source Sentence (English)	Reference Translation (Spanish)	Model Output (Spanish)	BLEU Score	Quality Label	Linguistic Analysis
The children are playing in the garden.	Los niños están jugando en el jardín.	Los niños están jugando en el jardín.	0.91	High	Perfect lexical match; correct verb tense and preposition usage; fluent and natural structure.
She gave me the book that I wanted.	Ella me dio el libro que yo quería.	Ella me dio el libro que quería.	0.82	High	Minor omission of pronoun "yo" does not affect meaning; grammatically correct and fluent.
I am not going to school today.	No voy a la escuela hoy.	No voy a la escuela hoy.	0.95	High	Accurate negation handling; correct word order and article usage; high semantic fidelity.
He wouldn't have said that under normal circumstances.	No lo habría dicho en circunstancias normales.	No lo habría dicho en circunstancias normales.	0.88	High	Correct conditional tense and idiomatic phrasing; natural and contextually appropriate.
They watched a movie that was very interesting.	Vieron una película que fue muy interesante.	Ellos vieron una película muy interesante.	0.78	High	Slight structural variation but semantically equivalent; fluent and acceptable translation.
The children are playing in the garden.	Los niños están jugando en el jardín.	Los niños jugando jardín.	0.42	Low	Missing auxiliary verb; incorrect structure; lacks grammatical completeness and fluency.
She gave me the book that I wanted.	Ella me dio el libro que yo quería.	Ella me dio libro que quería.	0.39	Low	Missing definite article; reduced syntactic clarity; less natural expression.
I am not going to school today.	No voy a la escuela hoy.	Yo no voy colegio hoy.	0.44	Low	Incorrect article usage and word order; partially understandable but grammatically weak.
He wouldn't have said that under normal circumstances.	No lo habría dicho en circunstancias normales.	Él no habría dicho eso bajo circunstancias normales.	0.48	Low	Literal translation ("bajo" vs. "en"); less idiomatic and slightly unnatural phrasing.
They watched a movie that was very interesting.	Vieron una película que fue muy interesante.	Ellos vieron película fue muy interesante.	0.36	Low	Missing article and connector; incorrect sentence structure; reduced coherence.

preprocessing-plus-lightweight-quality-estimation framework, not an exhaustive benchmark across all contemporary classifier architectures.

For classification, each sentence pair (English source and Spanish translation) was assigned a quality label based on either automatic metrics or human annotation. It should be emphasized that a gold-standard reference is required only to supervise the training of the classifier, not to apply it. Training labels were therefore derived only from data for which a reference translation (and hence a BLEU score) or a human rating was available, because supervised learning requires labeled examples. Once trained, however, the classifier estimates sentence-level quality directly from the source and candidate translation, without access to any reference at inference time. This separation between a reference-dependent training stage and a reference-free inference stage is precisely what allows the model to be reused as an operational screen in settings where gold translations are unavailable, and it is examined further as a limitation in Section 5.

In the primary experiment, we used BLEU scores as a weak-supervision signal for binary quality labeling. Translations with BLEU scores ≥ 0.7 were labeled as high-quality (label 1), and

those with scores ≤ 0.5 were labeled as low-quality (label 0). To ensure clarity in classification and reduce noise, sentence pairs with intermediate BLEU scores (between 0.5 and 0.7) were excluded from training. This setup was intended to construct a large set of clearly separated training examples for the classifier, rather than to provide a formal comparative benchmark against static BLEU-thresholding itself.

To make these binary quality categories more interpretable, representative sentence-level examples from the BLEU-labeled dataset are provided in Table 2. These examples illustrate the practical distinction between translations assigned to the high-quality group (BLEU ≥ 0.7) and those assigned to the low-quality group (BLEU ≤ 0.5), thereby clarifying the linguistic characteristics associated with each label.

The rationale for training a classifier rather than continuing to use a fixed BLEU threshold is twofold. First, BLEU requires a reference translation for each evaluated sentence and is therefore most useful in offline benchmarking or controlled experiments. Second, a trained classifier can learn reusable lexical, syntactic, and fluency-related patterns from many labeled examples and then function as a lightweight operational filter for fast quality screening. Accordingly, the binary CNN in this study should be interpreted as a weakly supervised quality-estimation component, whereas the human-labeled multi-class experiment provides a more semantically grounded complement to this initial setup.

In a secondary setup for multi-class classification, we employed human evaluation scores where annotators rated translations on a 5-point Likert scale for fluency and adequacy. Scores were then converted into categorical labels:

- *Poor* (1–2): Label 0
- *Fair* (3): Label 1
- *Good* (4): Label 2
- *Excellent* (5): Label 3

This multi-class approach enabled a more nuanced interpretation of translation quality but required a smaller, manually curated subset of the dataset.

For the CNN-based quality classification stage only, translated sentences were padded or truncated to a fixed maximum length of 40 tokens to create uniform input shapes for the classifier. This setting did not constrain the translation model itself; rather, it was introduced solely to standardize sentence representations for CNN training and inference. The value of 40 was selected as a practical fixed-length representation based on the post-preprocessing length characteristics of the translated sentences, allowing efficient batching while limiting excessive padding. Each word token was then converted to its corresponding vector representation using pre-trained FastText embeddings, which provide robust subword-level information and are effective for both English and Spanish vocabularies.

Therefore, the 50-token threshold used during dataset preparation and the 40-token threshold used for CNN input formatting are not intended to represent the same design choice: the former is a corpus-level filtering constraint, whereas the latter is a classifier-level tensor-shaping parameter.

Let each sentence $S = \{w_1, w_2, \ldots, w_T\}$ be represented as a matrix $X \in \mathbb{R}^{T \times d}$, where T is the fixed sequence length (padded), and $d = 300$ is the dimensionality of the word embeddings.

The CNN model follows a 1D convolutional design, commonly used for text classification tasks due to its effectiveness in detecting phrase-level patterns.

1. Embedding Layer:

   This layer maps each token in a sentence to a 300-dimensional FastText vector. It outputs a matrix of shape *(T,d)* for each sentence.

2. Convolutional Layers:

   We applied three separate 1D convolutional layers with kernel sizes of 3, 4, and 5, each with 100 filters and ReLU activation:

$$f_{i,j} = \mathrm{ReLU}(w_j * x_{i:i+k-1} + b_j) \tag{3}$$

where:

- w_j is the j-th filter of size k
- $x_{i:i+k-1}$ is the window of embeddings at position i
- b_j is the bias term
- $*$ denotes 1D convolution

Each convolution operation produces a feature map representing the presence of specific patterns (e.g., certain n-grams) across the sentence.

3. Global Max Pooling:

   After convolution, we apply Global Max Pooling to each feature map to retain only the most significant features:

$$p_j = \max_i f_{i,j} \tag{4}$$

This results in a fixed-size feature vector $\mathbf{p} \in \mathbb{R}^{3 \times 100}$ regardless of the input sequence length.

4. Fully Connected Layer:

   The concatenated pooled features are passed to a dense layer with 64 hidden units and ReLU activation:

$$h = \mathrm{ReLU}(W_p \cdot p + b_p) \tag{5}$$

5. Output Layer:

For binary classification, we used a sigmoid activation to predict the probability of high-quality translation:

$$\hat{y} = \sigma(W_h \cdot h + b_s) \tag{6}$$

For multi-class classification, the output layer employed a SoftMax activation to generate a probability distribution over the four classes:

$$\hat{y}_i = \frac{\exp(W_i \cdot h)}{\sum_{j=1}^{4} \exp(W_j \cdot h)} \tag{7}$$

The CNN model was compiled and trained with the following parameters:

- Loss Function:
  - Binary classification: Binary Crossentropy
  - Multi-class classification: Categorical Crossentropy
- Optimizer: Adam, with a learning rate of 0.001
- Batch Size: 32
- Epochs: 15
- Validation Split: 20%
- Dropout: 0.5 (before the dense layer, to reduce overfitting)

To monitor training, we recorded validation loss, accuracy, and F1-score at each epoch. Early stopping was applied with patience = 3 to avoid overfitting.

For both classification types, we evaluated the model using the following metrics:

- Accuracy: Overall correctness of predictions
- Precision: $\text{Precision} = \frac{TP}{TP + FP}$ (8)
- Recall: $\text{Recall} = \frac{TP}{TP + FN}$ (9)
- F1-Score: Harmonic mean of precision and recall

For both classification settings, accuracy, precision, recall, and F1-score were computed, with macro-averaging applied in the multi-class setting because of class imbalance. The resulting values are reported and discussed in Section 4 (Table 5).

Once trained, the CNN model was integrated as a post-evaluation layer into the translation pipeline. Each new translated sentence could be automatically passed through CNN to predict its quality. Sentences classified as low quality could be flagged for retraining, manual correction, or rejection, depending on the context of the application.

Moreover, this feedback loop allowed us to continuously refine both the translation model and the CNN classifier by accumulating more labeled data from user interactions or ongoing human evaluation. This design promotes a semi-supervised learning approach, where the CNN serves as a self-adaptive filter, continuously learning to predict and improve translation quality over time.

Figure 2 presents the architectural flow of the CNN model, including the embedding, convolution, pooling, dense, and output layers.

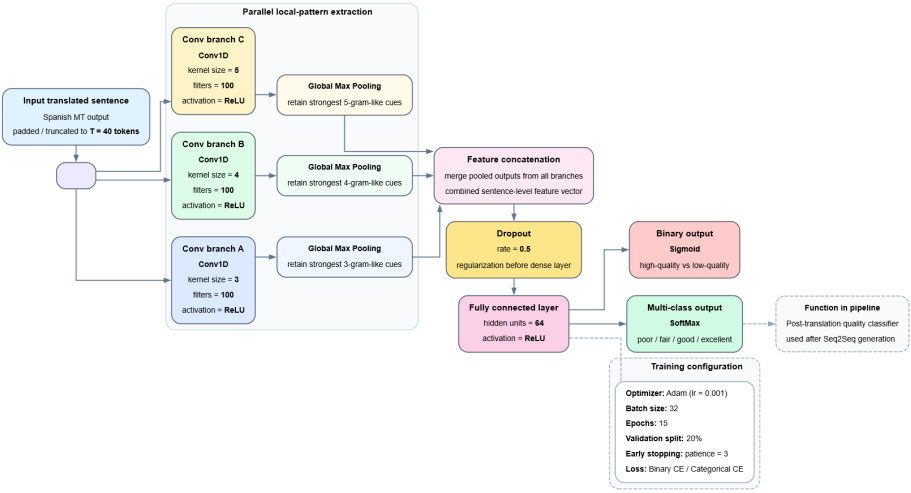

Figure 2: CNN architecture for translation quality classification.

4 Results and discussion

This section presents a detailed account of the experimental outcomes, analyzing the performance of the neural machine translation model and the convolutional neural network classifier within our integrated pipeline. The results are organized to reflect the successive phases of the system—translation modeling, quality classification, and the influence of preprocessing techniques—culminating in a comprehensive discussion on the synergy between the components.

4.1 Translation model training

The preprocessed translation model consistently outperformed the baseline model in both optimization behavior and translation quality. As shown in Figure 3, it converged earlier and achieved lower validation loss, indicating more stable learning dynamics.

Figure 3 illustrates the comparison of loss curves for baseline (raw) and preprocessed models. The preprocessed model achieves faster convergence and lower validation loss.

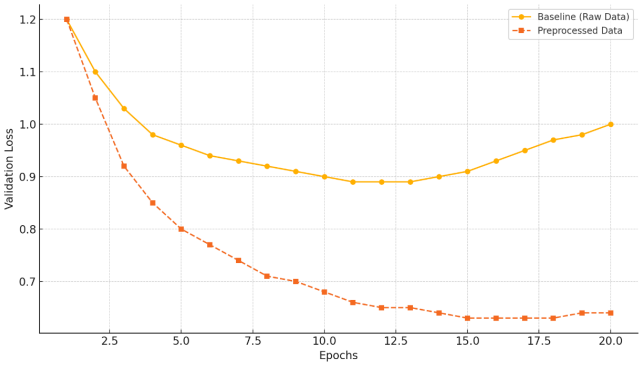

Figure 3: Training and Validation Loss over Epochs.

As depicted in Figure 3, the preprocessed model consistently exhibited more stable training behavior. It converged earlier (by epoch 11) and avoided the overfitting trend seen in the baseline model after epoch 15.

Qualitatively, translations generated by the preprocessed model were more fluent and grammatically consistent. Table 3 presents side-by-side comparisons of translation outputs with and without preprocessing.

Table 3: Translation Output Comparison

Source Sentence	Baseline Translation	Preprocessed Translation
She gave me the book that I wanted.	Ella me dio libro que quería.	Ella me dio el libro que yo quería.
The children are playing in the garden.	Los niños jugando jardín.	Los niños están jugando en el jardín.
I am not going to school today.	Yo no voy a colegio hoy.	No voy a la escuela hoy.

Figure 4 shows the preprocessed model's consistent improvements in BLEU and METEOR across all sentence length ranges.

Quantitatively, the results were equally encouraging:

- BLEU-4: increased from 0.548 to 0.621
- METEOR: improved from 0.468 to 0.509. These gains underscore the value of preprocessing, especially in morphologically rich languages like Spanish, where token normalization contributes significantly to alignment quality.

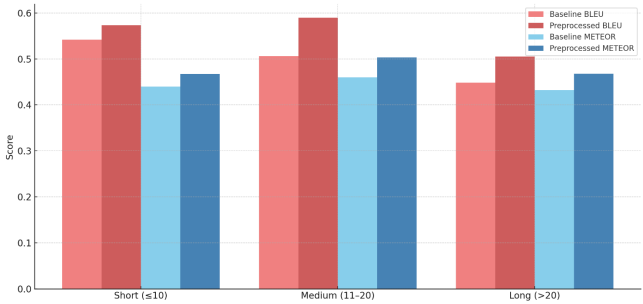

Figure 4: BLEU and METEOR Comparison between Raw and Preprocessed Models.

Although METEOR may in many settings align more closely with semantic adequacy than BLEU, the present study did not perform an independent checkpoint-selection comparison between the two metrics; BLEU was retained as the selection criterion for methodological consistency across the translation and quality-labeling stages.

Table 4: Metrics Comparison (Raw vs. Preprocessed)

Metric	Raw	Preprocessed
BLEU-4	0.548	0.621
METEOR	0.468	0.509
ROUGE-L	0.572	0.603

4.2 CNN quality classification training

The CNN-based quality estimator achieved strong and stable performance in both the binary and multi-class settings, indicating effective learning of sentence-level translation quality cues. As can be seen in Figure 5, the CNN model demonstrates high and stable accuracy across epochs, with minimal variance between training and validation.

The binary classification model (BLEU-labeled) achieved an accuracy of 88.7%, with an F1-score of 89.7%, suggesting a strong alignment between model predictions and ground truth quality labels. These results validate the model's capacity to learn from linguistic features without requiring handcrafted rules.

However, because the binary labels were derived from BLEU thresholds, these results should be interpreted as demonstrating successful learning of weakly supervised quality distinctions, not as a direct empirical comparison against BLEU-thresholding as a competing baseline.

Figure 6 highlights classification accuracy regarding true/false positives and negatives. The model maintained a low false negative rate, which is critical in avoiding mislabeling of poor-quality translations as acceptable.

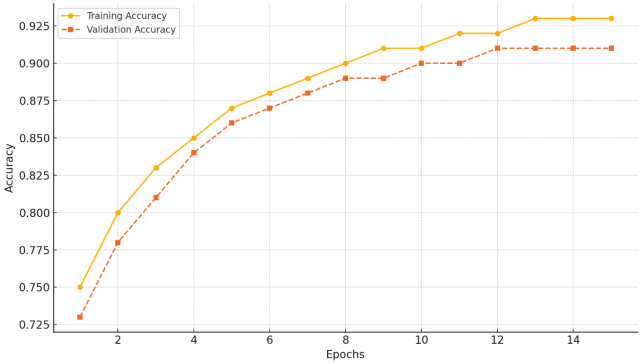

Figure 5: CNN Training and Validation Accuracy.

Figure 7 illustrates that the Multi-class CNN model correctly distinguishes translation quality levels, with the most substantial precision in extreme categories (poor and excellent).

For the multi-class model (human-labeled), performance remained robust:

- Macro F1-score: 0.82
- Precision for ”excellent” class: 0.89
- Precision for ”poor” class: 0.85

These results confirm that CNNs can effectively distinguish coarse- and fine-grained variations in translation quality, making them suitable for real-time feedback systems or human-in-the-loop review workflows (H. Chen et al., 2025; Kumar et al., 2024).

More specifically, these results indicate that the classifier learned sentence-level quality cues from BLEU-derived weak supervision rather than merely restating a fixed threshold at inference time. The practical value of this setup lies in transforming historical reference-based evaluations into a reusable screening model for subsequent quality assurance workflows.

Table 5: Performance Metrics (Binary and Multi-class)

Metric	Binary Classification	Multi-Class Classification
Accuracy	0.887	0.850
Precision	0.903	0.823
Recall	0.891	0.835
F1-Score	0.897	0.824

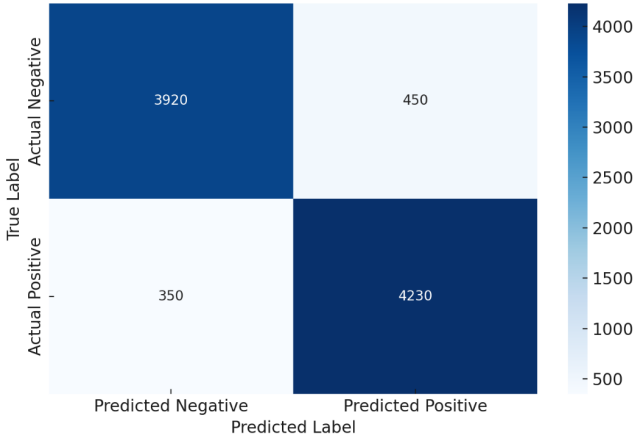

Figure 6: Confusion Matrix for CNN Quality Classification (Binary Labels)

4.3 Impact of preprocessing on translation output

Understanding the effect of preprocessing on the quality of machine-translated output is central to evaluating the utility of this study's proposed enhancements. While recent advancements in transformer-based neural machine translation have focused primarily on architectural improvements, our results reaffirm the often-underappreciated yet powerful role of preprocessing in sequence modeling tasks (Halkiopoulos et al., 2025).

Table 6 presents BLEU score changes by sentence-length category for the raw and preprocessed systems. Preprocessing yielded consistent gains across all three groups. The greatest improvement was achieved for medium-length sentences, followed by long sentences, whereas short sentences showed the smallest but still positive gain. This pattern indicates that preprocessing most strongly benefits sentences with moderate structural complexity, while remaining broadly useful across the full range of input lengths.

Table 6: BLEU Score Improvement by Sentence Length (Raw vs. Preprocessed)

Length Category	Raw BLEU	Preprocessed BLEU	Gain
Short (≤ 10)	0.542	0.573	+3.1
Medium (11–20)	0.506	0.590	+8.4
Long (> 20)	0.448	0.505	+5.7

To better illustrate the transformation, we provide several real translation examples in Table 7, comparing raw and preprocessed outputs. After preprocessing, examples show improved verb handling, article usage, and sentence flow.

Figure 7: Confusion Matrix for Multi-Class Classification.

Qualitative review by bilingual annotators confirmed that preprocessed outputs demonstrated improvements in fluency, idiomatic correctness, and syntactic alignment. In handling negation, compound tenses, and clause-level cohesion, specific gains were observed. These improvements stem from reduced lexical variance and enhanced sentence structure consistency due to the preprocessing pipeline.

Taken together, these results suggest that even in state-of-the-art NMT systems, simple yet linguistically aware preprocessing continues to play a critical role in model performance, particularly for sentence types involving syntactic and contextual dependencies.

Table 7: Sample Translations Before and After Preprocessing

Source Sentence	Baseline Translation (Raw)	Preprocessed Translation
She gave me the book that I wanted.	Ella me dio libro que quería.	Ella me dio el libro que yo quería.
The children are playing in the garden.	Los niños jugando jardín.	Los niños están jugando en el jardín.
I am not going to school today.	Yo no voy a colegio hoy.	No voy a la escuela hoy.
He wouldn't have said that under normal circumstances.	Él no habría dicho eso bajo circunstancias normales.	No lo habría dicho en circunstancias normales.
They saw a movie that was really good.	Ellos vieron película que fue muy buena.	Ellos vieron una película que fue muy buena.

4.4 Performance of CNN classifier

The CNN classifier was designed to serve as an automated evaluator of translated output, supporting traditional metric-based or human quality checks and, in deployment-oriented settings, acting as a fast pre-screening layer when direct reference-based evaluation is unavailable or impractical (Li et al., 2024). One of the key advantages of using a CNN in this context is its ability to detect localized linguistic patterns (e.g., word n-grams, syntax fragments) that correlate with quality indicators such as fluency, clarity, and semantic fidelity (Lu et al., 2019).

At the same time, we acknowledge that transformer-based quality estimators may capture longer-range semantic dependencies more effectively than CNN-based models. However, a direct empirical comparison was outside the scope of the present study, whose primary purpose was to assess whether a lightweight CNN could provide sufficiently strong sentence-level quality estimation within an efficient end-to-end pipeline. The results obtained here should therefore be interpreted as evidence of the effectiveness of a lightweight quality estimation design, rather than as a claim of superiority over transformer-based alternatives.

The binary classifier, trained on ~26,000 BLEU-labeled samples, demonstrated strong generalization capabilities. Performance metrics on the test set included an accuracy of 88.7%, a precision of 90.3%, a recall of 89.1%, and a macro F1-score of 89.7%. These results indicate that the model can reliably distinguish between high- and low-quality translations. As can be seen in Figure 8, AUC = 0.93, indicating excellent discriminatory power of the CNN model.

As illustrated by the representative examples in Table 2, these distinctions correspond not only to metric thresholds but also to observable differences in adequacy, grammatical completeness, and lexical alignment.

The ROC curve shown in Figure 8 reveals a high area under the curve (AUC), reflecting the model's capacity to effectively balance true and false positive rates. While this indicates strong discriminatory performance on the binary quality-labeling task, the present study did not include a direct head-to-head experimental comparison against a static BLEU-threshold baseline. Therefore, the results should be interpreted as evidence that the CNN can learn BLEU-derived quality distinctions effectively, rather than as proof of superiority over threshold-based evaluation schemes.

Importantly, the goal of the binary classifier was not to argue that BLEU should be discarded when reference translations are available. Rather, BLEU-derived labels were used to provide scalable weak supervision, allowing CNN to learn sentence-level quality cues that can later be reused in operational screening scenarios. In this sense, the classifier complements metric-based evaluation by translating historical reference-based judgments into a deployable predictive layer.

The multi-class CNN model also performed admirably when applied to a smaller human-annotated dataset of ~2,000 samples. It correctly classified translations into four categories—*poor*, *fair*, *good*, and *excellent*—with a macro F1-score of 0.82. Class-wise performance was strongest at the extremes (poor and excellent), where syntactic and semantic errors were more distinguishable. The classifier had more difficulty distinguishing between the adjacent classes *fair* and *good*, because these outputs often shared surface-level fluency, acceptable syntax, and substantial lexical overlap while differing in more subtle aspects such as idiomatic naturalness, collocational appropriateness, and completeness of semantic transfer (Table 8).

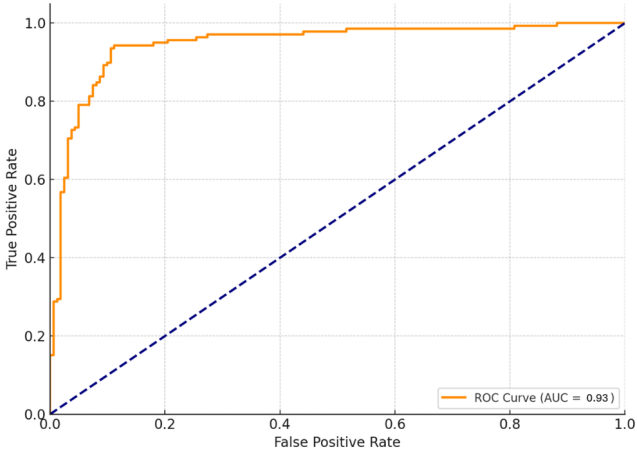

Figure 8: ROC Curve and AUC for Binary CNN Model.

A closer inspection of the misclassified boundary cases suggests that CNN was generally more reliable in detecting clearly poor translations and clearly strong translations than in resolving intermediate outputs. In particular, borderline errors often involved literal but grammatically well-formed renderings, acceptable local phrasing with weak idiomaticity, or translations that preserved the main propositional content but failed to reach fully natural target-language usage. These cases are difficult for CNN architecture because its local pattern detectors are well suited to capturing phrase-level fluency cues, but less suited to modeling deeper semantic appropriateness and discourse-sensitive naturalness.

Table 8: Class-Wise Performance on Human-Labeled Dataset

Class	Precision	Recall	F1-Score
Poor	0.85	0.82	0.84
Fair	0.74	0.77	0.75
Good	0.81	0.84	0.82
Excellent	0.89	0.87	0.88

In error analysis, the CNN occasionally assigned overly favorable labels to grammatically well-formed translations that remained weak in idiomatic adequacy. A representative example is the following:

Source:	“The plan fell through at the last minute.”
MT Output:	“El plan cayó al último minuto.”
Gold label:	“Fair”
CNN prediction:	higher-quality than gold label

Although the output is syntactically acceptable and preserves part of the source meaning, it reflects a literal rendering of the idiomatic expression “fell through”, resulting in a translation that is understandable but not fully natural in Spanish. This type of error helps explain why intermediate human-rated classes were harder to separate than the extreme classes in Table 8, the model was sensitive to local grammatical well-formedness, but less sensitive to idiomatic and contextual appropriateness.

4.5 Integrated pipeline effectiveness

The dual-stage pipeline developed in this study—consisting of (1) translation enhancement via preprocessing and (2) automatic quality classification via CNN—demonstrates a synergistic approach to improving and validating machine translation outputs.

As illustrated in Figure 9, input sentences first undergo preprocessing steps, including tokenization, lowercasing, lemmatization, and normalization. These refined inputs are then passed to the translation model, which generates Spanish outputs. Simultaneously, the CNN classifier processes these outputs to predict a quality label.

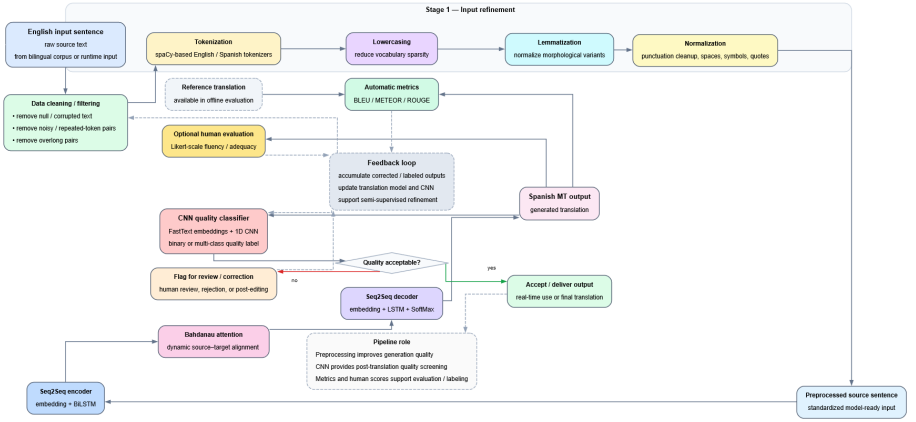

Figure 9: End-to-End Pipeline Flowchart.

This system offers significant advantages over traditional pipelines:

- Automated Quality Assurance: Translations predicted as low-quality can be routed to a human reviewer or automatically post-edited by a separate correction model. This significantly reduces human workload and improves review efficiency.
- Real-Time Feedback in MT Engines: Because of CNN’s lightweight architecture, it can be embedded into online systems to flag poor translations on the fly, thus enhancing user trust in real-time applications such as chatbots or customer service platforms.

- Adaptive Learning through Feedback: Labeled outputs from human intervention or the CNN itself can be periodically fed back into the model to support active learning, where the translation and classification models evolve over time to accommodate user preferences and domain-specific style.

Ultimately, this integrated pipeline ensures higher accuracy and fluency of translated content and establishes a framework for continuous translation quality improvement. Unlike one-time evaluation systems, the proposed design can grow and adapt, which is especially valuable in dynamic multilingual environments like legal, medical, and technical documentation workflows.

In sum, this study demonstrates that preprocessing and quality classification are not isolated auxiliary steps but strategically powerful components that, when unified, can raise the standard of machine translation quality and reliability in both academic and industrial settings.

5 Conclusion and future work

This study presented a comprehensive and practical framework for enhancing and assessing the quality of NMT using two complementary components: a preprocessing pipeline and a CNN-based translation quality classifier. By focusing on input refinement and output evaluation, our work bridges a critical gap in the machine translation lifecycle, where most research tends to emphasize only model architecture, leaving data quality and downstream validation underexplored.

This work's key contribution is demonstrating that simple, lightweight, linguistically informed preprocessing techniques—including tokenization, lowercasing, lemmatization, and punctuation normalization—can lead to statistically significant improvements in translation quality. The experimental results showed that preprocessing improved BLEU scores by up to 8.4 points and METEOR by over 4 points in medium-length sentences, where grammatical structure and context play a crucial role. These findings validate the hypothesis that data quality directly influences attention alignment and output fluency in Seq2Seq-based models.

Complementing this, our CNN classifier achieved high accuracy (88.7%) and F1-score (89.7%) in automatically distinguishing between high- and low-quality translations. This model leveraged fast, localized pattern recognition capabilities to assess syntactic and semantic quality at the sentence level. Its performance demonstrated that a lightweight CNN can effectively learn sentence-level quality distinctions from BLEU-derived supervision; however, no direct experimental comparison against static BLEU thresholding was conducted in the present study. Moreover, the CNN maintained robust generalization in the multi-class setting using human-labeled data, especially in accurately identifying poor and excellent outputs.

Together, these two components formed an integrated, feedback-driven pipeline that improved the translation process and validated the output dynamically. This pipeline is modular, lightweight, and practical enough to be embedded into real-time translation engines, post-editing workflows, and quality assurance systems.

Despite the promising outcomes, several limitations constrain the generalizability and scalability of the current framework.

First, the CNN classifier was trained using binary and four-class labels, with the binary setup relying primarily on BLEU-derived weak supervision and the multi-class setup relying on human

assessments. Accordingly, the binary model should be interpreted as a practical screening component learned from historical reference-based judgments rather than as a universal replacement for direct metric computation when gold references are available. Although this approach enables scalable training, it still inherits some of BLEU's coarse labeling assumptions.

Second, the translation model and classifier were trained on a single language pair (English-Spanish). Although this language pair is considered high-resource and typologically compatible, the performance gains observed here may not directly translate to low-resource or morphologically divergent languages such as Turkish, Finnish, or Hindi. Similarly, the preprocessing steps used (e.g., lemmatization) are highly language-dependent, and their adaptation to other scripts and syntactic structures may require additional tuning or rule customization. Equally important, all reported preprocessing gains were obtained with a single lightweight Seq2Seq model; with a stronger translation model the same preprocessing could yield little or no benefit, or might even degrade output, so these results should not be extrapolated beyond the simple model studied here.

Third, no direct transformer-based baseline was included in the quality-estimation stage. This was a deliberate scope decision, since the present work focused on testing a lightweight CNN-based evaluator within a computationally efficient pipeline rather than conducting a full architectural benchmark. While CNNs are efficient and interpretable, they do not capture long-range dependencies and global context as effectively as newer transformer-based classifiers. Consequently, certain semantic or discourse-level errors may remain less detectable in the current framework.

Finally, the scope of evaluation was limited to sentence-level quality on an open-domain English–Spanish corpus. Document-level translation quality, coherence, and consistency were not addressed, and the framework was not validated on domain-specific corpora such as legal, scientific, or medical text, where terminology control and multi-sentence context are especially important. In addition, the human-evaluation stage is not accompanied by a fully documented annotator profile or recruitment protocol, which may reduce the reproducibility of the Likert-scale assessment.

Future work should directly compare BLEU-based and METEOR-based checkpoint selection under the same training protocol to determine whether semantically oriented checkpointing yields more robust downstream translation quality and classification behavior.

Several further directions could extend this work. Larger and multilingual corpora such as the WMT shared-task data, OPUS, or Tatoeba would test whether the gains generalise across domains and language families, including low-resource pairs. The CNN evaluator could also be benchmarked against transformer-based estimators such as BERT, XLM-RoBERTa, or DeBERTa to quantify the trade-off between contextual modelling and computational cost. Finally, auxiliary linguistic features (part-of-speech tags, dependency parses, or alignment scores) could be added to improve robustness and interpretability, the framework could be extended from sentence-level to document-level quality estimation for discourse-sensitive settings, and a user-feedback loop could retrain the classifier in a semi-supervised or reinforcement-learning setting so that it adapts over time.

The dual-stage framework also has practical implications across several settings: in education it can give learners real-time feedback on translation exercises; for translation agencies and freelancers it can pre-filter low-quality machine output before human review, reducing editing effort; and in real-time communication tools such as multilingual chatbots and customer-support

platforms it can flag weak responses on the fly. In high-risk domains such as legal or medical translation, a classifier of this type could in principle act as a preliminary screen that flags outputs for mandatory human review, but only after dedicated validation on domain-specific corpora, since the present results do not support deployment in compliance-critical settings.

This study demonstrates that lightweight, linguistically informed preprocessing can substantially improve a simple Seq2Seq translation model, and that a CNN-based classifier can screen the resulting output efficiently. Whether these preprocessing gains carry over to stronger models remains an open question; what is shown here is that, for a lightweight model, pairing such preprocessing with a learned quality screen can improve the reliability and utility of the resulting system. By uniting data-level optimization with model-driven quality assessment, we offer a practical and extensible approach for improving both the process and the product of machine translation.

As machine translation continues to evolve, the future of quality assurance lies in such modular, intelligent, and adaptable pipelines—systems that generate and understand their own output, ensuring that translation becomes faster, more accurate, inclusive, and trustworthy.

References

Akter, M., Bansal, N., & Karmaker, S. K. (2022). Revisiting automatic evaluation of extractive summarization task: Can we do better than rouge? *Findings of the Association for Computational Linguistics: ACL 2022*, 1547–1560. https://doi.org/10.18653/v1/2022.findings-acl.122

Altinok, D. (2021). *Mastering spacy: An end-to-end practical guide to implementing nlp applications using the python ecosystem*. Packt Publishing Ltd.

Bahdanau, D., Cho, K., & Bengio, Y. (2016). Neural machine translation by jointly learning to align and translate. https://arxiv.org/abs/1409.0473

Bilal, M., Ali, G., Iqbal, M. W., Anwar, M., Malik, M. S. A., & Kadir, R. A. (2022). Auto-prep: Efficient and automated data preprocessing pipeline. *IEEE Access*, *10*, 107764–107784. https://doi.org/10.1109/ACCESS.2022.3198662

Brook, J. W. (2023). Towards improving neural machine translation systems for lower-resourced languages: Optimising preprocessing and data augmentation techniques for english to irish translation. https://doi.org/10.13140/RG.2.2.27649.26721

Calzolari, N., Bechet, F., Blache, P., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Isahara, H., Maegaard, B., Mariani, J. J., Mazo, H., Odijk, J., & Piperidis, S. (2022). *Language resources and evaluation conference lrec 2022 proceedings*. European Language Resources Association. https://hal.science/hal-04413343

Camacho-Collados, J., & Pilehvar, M. T. (2018). On the role of text preprocessing in neural network architectures: An evaluation study on text categorization and sentiment analysis. In T. Linzen, G. Chrupała, & A. Alishahi (Eds.), *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP* (pp. 40–46). Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5406

Castilho, S., Doherty, S., Gaspari, F., & Moorkens, J. (2018). Approaches to human and machine translation quality assessment. In *Translation quality assessment: From principles to practice* (pp. 9–38). Springer International Publishing. https://doi.org/10.1007/978-3-319-91241-7_2

Chai, C. P. (2023). Comparison of text preprocessing methods. *Natural Language Engineering*, *29*(3), 509–553. https://doi.org/10.1017/S1351324922000213

Chauhan, S., & Daniel, P. (2023). A comprehensive survey on various fully automatic machine translation evaluation metrics. *Neural Processing Letters*, *55*(9), 12663–12717. https://doi.org/10.1007/s11063-022-10835-4

Chen, H., Li, S., Fan, J., Duan, A., Yang, C., Navarro-Alarcon, D., & Zheng, P. (2025). Human-in-the-loop robot learning for smart manufacturing: A human-centric perspective. *IEEE Transactions on Automation Science and Engineering*, *22*, 11062–11086. https://doi.org/10.1109/TASE.2025.3528051

Chen, S., Liu, C., Haque, M., Song, Z., & Yang, W. (2022). Nmtsloth: Understanding and testing efficiency degradation of neural machine translation systems, 1148–1160. https://doi.org/10.1145/3540250.3549102

Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. *Expert Systems with Applications*, *72*, 221–230. https://doi.org/10.1016/j.eswa.2016.10.065

Chen, T. (2024). Design of translation error correction system based on improved seq2seq. *Procedia Computer Science*, *243*, 663–669. https://doi.org/10.1016/j.procs.2024.09.080

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. *Advances in Neural Information Processing Systems*, *28*. https://proceedings.neurips.cc/paper/2015/hash/1068c6e4c8051cfd4e9ea8072e3189e2-Abstract.html

Costa-jussà, M. R., Cross, J., Çelebi, O., Elbayad, M., Heafield, K., Heffernan, K., Kalbassi, E., Lam, J., Licht, D., Maillard, J., Sun, A., Wang, S., Wenzek, G., Youngblood, A., Akula, B., Barrault, L., Gonzalez, G. M., Hansanti, P., Hoffman, J., … Team, N. L. L. B. (2024). Scaling neural machine translation to 200 languages. *Nature*, *630*(8018), 841–846. https://doi.org/10.1038/s41586-024-07335-x

Danenas, P., & Skersys, T. (2022). Exploring natural language processing in model-to-model transformations. *IEEE Access*, *10*, 116942–116958. https://doi.org/10.1109/ACCESS.2022.3219455

Datta, G., Joshi, N., & Gupta, K. (2022). Analysis of automatic evaluation metric on low-resourced language: Bertscore vs bleu score. *Speech and Computer*, 155–162. https://doi.org/10.1007/978-3-031-20980-2_14

Deep, K., Kumar, A., & Goyal, V. (2021). Smt versus nmt: An experiment with punjabi–english. *Recent Innovations in Computing*, 63–71. https://doi.org/10.1007/978-981-15-8297-4_6

Domingo, M., García-Martínez, M., Helle, A., Casacuberta, F., & Herranz, M. (2023). How much does tokenization affect neural machine translation? In A. Gelbukh (Ed.), *Computational linguistics and intelligent text processing* (pp. 545–554). Springer Nature Switzerland.

Durrani, N., Dalvi, F., Sajjad, H., Belinkov, Y., & Nakov, P. (2019). One size does not fit all: Comparing NMT representations of different granularities. In J. Burstein, C. Doran, & T. Solorio (Eds.), *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)* (pp. 1504–1516). Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1154

Egger, R., & Gokce, E. (2022). Natural language processing (nlp): An introduction. In R. Egger (Ed.), *Applied data science in tourism: Interdisciplinary approaches, methodologies, and applications* (pp. 307–334). Springer International Publishing. https://doi.org/10.1007/978-3-030-88389-8_15

Eisenstein, J. (2019). *Introduction to natural language processing*. MIT Press.

Evtikhiev, M., Bogomolov, E., Sokolov, Y., & Bryksin, T. (2023). Out of the bleu: How should we assess quality of the code generation models? *Journal of Systems and Software*, *203*, 111741. https://doi.org/10.1016/j.jss.2023.111741

Falgueras, J., Lara, A. J., Fernández-Pozo, N., Cantón, F. R., Pérez-Trabado, G., & Claros, M. G. (2010). Seqtrim: A high-throughput pipeline for pre-processing any type of sequence read. *BMC Bioinformatics*, *11*(1), 38. https://doi.org/10.1186/1471-2105-11-38

Fan, C., Chen, M., Wang, X., Wang, J., & Huang, B. (2021). A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. *Frontiers in Energy Research*, *9*. https://doi.org/10.3389/fenrg.2021.652801

Fields, J., Chovanec, K., & Madiraju, P. (2024). A survey of text classification with transformers: How wide? how large? how long? how accurate? how expensive? how safe? *IEEE Access*, *12*, 6518–6531. https://doi.org/10.1109/ACCESS.2024.3349952

Freitag, M., Foster, G., Grangier, D., Ratnakar, V., Tan, Q., & Macherey, W. (2021). Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, *9*, 1460–1474. https://doi.org/10.1162/tacl_a_00437

Gao, Y., & Bu, S. (2024). Genetically optimized neural network for college english teaching evaluation method. *Education and Information Technologies*, *29*(17), 22371–22405. https://doi.org/10.1007/s10639-024-12736-6

Gao, Y. (2025). Deep learning-based strategies for evaluating and enhancing university teaching quality. *Computers and Education: Artificial Intelligence*, *8*, 100362. https://doi.org/10.1016/j.caeai.2025.100362

Halkiopoulos, C., Gkintoni, E., Aroutzidis, A., & Antonopoulou, H. (2025). Advances in neuroimaging and deep learning for emotion detection: A systematic review of cognitive neuroscience and algorithmic innovations. *Diagnostics*, *15*(4), 456. https://doi.org/10.3390/diagnostics15040456

Hameed, Z., & Garcia-Zapirain, B. (2020). Sentiment classification using a single-layered bilstm model. *IEEE Access*, *8*, 73992–74001. https://doi.org/10.1109/ACCESS.2020.2988550

Ibraheem, S., Altieri, N., & DeNero, J. (2017). Learning an interactive attention policy for neural machine translation. In S. Kurohashi & P. Fung (Eds.), *Proceedings of machine translation summit xvi: Research track* (pp. 108–115). https://aclanthology.org/2017.mtsummit-papers.9/

Islam, M. A., Anik, M. S. H., & Islam, A. B. M. A. A. (2021). Towards achieving a delicate blending between rule-based translator and neural machine translator. *Neural Computing and Applications*, *33*(18), 12141–12167. https://doi.org/10.1007/s00521-021-05895-x

Kaur, J., & Sohal, R. S. (2024). Noise estimation and removal in natural language processing. In *Handbook of vibroacoustics, noise and harshness* (pp. 1–25). Springer Nature. https://doi.org/10.1007/978-981-99-4638-9_38-1

Khalil, F., & Pipa, G. (2022). Transforming the generative pretrained transformer into augmented business text writer. *Journal of Big Data*, *9*(1), 112. https://doi.org/10.1186/s40537-022-00663-7

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

Kim, B.-K., Roh, J., Dong, S.-Y., & Lee, S.-Y. (2016). Hierarchical committee of deep convolutional neural networks for robust facial expression recognition. *Journal on Multimodal User Interfaces*, *10*(2), 173–189. https://doi.org/10.1007/s12193-015-0209-0

Kulkarni, A., & Shivananda, A. (2019). Advanced natural language processing. In *Natural language processing recipes: Unlocking text data with machine learning and deep learning using python* (pp. 97–128). Apress. https://doi.org/10.1007/978-1-4842-4267-4_4

Kumar, S., Datta, S., Singh, V., Datta, D., Kumar Singh, S., & Sharma, R. (2024). Applications, challenges, and future directions of human-in-the-loop learning. *IEEE Access*, *12*, 75735–75760. https://doi.org/10.1109/ACCESS.2024.3401547

Lavie, A., & Denkowski, M. J. (2009). The meteor metric for automatic evaluation of machine translation. *Machine Translation*, *23*(2), 105–115. https://doi.org/10.1007/s10590-009-9059-4

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

Lu, L., Yi, Y., Huang, F., Wang, K., & Wang, Q. (2019). Integrating local cnn and global cnn for script identification in natural scene images. *IEEE Access*, *7*, 52669–52679. https://doi.org/10.1109/ACCESS.2019.2911964

Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, *3*(1), 91–99. https://doi.org/10.1016/j.gltp.2022.04.020

Mashtalir, S. V., & Nikolenko, O. V. (2023). Data preprocessing and tokenization techniques for technical ukrainian texts. *Прикладні аспекти інформаційних технологій*, *6*(3), 318–326. https://doi.org/10.15276/aait.06.2023.22

Mayfield, E., & Black, A. W. (2020). Should you fine-tune bert for automated essay scoring? *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 151–162. https://doi.org/10.18653/v1/2020.bea-1.15

Mohamed, Y. A., Khanan, A., Bashir, M., Mohamed, A. H. H. M., Adiel, M. A. E., & 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

Naik, D., & Jaidhar, C. D. (2022). A novel multi-layer attention framework for visual description prediction using bidirectional lstm. *Journal of Big Data*, *9*(1), 104. https://doi.org/10.1186/s40537-022-00664-6

Papineni, K. (2002). Machine translation evaluation: N-grams to the rescue. In M. González Rodríguez & C. P. Suarez Araujo (Eds.), *Proceedings of the third international conference on language resources and evaluation (LREC'02)*. European Language Resources Association (ELRA). https://aclanthology.org/L02-1347/

Patil, R., Boit, S., Gudivada, V., & Nandigam, J. (2023). A survey of text representation and embedding techniques in nlp. *IEEE Access*, *11*, 36120–36146. https://doi.org/10.1109/ACCESS.2023.3266377

Raj, V. S. (2021). Seq2seq learning chatbot with attention mechanism. https://www.researchgate.net/publication/351837227_Performance_of_Seq2Seq_learning_Chatbot_with_Attention_layer_in_Encoder_decoder_model

Sai, A. B., Mohankumar, A. K., & Khapra, M. M. (2022). A survey of evaluation metrics used for nlg systems. *ACM Comput. Surv.*, *55*(2), 26:1–26:39. https://doi.org/10.1145/3485766

Sanni, B. (2021). Transfer learning in nlp: Designing scalable solutions to address low-resource language challenges in real-world applications. https://www.researchgate.net/publication/386887881_Transfer_Learning_in_NLP_Designing_Scalable_Solutions_to_Address_Low-Resource_Language_Challenges_in_Real-World_Applications

Sarkar, S., Das, S., Nath, B., & Mukhopadhyay, S. (2024). A multilingual neural machine translation model for low resource north eastern languages. https://doi.org/10.21203/rs.3.rs-4492445/v1

Scotti, V., Sbattella, L., & Tedesco, R. (2023). A primer on seq2seq models for generative chatbots. *ACM Comput. Surv.*, *56*(3). https://doi.org/10.1145/3604281

Shahin, N., & Ismail, L. (2024). From rule-based models to deep learning transformers architectures for natural language processing and sign language translation systems: Survey, taxonomy and performance evaluation. *Artificial Intelligence Review*, *57*(10), 271. https://doi.org/10.1007/s10462-024-10895-z

Spring, R., & Johnson, M. (2022). The possibility of improving automated calculation of measures of lexical richness for efl writing: A comparison of the lca, nltk and spacy tools. *System*, *106*, 102770. https://doi.org/10.1016/j.system.2022.102770

Stasimioti, M., Sosoni, V., Kermanidis, K., & Mouratidis, D. (2020). Machine translation quality: A comparative evaluation of SMT, NMT and tailored-NMT outputs. In A. Martins, H. Moniz, S. Fumega, B. Martins, F. Batista, L. Coheur, C. Parra, I. Trancoso, M. Turchi, A. Bisazza, J. Moorkens, A. Guerberof, M. Nurminen, L. Marg, & M. L. Forcada (Eds.), *Proceedings of the 22nd annual conference of the european association for machine translation* (pp. 441–450). European Association for Machine Translation. https://aclanthology.org/2020.eamt-1.47/

Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, 3104–3112.

Tamine, L., & Goeuriot, L. (2021). Semantic information retrieval on medical texts. *ACM Computing Surveys (CSUR)*. https://doi.org/10.1145/3462476

Tian, L., Su, S., Dong, X., Amann-Zalcenstein, D., Biben, C., Seidi, A., Hilton, D. J., Naik, S. H., & Ritchie, M. E. (2018). Scpipe: A flexible r/bioconductor preprocessing pipeline for single-cell rna-sequencing data. *PLOS Computational Biology*, *14*(8), e1006361. https://doi.org/10.1371/journal.pcbi.1006361

Umer, M., Imtiaz, Z., Ahmad, M., Nappi, M., Medaglia, C., Choi, G. S., & Mehmood, A. (2023). Impact of convolutional neural network and fasttext embedding on text classification. *Multimedia Tools and Applications*, *82*(4), 5569–5585. https://doi.org/10.1007/s11042-022-13459-x

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*. https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Wang, C., Zhang, J., & Chen, H. (2018). Semi-autoregressive neural machine translation. In E. Riloff, D. Chiang, J. Hockenmaier, & J. Tsujii (Eds.), *Proceedings of the 2018 conference*

on empirical methods in natural language processing (pp. 479–488). Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1044

Wang, H., Li, J., Wu, H., Hovy, E., & Sun, Y. (2023). Pre-trained language models and their applications. *Engineering*, *25*, 51–65. https://doi.org/10.1016/j.eng.2022.04.024

Wang, H., Wu, H., He, Z., Huang, L., & Church, K. W. (2022). Progress in machine translation. *Engineering*, *18*, 143–153. https://doi.org/10.1016/j.eng.2021.03.023

Zayyanu, Z. M. (2024). Revolutionising translation technology: A comparative study of variant transformer models - bert, gpt, and t5. *Computer Science & Engineering: An International Journal*, *14*(3), 15–27. https://doi.org/10.5121/cseij.2024.14302

Zhong, B., Xing, X., Love, P., Wang, X., & Luo, H. (2019). Convolutional neural network: Deep learning-based classification of building quality problems. *Advanced Engineering Informatics*, *40*, 46–57. https://doi.org/10.1016/j.aei.2019.02.009

Zhou, C., Qiu, C., Liang, L., & Acuna, D. E. (2025). Paraphrase identification with deep learning: A review of datasets and methods. *IEEE Access*, 1–1. https://doi.org/10.1109/ACCESS.2025.3556899

Correspondence

Zahra Moradi

Faculty of Foreign Languages and Literature
Islamic Azad University
Tehran, Iran
Zahramoradi.english@gmail.com