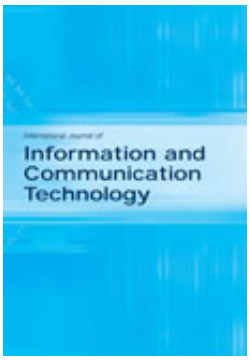
INDERSCIENCE PUBLISHERS
Linking academia, business and industry through research



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

Mask-embedded transformer for English text recognition and correction

Haiying Sang

DOI: 10.1504/IJICT.2025.10073376

Article History:

Received:	28 June 2025
Last revised:	23 July 2025
Accepted:	23 July 2025
Published online:	10 October 2025

Mask-embedded transformer for English text recognition and correction

Haiying Sang

School of Foreign Languages,
Heze University,
Heze – 274000, China
Email: lucky905896728@163.com

Abstract: As the digital age moves quickly, automatic recognition and correction of English text has become a significant job in the field of natural language processing (NLP). Most traditional ways of correcting text use simple statistical models and manual procedures, which do not work well with complicated grammatical, spelling, and semantic mistakes. This paper suggests an English text recognition and correction framework called MT-Tec, which is based on the improved transformer model and the masked embedding technique. MT-Tec can find and fix spelling mistakes, grammar mistakes, and vocabulary mistakes through multilevel context modelling and accurate error correction mechanisms. The MT-Tec framework works very well with many kinds of text errors and text qualities, and it is especially good at handling low-quality text. In general, the MT-Tec framework can be quite helpful for automatic proofreading, revising text, and learning a new language.

Keywords: English text recognition and correction; improved transformer; masked embedding; natural language processing; NLP.

Reference to this paper should be made as follows: Sang, H. (2025) 'Mask-embedded transformer for English text recognition and correction', *Int. J. Information and Communication Technology*, Vol. 26, No. 35, pp.1–17.

Biographical notes: Haiying Sang received her Master's degree from the Liaocheng University in 2018, and received her PhD from the De La Salle University-Dasmarinas in 2022. Currently, she works in Heze University. Her research interests include intelligent English teaching and teacher's professional development.

1 Introduction

With the advent of the digital age, English, as one of the most widely spoken languages in the world, is increasingly used on the Internet, social media, education, business and government. As the acceleration of globalisation, English not only dominates international communication, but also widely penetrates various digital platforms and information dissemination channels (Eke et al., 2023). Every day, there is an exponential growth in the number of English texts generated globally, which cover a wide range of content from academic research to daily communication, including news reports, social media comments, emails, online dialogues, and various technical documents. The large amount of textual data brings a wealth of information, but at the same time, the noise

problem is becoming more and more significant, especially in terms of spelling errors, grammatical irregularities, semantic ambiguities, and lack of clarity in context, which makes the task of automatic recognition and correction of English texts exceptionally complex and puts higher demands on NLP techniques.

With the continuous development of intelligent technology, machine learning (ML) and deep learning (DL) techniques have made significant progress in the field of natural language processing (NLP) (Sarker, 2021). In particular, the wide application of technologies such as deep neural network (DNN) and convolutional neural network (CNN) has greatly improved the capability of language processing tasks.

Although the transformer architecture has demonstrated excellent performance in NLP tasks such as machine translation and text generation, it still faces many technical bottlenecks in the application of text correction. The complexity of the text correction task is far more than just fixing spelling or grammatical errors, and the core of the task is that the model is required to have a deep understanding of the contextual semantics. Not only do we need to infer missing information based on the context and correct expression deviations, but we also need to ensure the grammatical normality and semantic coherence of the repaired text. Especially when facing texts with high noise density and serious loss of serious information, the error correction accuracy (CA) of the existing transformer model often drops significantly (Ganesh et al., 2021). The root of the problem is that although the architecture has strong modelling ability for long-range semantic dependencies, there are still inherent limitations in capturing fine features of local linguistic units (e.g., lexical collocations, microscopic deviations of syntactic structures) and logical deduction of missing information, which makes it difficult for the model to achieve accurate error localisation and semantic completions when dealing with low-quality texts. This technical bottleneck reflects the high demand of text correction task on the model's comprehensive language comprehension ability and points out the research direction of strengthening the local semantic modelling and reasoning mechanism for subsequent algorithm optimisation.

The masked embedding method not only helps the model understand the context better, but it also helps it fix mistakes in the text when it is processed. Masked embedding can make text restoration work much better, especially when the texts are loud or missing some information. It gives text correction chores new ideas and answers.

This study suggests a framework for recognising and fixing English text called MT-Tec. It is built on enhanced transformer architecture and the masked embedding technique. MT-Tec makes the model much better at correcting complex syntax and semantics by using multi-level context modelling and precise error correction methods which improves the standard transformer model by adding a masked embedding technique.

2 Relevant work

2.1 The transformer model

The pioneering work on transformer architecture, first proposed by Vaswani et al. in 2017, has fundamentally revolutionised the technological paradigm in the field of NLP, especially compared to traditional recurrent neural network (RNN) and long short-term memory network (LSTM). Traditional models parse input data through sequential

processing mechanisms and rely on temporal information to capture contextual associations. Although LSTM mitigates the gradient vanishing problem of RNN to some extent by gating unit design, and is able to deal with short-range semantic dependencies, both of them face the technical bottleneck of long-range dependency capture in long sequence modelling (Pandey and Kumar, 2025). In addition, the recursive computing mechanism leads to high computational complexity and limited efficiency of RNN and LSTM, making it difficult to realise parallel processing of large-scale data, a defect that is particularly significant in long text scenarios.

In contrast, the transformer model completely abandons recursive architecture and uses self-attention to parallelise the input sequence. More importantly, the parallel computing feature gives transformer a training efficiency far exceeding that of recursive models, which gives it a significant advantage in dealing with long text corpus and large-scale datasets, and this feature provides a key technical support for the engineering of NLP tasks.

The core architecture of the transformer model consists of a multi-head self-attention mechanism and position encoding, which together give the model the ability to capture sequence semantic dependencies and temporal information. In particular, the softmax function is used to normalise the inner product of the Query vector and the Key vector to create a weighting matrix. After that, the value vectors are weighted and added together using this matrix to get a weighted representation of each point. The core principle of the self-attention mechanism is to dynamically construct the relevance weights of each position in the input sequence through the interaction of three semantic representation vectors: query, key and value (Zheng et al., 2020). Its mathematical expression is:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where Q, K, and V are vectors of query, key, and value, and d_k is the size of the vector.

The transformer architecture is made up of many identical encoders and decoder layers stacked on top of each other. Each encoder layer has a multi-head self-attention mechanism and a feed-forward neural network. The decoder layer has multi-head self-attention, encoder-decoder attention, and a feed-forward neural network. By calculating several attention weights at the same time, the multi-head self-attention mechanism can capture multidimensional information about the input data from a number of different subspaces (Leng et al., 2021). This makes the model much more powerful. The decoder's job is to make each part of the target sequence based on how the encoder output looks. This highly parallelised and extensively stacked design lets the transformer model learn long-distance dependencies quickly, handle complicated syntactic and semantic information, and do very well on many NLP tasks.

The success of transformer is not just in machine translation, it also leads to the creation of additional pre-trained models like BERT, GPT, and others (Nassiri and Akhloufi, 2023). These later models are built on transformer's architecture, and after being pre-trained on a huge scale without supervision, they may be fine-tuned for a wide range of downstream jobs with outcomes that have never been seen before. Transformers are the most popular design in the NLP area right now because they are so flexible and efficient. It is also great for many kinds of language understanding and creation tasks.

2.2 Masked embedding

Masked embedding is a self-supervised learning method that has been used a lot in NLP and other fields for learning representations. The main idea is to add mask markers (like special symbols) to some sections of the input sequence and let the model figure out what these masked bits really mean based on what it already knows about the context. Masked embedding not only works with inputs that include missing information or noise but also helps the model grasp how things are related in context, which lets it reason and recover from incomplete data (Liu et al., 2021). This method is quite useful for processing text data, especially when there is not enough labelled data. It can also make the model better at expressing itself and generalising through self-supervision.

A typical application of masked embedding in NLP is the bidirectional transformer encoder representation (BERT) model, which forces the model to make full use of bi-directional contextual information (Shobana and Murali, 2023). BERT forces the model to make full use of bi-directional contextual information when predicting the masked words – this means that the model can infer the missing words with the help of both left and right contextual information at the same time, breaking the temporal constraints of unidirectional language models. BERT's bi-directional training method is better at understanding words with multiple meanings, complicated sentence structures, and changes in context than typical unidirectional language models. This training mechanism enables it to show significant advantages in multi-tasks such as text classification, machine translation, and question and answer systems, which fully proves the effectiveness of bidirectional context modelling.

The core advantage of the masked embedding approach is reflected in its self-supervised learning feature, where the paradigm does not need to rely on large-scale manually labelled data, but achieves representational learning by mining the intrinsic linguistic structure in unlabelled text. This data-efficient learning paradigm has important practical value in scenarios with high annotation costs such as medical texts and small language corpora (Ericsson et al., 2022). What is more noteworthy is that the masking strategy itself has a high degree of flexibility: the model can adjust the masking granularity according to the task requirements, from single words to phrases and even whole sentences, so that it can be adapted to different types of natural language understanding tasks. For example, in cross-language transfer learning, cross-language alignment of phrases by masking can effectively enhance the model's cross-linguistic semantic representation ability, and this adaptability makes masked embedding technology a key bridge between pre-training and downstream tasks. Researchers can freely configure the masks from micro word-level masks to macro sentence-segment-level masks according to the specific task requirements. For example, one can focus on phrase-level masking in a syntax-sensitive task, whereas a random word-level mask is more suitable for a semantic comprehension task (Strijkers et al., 2019). This customisability allows the model to be targeted to enhance different levels of language comprehension.

More notably, mask training gives the model excellent fault tolerance. Through repeated practice of reasoning in the presence of missing information, the model gradually masters the ability to 'see the smallest picture and know the big picture'. In practice, this ability translates into an amazing resilience to misspellings, dialectal variations, and even missing data.

From a more macro perspective, masked embedding represents a paradigm shift – from relying on external annotations to mining data for intrinsic associations. This line of thinking not only advances the field of NLP but also opens new avenues for cross-modal learning. Currently, pre-training methods based on the mask principle have been successfully extended to speech, image, and even video processing, and continue to set new records in various benchmark tests. Looking ahead, with the deepening of self-supervised learning theory, masking technology is expected to unleash its potential in a wider range of AI application scenarios.

2.3 English text recognition and correction

A lot of people are interested in English text recognition and correction, which is an important part of NLP. The rise of the information age, notably the rise of the Internet and social media, has greatly increased the pace and volume of text data creation. This makes the task of recognising and correcting text much more difficult. How to properly and quickly recognise and repair English text has become a key area of research in the field of NLP, especially for things like automatically generating content, verifying grammar, and fixing spelling mistakes.

Most of the time, traditional methods for recognising and correcting English text are based on matching rules and dictionaries (Azmi et al., 2019). These approaches use rules that people have written down to find spelling mistakes, grammar mistakes, or words that are used in the wrong context. For instance, spelling correction systems commonly employ edit distances (like Levenshtein distances) or n-gram models to figure out how similar two words are so that they can select the best candidates for correction. But these methods can usually only fix simple spelling mistakes and a few grammatical structures, which makes them less useful in real-world situations.

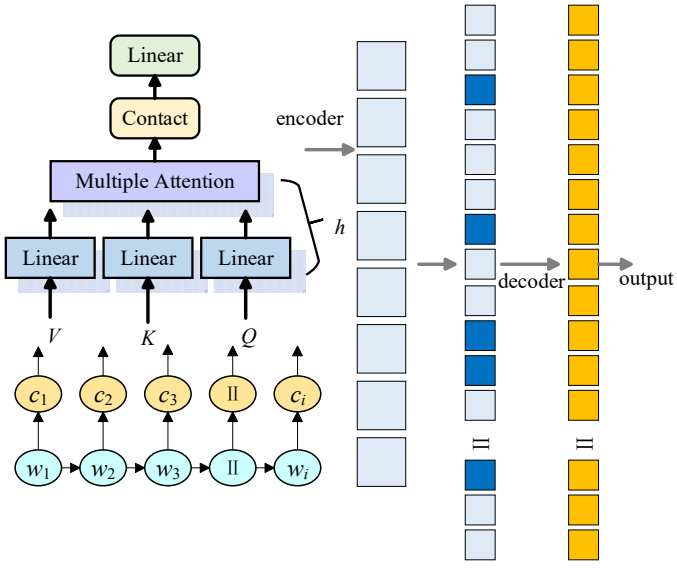
With the rise of statistical learning methods, ML-driven text correction techniques have gradually replaced traditional regularisation methods. Early statistical methods relied heavily on feature engineering, usually by training classifiers to determine whether words are spelled correctly or sentence structures conform to grammatical rules. Although these statistical learning methods are capable of handling simpler types of errors, they tend to rely on hand-designed feature sets that struggle to adequately capture complex linguistic phenomena (Baltrušaitis et al., 2018). Simple Bayes, support vector machines (SVM), and other common algorithms use a feature space to find and fix errors. Choosing and designing features is very important in this method, and how to get useful feature information from a lot of text input is a big aspect that affects speed. Specifically, hand-constructed features often fail to cover subtle semantic associations and syntactic variants in natural language, resulting in models that are limited in their ability to generalise in the face of diverse textual scenarios.

Generative models replace erroneous parts by generating new text instead of directly selecting a most likely correction. In recent years, the successful application of sequence to sequence (Seq2Seq) modelling, especially in machine translation tasks, has provided new ideas for English text correction. Seq2Seq modelling maps the input erroneous text to a high-dimensional space through an encoder and decoder architecture, and then generates a new, correct output text. Unlike traditional classification methods, the generative model can handle more complex errors and generate more fluent and natural text through context.

Currently, many English text recognition and correction systems have begun to combine a variety of technological tools and adopt an integrated approach to improve performance. For example, some systems use ML algorithms to detect errors by combining rule-based spell-checking with ML models and use rule engines to provide correction suggestions (Gondaliya et al., 2022). Other approaches combine techniques such as syntactic analysis and syntactic tree generation to improve the accuracy of corrections through more fine-grained analyses of language structure.

In short, the current state of research on recognising and correcting English text shows that there are many methods that work well in different situations. However, the focus of current and future research is still on how to make the system work better with complex text, long text, and multi-domain applications.

Figure 1 Framework of MT-Tec (see online version for colours)



3 Methodology

3.1 English text recognition and correction framework

The MT-Tec framework works well because many modules work together to quickly find and fix mistakes in English text. In particular, the tasks of each module in the framework help each other, and together they make the model better at finding and fixing spelling and grammar mistakes. See Figure 1. Next, we'll go into great depth on how each module works and how to use it.

3.1.1 Input text recognition module

Firstly, the text is subjected to a segmentation process that breaks the text into words or sub-word units, which is the basis of NLP. Subsequently, irrelevant symbols, punctuation marks, and stop words are removed from the text to ensure that the model only processes

task-relevant information. Next, the text is word-embedded, mapping each word into a fixed-dimension variable representation, which provides the basis for subsequent contextual modelling.

Based on this, the module also performs preliminary error recognition. With the pre-trained language model, the module analyses the text for potential spelling errors, grammatical problems, and contextual inconsistencies, providing feedback for the subsequent deep correction module.

Specifically, each word in the input is mapped to a low-dimensional variable through a predefined word embedding matrix, which is fixedly learnt during the training process, and can effectively capture the semantic and contextual information of the word (Camacho-Collados and Pilehvar, 2018). The formula is expressed as:

$$e_w = f(w) \tag{2}$$

where w is a word in the vocabulary and e_w is a fixed word embedding variable for that word. This word embedding will be used as model input to the subsequent context modelling phase, providing the basis for subsequent syntactic and semantic correction tasks.

The module then finds and marks possible mistakes in the input text depending on the information around it. It can find errors by using the following formula to figure out how similar each word is to its context:

$$Error\ detection(w_i) = Similarity(w_i, c_i) \tag{3}$$

where *error detection*(w_i) is the outcome of finding an error in the i^{th} word w_i; c_i is the context information for the word, and *similarity*(w_i, c_i) is the measure of how similar the word is to its context. If the similarity is below a particular level, the term is classified as possibly having a mistake.

This stage not only finishes the text recognition module's preparation and formatting of the text, but it also lets it find possible mistakes in the text ahead of time, which is the first step in the repair work.

3.1.2 Multi-level context modelling module

This module is the most important aspect of the MT-Tec framework which is mostly in charge of deeply modelling the contextual information in the input text. It uses the enhanced transformer architecture's powerful self-attention mechanism to completely capture long-range dependencies and complicated syntactic semantic information in the text. The optimised multi-scale self-attention mechanism is computed by the following formula:

$$Multi\text{-}scale\ attention(Q, K, V) = \sum_{i=1}^{N} softmax\left(\frac{QK_i^T}{\sqrt{d_k}}\right)V_i \tag{4}$$

where K_i and V_i represent the key matrix and value matrix under the i^{th} attention scale, respectively, and N denotes the number of preset attention scales. This mechanism enables the model to synchronously parse the local semantic associations and global contextual structure of the text at multiple semantic levels: capturing the subtle deviations of lexical collocations at the micro level and constructing the semantic dependency

network across paragraphs at the macro level, which significantly strengthens the depth and accuracy of the contextual modelling. For example, when dealing with complex sentences containing long-distance references, the multi-scale mechanism can simultaneously utilise local syntactic structures and chapter-level semantic cues to effectively improve the model's ability to detect covert language errors.

Compared with the traditional transformer architecture, MT-Tec further optimises the efficiency of attention computation through a dynamic weight adaptation mechanism. Each attention head in the framework can automatically adjust the weight allocation strategy based on the real-time features of the text input (e.g., error density, semantic complexity, etc.), so that the model can dynamically focus on key information nodes in different contexts (Kumar, 2022). Specifically, the dynamic adjustment mechanism is realised by the following formula:

$$\alpha_i = \frac{\exp(\alpha_i)}{\sum_{i=1}^{n} \exp(\alpha_i)} \tag{5}$$

where a_i denotes the adaptive weight of the i^{th} attention head and n is the number of attention heads.

To enhance the stability and training efficiency of the model, MT-Tec also introduces residual linking and layer normalisation techniques. This ensures that the output of each layer is delivered stably and avoids the problem of vanishing gradients. The residual connection is calculated by the formula:

$$y_i = x_i + LayerNorm\left(Attention\left(x_i\right)\right) \tag{6}$$

where x_i is the input for layer i and y_i is the output after connecting the residuals.

With these improvements, the MT-Tec framework can provide higher precision syntactic and semantic corrections, which also enhances the comprehensiveness and accuracy of text understanding.

3.1.3 Mask embedding module

The multi-level context modelling module has an improved transformer architecture that lets the model capture long-range dependencies and complex contextual information. However, it needs more reasoning power to fix mistakes in text correction tasks quickly, especially when it has to deal with noisy or missing contextual information. This necessity drives us to add the masked embedding module as the second most important part of the MT-Tec framework.

Compared with the limitation that traditional spelling correction methods can only deal with superficial spelling errors, this module has a deeper semantic comprehension capability, which can not only accurately locate spelling deviations, but also derive semantic missing units based on grammatical rules and contextual associations in contextually ambiguous or missing information scenarios, thus realising the leap from superficial form repair to deep semantic completion. This module works closely with the multi-level context modelling module, which gives the model a lot of useful contextual information, and the masked embedding module, which helps the model use this information to figure out what sections are missing while it is working with partial text.

In the model training phase, the masked embedding module uses a cross-entropy loss function to optimise the prediction performance. The mechanism takes the mask position as the training anchor point and constructs a gradient back-propagation path by calculating the degree of difference between the model's predicted output and the real semantic labels. The cross-entropy loss function can be expressed as:

$$L_{mask} = -\sum_{i=1}^{N} y_i \log\left(\hat{y}_i\right) \tag{7}$$

where y_i is the true label of the mask location, $\hat{y}_i$ is the model's prediction of the mask location, and N is the number of mask locations.

In addition, the masked embedding module introduces an adaptive embedding generation mechanism to dynamically adjust the embedding representation for each masked location (Li et al., 2022). This embedding generation process can be represented as:

$$e_w^{masked} = Embedding(w) + \alpha \cdot Context(w) \tag{8}$$

where e_w^{masked} is the embedding representation of the masked embedding position, w is the word to be embedded, α is a learnable weighting factor, and $context(w)$ is an adjustment variable generated based on the context.

3.1.4 Error detection and correction module

The purpose of correcting the model is to make the corrected text as close to the correct text as possible. The quality of the correction is measured by mean squared error (MSE):

$$L_{correct} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{9}$$

where y_i is the real label of the corrected text, $\hat{y}_i$ is the corrected text that the model made, and N is the number of words in the text. By minimising this loss function, MT-Tec can correctly change the model to provide the best results from text repair.

MT-Tec further improves the corrective effect during the error correction process by adjusting the weight of the context. In particular, the model changes the correction method based on how relevant each word is to the context, which lets the correction of each word be optimised based on how important it is to the phrase (Bryant et al., 2023). To do this, we figure out the context relevance score for each word and change the correction technique based on that score:

$$w_i^{context} = \frac{\exp\left(u_i^{\top} c_i\right)}{\sum_{i=1}^{N}\exp\left(u_i^{\top} c_i\right)} \tag{10}$$

where u_i is the embedding variable of the input term w_i, c_i is the context variable associated to the term, $u_i^{\top} c_i$ is the similarity between the term and its context, and $w_i^{context}$ is the weight of the term in the present context.

3.1.5 Output generation module

The output generation module of the MT-Tec framework, as the final execution unit of the whole system, undertakes the key task of transforming the analysis results of the prelude module into high-quality corrected text.

In the process of text correction, the MT-Tec framework adopts Seq2Seq, which is a generative model based on the encoder decoder architecture. During the generation process, the model will gradually generate revised text based on the current contextual information and previously generated text.

To further optimise the generation process, the MT-Tec framework adopts beam search. Bundle search ensures higher quality corrected text by selecting the candidate with the highest probability from multiple candidate sequences, avoiding errors such as grammatical errors or semantic inconsistencies (Ji et al., 2023). The goal of beam search is to maximise the probability of generating text, as follows:

$$P_{output} = \prod_{t=1}^{T} P\left(w_t \left| w_1, w_2, \ldots, w_{t-1}\right.\right) \tag{11}$$

where P_{output} denotes the overall probability of generating the text, P is the conditional probability of generating the t^{th} vocabulary, and T is the length of the text.

3.2 Data collection and pre-processing

In this study, the dataset used was Lang-8 Learner Corpus, which consisted of English texts provided by non-native English speakers and annotated for spelling, grammar, and vocabulary errors. The Lang-8 dataset provides rich annotated data for English error recognition and correction tasks, suitable for training and evaluating the MT-Tec framework proposed in this paper. Its information is shown in Table 1.

Table 1 Lang-8 Learner Corpus Dataset overview

Dataset name	*Lang-8 Learner Corpus*
Data source	Lang-8 website, English texts from non-native speakers
Language	English
Error types	Spelling errors, grammatical errors, vocabulary errors, collocation errors
Annotation content	Error types (spelling, grammar, vocabulary, etc.), corrections
Data format	Text pairs with original text and annotated errors and corrections
Use case	English text error recognition and correction, grammar fixing, spelling correction, etc.
Applicable tasks	Automatic error recognition and correction for non-native English learner texts

Before using the Lang-8 dataset with the MT-Tec framework, the data was pre-processed to make sure it would fit the model inputs' formatting needs. The initial stage in pre-processing was to clean up the text by getting rid of unnecessary symbols, punctuation marks, special characters, and words that were not used. This step makes sure that the model only works with useful information that is related to the task and does not get confused by noise. The cleaned text is shorter, which makes it easier to analyse and process later.

The next stage, text segmentation, breaks the cleaned text up into words or smaller units of words so that grammar modelling and error recognition may happen. Based on this, the error annotation extraction gets the mistakes and their remedies from each text in the dataset to make sure that the annotation information is comprehensive. Lastly, a pre-trained word embedding model is used to make a low-dimensional vector representation for each word. This is the basis for the next step, which is context modelling. The data is now ready to be put into the MT-Tec framework for finding and fixing errors after these phases of pre-processing.

4 Experiments and results

4.1 Experimental setup

This study sets up a configuration with several important parameters that directly affect the training effect and performance of the model to test how well the proposed MT-Tec framework works for recognising and correcting English text.

The learning rate was set to 0.001 and Adam optimiser was employed during training. The tests used a batch size of 32 and 50 training rounds to make sure that the model converged and the training was stable (Ma et al., 2021). The same computer hardware was utilised to train all the models. For example, GPU-accelerated training was done on a machine with an NVIDIA GTX 1080 Ti graphics card.

We also split the dataset into three parts: a training set (70%), a validation set (15%), and a test set (15%). This is a popular way to approach it. The pre-processing stage included dividing words, deactivating words, and embedding words in all texts to make sure that the input data was in the right format for the model.

This experiment uses the dropout method with a dropout rate of 0.3 to make the model work better and avoid overfitting. Also, after each training session, the validation set checks how well the model works and changes the model parameters based on the validation findings which make sure that the final trained model can generalise better.

4.2 Experimental indicators

To fully assess how well the MT-Tec system worked for recognising and fixing English text, we picked a number of measures. The three key experimental measures used to judge how well the model works in different areas are as follows.

- *Error detection rate (EDR):* EDR is the ratio of the number of mistakes the model properly finds in text to the actual number of errors in the text. It shows how well the model can find errors in text (Gong et al., 2021). A greater EDR means that the model can correctly find mistakes in the text. The formula is:

$$EDR = \frac{T_{detected}}{T_{total}} \times 100 \tag{12}$$

  where $T_{detected}$ stands for the number of errors that the model successfully found, and T_{total} stands for the total number of errors that are really in the text.

- *CA:* CA is a way to find out how accurate the model is at fixing the faults it finds (Kim and Katipamula, 2018). The formula is:

$$CA = \frac{T_{corrected}}{T_{detected}} \times 100 \tag{13}$$

where $T_{corrected}$ is the number of mistakes the model was able to fix, and $T_{detected}$ is the number of mistakes the model was able to find.

F1-score is a combination of precision and recall that is good for judging how well text recognition and correction models work overall (Chicco and Jurman, 2020). The precision rate tells you how accurate the model is at fixing mistakes, while the recall rate tells you how many mistakes the model rectified out of all the mistakes. The formula is:

$$F1 = 2 \times \frac{precision \times recall}{precision + recall} \tag{14}$$

These metrics enable a comprehensive assessment of the MT-Tec framework's performance in error identification and correction, helping to analyse the model's strengths and weaknesses in each area.

4.3 Performance evaluation in multi-type error recognition and correction

Experiment 1 systematically evaluates the performance of the MT-Tec framework for text error processing which focuses on the framework's ability to recognise and correct three types of common textual errors: spelling errors, grammatical errors and lexical errors.

The experiments adopt an end-to-end evaluation methodology to quantitatively analyse the framework performance through three core metrics, namely EDR, CA and F1 value. Of note, this study is the first to differentially evaluate the performance of the MT-Tec framework on different error types. During the testing process, the framework needs to simultaneously handle character-level errors at the spelling level, structural errors at the grammar level, and semantic errors at the lexical level, and this multi-task evaluation better reflects the text processing needs in real scenarios. Then, the dataset is divided into training, validation and test sets, with the training set accounting for 70% of the dataset, the validation set for 15% and the test set for 15%.

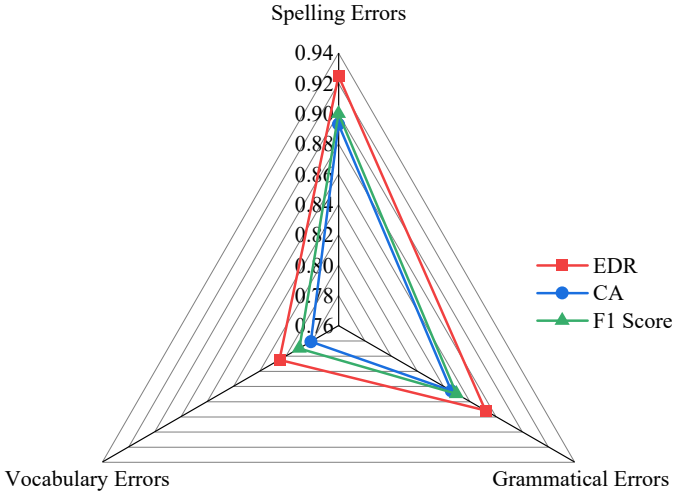
The experimental results are shown in Figure 2.

The EDR of MT-Tec is as high as 92.5%, the CA for spelling errors reaches 89.3%, a result that not only confirms the model's detection ability but also highlights its accuracy at the level of error repair. In addition, the F1-score of 0.90 indicates that the framework achieves a good balance between precision and recall, which further supports its dominant position in the spelling error correction task.

Compared with spelling errors, grammatical errors are more difficult to handle, and the EDR of the MT-Tec framework is 87.2%, which is slightly lower than that of the spelling error detection index but still maintains a high level. The CA of the model is 84.6%, indicating that the framework can relatively accurately fix syntax errors after identifying them. Although grammar errors are usually more complex than spelling errors, the MT-Tec framework still demonstrates a relatively ideal ability to correct them,

with an F1-score of 0.85, indicating that the framework can provide stable performance in the task of identifying and correcting grammar errors.

Figure 2 Performance evaluation of MT-Tec in error recognition and correction (see online version for colours)



When dealing with vocabulary errors, the EDR value of the MT-Tec framework is 80.5%, which is slightly lower than the recognition ability of spelling and grammar errors, indicating that the recognition of vocabulary errors is relatively more challenging. CA is 78.1%, indicating that the model's performance in vocabulary error correction is slightly inferior to other types of errors. Despite this, the F1-score is 0.79, indicating that the MT-Tec framework still has a certain ability in repairing vocabulary errors, although there is a slight gap in performance compared to spelling and grammar errors.

For grammatical and lexical errors, the framework shows stable performance despite the language complexity constraints. This finding provides empirical support for the application of MT-Tec in the field of automatic English text correction, especially in the scenarios of text correction systems and intelligent writing aids in the field of education, where the framework's ability to efficiently handle common spelling and grammatical errors is of great practical value. Future research can further optimise the lexical semantic understanding module to improve the model's processing accuracy for complex linguistic errors.

4.4 Robustness test with different text complexity

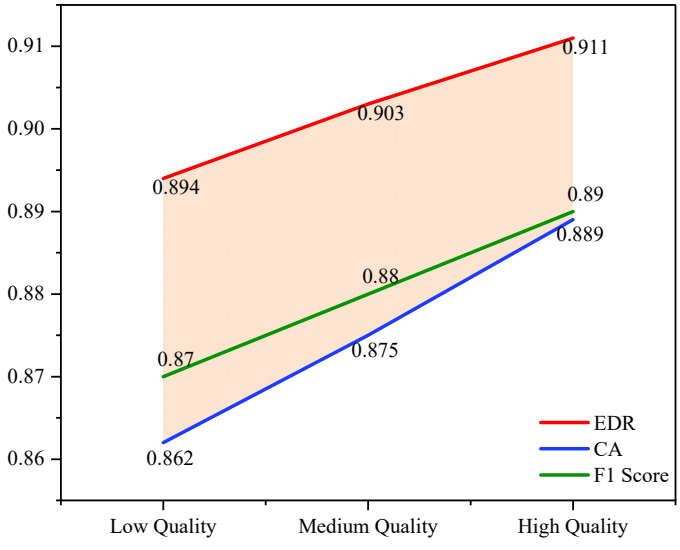
This study aims to systematically evaluate the robust performance of the MT-Tec framework under multi-dimensional text complexity, focusing on the differences in its performance in low-, medium- and high-quality text scenarios. Considering that the quality of texts in real corpora generally shows significant variability, such as the fluctuation of error density in written texts and the normative differences in professional literature, this experiment comprehensively tests the framework's adaptability in different error-loaded environments by constructing a multi-gradient text quality system,

especially focusing on the stability of the framework in detecting and correcting errors in high-frequency error scenarios.

The experiment adopts Lang-8 Learner Corpus as the base dataset, which covers real texts of non-native English learners and is rich in error type annotations. The research team classified the texts into low, medium and high quality based on the dimensions of error frequency (e.g., the number of errors per 100 words) and error complexity (the proportion of syntactic errors, the degree of semantic deviation, etc.). Low-quality texts have a lot of spelling and grammar mistakes and sentences that don't follow the rules; medium-quality texts have some spelling or grammar mistakes, but the overall structure is clearer; and high-quality texts are grammatically correct, spelled correctly, and almost error-free.

Next, the dataset is divided into training, validation and testing sets, and the three types of texts are trained and tested respectively using the MT-Tec framework. The experiments focus on EDR, CA and F1-score and compare the performance differences of the framework in the three types of texts. The adaptability of the framework between low-quality and high-quality texts is further analysed by evaluating the model's recognition and correction ability on different quality texts. The experimental results are shown in Figure 3.

Figure 3 Performance evaluation of MT-Tec in different text quality levels (see online version for colours)



The experimental results reveal the differentiated performance of the MT-Tec framework in different quality text processing. In the low-quality text scenario, the EDR of the framework reaches 89.4%, which fully demonstrates its ability to identify complex spelling and grammatical errors; and the CA of 86.2% further confirms the accuracy of the framework in dealing with high-density errors. In addition, the F1-score of 0.87 indicates that MT-Tec is able to maintain a stable overall performance in low-quality text, effectively handling the challenge of a dense distribution of error types.

As the text quality increases to medium level, the performance of MT-Tec framework shows a significant optimisation trend: the EDR and CA climb to 90.3% and 87.5%

respectively, reflecting that the model is able to complete the task of error identification and correction with higher accuracy under the condition of lower error density. At this time, the F1-score reaches 0.88, which not only verifies the stability of the framework in medium quality text, but also indicates that its repair capability has been further strengthened.

When processing high-quality text, the MT-Tec framework shows even better performance: the EDR increases to 91.1%, which is the highest among the three types of text, reflecting the framework's ability to capture subtle errors; the CA of 88.9% proves that the framework still maintains a high level of fixing accuracy in low error density scenarios; and the F1-score of 0.89 confirms that the framework can not only achieve detailed error recognition and correction in high-quality text, but also improve the accuracy of fixing errors in the middle-quality text. The 0.89 F1-score further confirms that the framework is not only able to achieve detailed error correction in high-quality text processing, but also able to optimise text quality in depth.

In summary, the performance of MT-Tec framework in different text quality shows a gradual improvement, from effectively coping with complex errors in low-quality text to achieving fine-grained correction in high-quality text, which fully demonstrates its robustness and adaptability in multi-scenario applications. These experimental data provide strong empirical support for the wide deployment of the MT-Tec framework in real language processing tasks.

5 Conclusions and discussion

This paper provides the MT-Tec framework for recognising and fixing English text based on the enhanced transformer model with the masked embedding technique. The MT-Tec framework does a great job of finding and fixing spelling, grammar, and vocabulary mistakes, and it does far better than traditional methods. The framework is very stable when working with texts of varying levels of complexity, and it can handle faults in both noisy and high-quality texts, showing that it is very robust.

However, although the MT-Tec framework has shown good performance in many tests, there is still room for improvement in its practical application and algorithm optimisation. In terms of error types, the current framework mainly focuses on identifying and correcting spelling errors, grammatical errors and lexical errors, which are common in regular texts, but the framework's ability is still insufficient in the face of higher-order linguistic problems such as semantic ambiguity and discourse logic faults. Subsequent research can further expand the coverage dimension of error types, especially for complex errors such as terminology misuse and logical connection errors, which are unique to professional texts, to improve the adaptability of the framework in multi-scenario text correction.

In terms of computational efficiency and processing time, the MT-Tec framework relies on the complex transformer architecture to achieve high-precision text processing. To break through this technical bottleneck, research can explore model lightweighting strategies, such as model pruning, parameter quantisation and knowledge distillation, to reduce the model computational overhead under the premise of guaranteeing CA, to enhance the framework's practicability in real-time text processing scenarios.

In addition, the performance of the framework is strongly dependent on large-scale annotated datasets such as Lang-8 Learner Corpus, which may limit its application scope in low-resource scenarios (e.g., small-language texts, professional domain corpora). Future research can try to introduce migration learning and semi-supervised learning mechanisms to enhance the learning ability of the framework in data-scarce environments through efficient feature extraction and knowledge migration from small-scale labelled data and then expand its application boundaries in multi-domain text correction tasks. These limitations not only provide a clear direction for the iterative optimisation of the MT-Tec framework but also provide a reference point for the development of automatic text correction in the field of NLP.

Acknowledgements

This work is supported by the Heze University Doctoral Fund Project (No. XY22BS61) and the Social Science Project of Heze City (No. 2023ZC96).

Declarations

All authors declare that they have no conflicts of interest.

References

Azmi, A.M., Almutery, M.N. and Aboalsamh, H.A. (2019) 'Real-word errors in Arabic texts: a better algorithm for detection and correction', *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 27, No. 8, pp.1308–1320.

Baltrušaitis, T., Ahuja, C. and Morency, L-P. (2018) 'Multimodal machine learning: a survey and taxonomy', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 41, No. 2, pp.423–443.

Bryant, C., Yuan, Z., Qorib, M.R., Cao, H., Ng, H.T. and Briscoe, T. (2023) 'Grammatical error correction: a survey of the state of the art', *Computational Linguistics*, Vol. 49, No. 3, pp.643–701.

Camacho-Collados, J. and Pilehvar, M.T. (2018) 'From word to sense embeddings: a survey on vector representations of meaning', *Journal of Artificial Intelligence Research*, Vol. 63, pp.743–788.

Chicco, D. and Jurman, G. (2020) 'The advantages of the Matthews correlation coefficient (MCC) over F1-score and accuracy in binary classification evaluation', *BMC Genomics*, Vol. 21, pp.1–13.

Eke, C., Yibowei, N., Bufumoh, A., Ndionyenma, J.P. and George, N.N. (2023) 'New media dynamics in Nigeria: a survey of globalisation influence on communication channels', *Research Journal of Mass Communication and Information Technology*, Vol. 9, No. 4, pp.46–58.

Ericsson, L., Gouk, H., Loy, C.C. and Hospedales, T.M. (2022) 'Self-supervised representation learning: introduction, advances, and challenges', *IEEE Signal Processing Magazine*, Vol. 39, No. 3, pp.42–62.

Ganesh, P., Chen, Y., Lou, X., Khan, M.A., Yang, Y., Sajjad, H., Nakov, P., Chen, D. and Winslett, M. (2021) 'Compressing large-scale transformer-based models: a case study on BERT', *Transactions of the Association for Computational Linguistics*, Vol. 9, pp.1061–1080.

Gondaliya, Y., Kalariya, P., Panchal, B.Y. and Nayak, A. (2022) 'A rule-based grammar and spell checking', *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, Vol. 14, No. 1, pp.48–54.

Gong, X., Lu, J., Zhou, Y., Qiu, H. and He, R. (2021) 'Model uncertainty based annotation error fixing for web attack detection', *Journal of Signal Processing Systems*, Vol. 93, pp.187–199.

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y.J., Madotto, A. and Fung, P. (2023) 'Survey of hallucination in natural language generation', *ACM Computing Surveys*, Vol. 55, No. 12, pp.1–38.

Kim, W. and Katipamula, S. (2018) 'A review of fault detection and diagnostics methods for building systems', *Science and Technology for the Built Environment*, Vol. 24, No. 1, pp.3–21.

Kumar, A. (2022) 'Contextual semantics using hierarchical attention network for sentiment classification in social internet-of-things', *Multimedia Tools and Applications*, Vol. 81, No. 26, pp.36967–36982.

Leng, X-L., Miao, X-A. and Liu, T. (2021) 'Using recurrent neural network structure with enhanced multi-head self-attention for sentiment analysis', *Multimedia Tools and Applications*, Vol. 80, pp.12581–12600.

Li, G., Zheng, H., Liu, D., Wang, C., Su, B. and Zheng, C. (2022) 'Semmae: semantic-guided masking for learning masked autoencoders', *Advances in Neural Information Processing Systems*, Vol. 35, pp.14290–14302.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J. and Tang, J. (2021) 'Self-supervised learning: generative or contrastive', *IEEE Transactions on Knowledge and Data Engineering*, Vol. 35, No. 1, pp.857–876.

Ma, Z., Xu, Y., Xu, H., Meng, Z., Huang, L. and Xue, Y. (2021) 'Adaptive batch size for federated learning in resource-constrained edge computing', *IEEE Transactions on Mobile Computing*, Vol. 22, No. 1, pp.37–53.

Nassiri, K. and Akhloufi, M. (2023) 'Transformer models used for text-based question answering systems', *Applied Intelligence*, Vol. 53, No. 9, pp.10602–10635.

Pandey, A. and Kumar, P. (2025) 'BGRU-MTRA: bilinear GRU networks with multi-path temporal residual attention for suspicious activity recognition', *Neural Computing and Applications*, Vol. 37, No. 1, pp.185–212.

Sarker, I.H. (2021) 'Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions', *SN Computer Science*, Vol. 2, No. 6, pp.1–20.

Shobana, J. and Murali, M. (2023) 'An improved self attention mechanism based on optimized BERT-BiLSTM model for accurate polarity prediction', *The Computer Journal*, Vol. 66, No. 5, pp.1279–1294.

Strijkers, K., Chanoine, V., Munding, D., Dubarry, A-S., Trébuchon, A., Badier, J-M. and Alario, F-X. (2019) 'Grammatical class modulates the (left) inferior frontal gyrus within 100 milliseconds when syntactic context is predictive', *Scientific Reports*, Vol. 9, No. 1, p.4830.

Zheng, Z., Huang, S., Weng, R., Dai, X-Y. and Chen, J. (2020) 'Improving self-attention networks with sequential relations', *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 28, pp.1707–1716.