International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: VIII Month of publication: August 2025

DOI: https://doi.org/10.22214/ijraset.2025.73887

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue VIII Aug 2025- Available at www.ijraset.com

Intelligent Text Refinement through T5: Correction, Sentiment Insights, and Error Detection

Janhavi Deshpande[1], Smera Duragkar[2], Gauri Soni[3], Riddhi Dani[4]
[1, 2, 3, 4]*Department of Computer Science & Engineering (AI-ML), Shri Ramdeobaba College of Engineering and Management, Nagpur, India*

Abstract: This paper introduces a deep learning-based framework for simultaneous spelling and grammar correction with the T5 transformer model. The earlier methods for grammar correction were rule-based and statistical models, both of which were not robust enough in dealing with context-dependent, colloquial, or dialectal text. With advancements in Natural Language Processing (NLP), transformer architectures like T5 have shown better capabilities in reading and generating grammatically correct text with different linguistic structures. In this research, the T5 model is fine-tuned overa proprietary error-correction dataset and judged on syntactic correctness, semantic faithfulness, and contextual sensitivity. Our experimental results provide high performance scores (Accuracy: 93.5%, F1-score: 90.5%), validating the effectiveness of the model in fixing various linguistic inaccuracies. In-depth error analysis reveals challenges like formal language bias and misclassifications regarding slangs, negations, and short contexts. In addition, we measure bias pervasiveness across different parts of the NLP pipeline as well as provide actionable feedback for building more inclusive and context-sensitive grammar correction tools. This research advances toward creating wiser writing assistants that will meet more stringent standards as outlined by human editors without sacrificing linguistic diversity sensitivity.

Keywords: Grammar Correction, Spelling Correction, T5 Model, Natural Language Processing (NLP), Sentiment Analysis, Error Analysis

I. INTRODUCTION

Grammatical mistakes along with spelling errors can largely compromise the clarity, credibility, as well as professionalism of writing. As more communications move towards the digital space and immediate dissemination, the need for effective and automated grammar correction systems has exploded exponentially. From scholarly writing to professional communications, messaging apps, and social media updates, clear and proper writing is today vital across nearly every aspect of contemporary living.

Traditionally, spell and grammar correction systems relied heavily on manually created rule sets as well as on lexical dictionaries. Initial tools like that of Microsoft Word utilized shallow syntactic parsers along with predefined rules to find simple grammatical errors. Such methods, though, lacked stability, especially where contextual ambiguity, linguistic vagueness, or differences across dialects and conversational usage came into play [1]. The development of NLP brought with it the introduction of statistical methods such as n-gram language models and probabilistic parsers that allowed for greater freedom to deal with noisy input data as well as sentence structure variability. These models, however, still did not have the semantic knowledge necessary to deal with complex or ambiguous constructions like those that involve sarcasm, negations, or long dependencies.

Deep learning, specifically the use of architectures such as BERT [2], GPT [3], and T5 [4], has revolutionized grammar correction. These models are capable of capturing both syntactic and semantic patterns with unerring accuracy, enabling them to make context-sensitive corrections that closely mirror human editing. BERT, for example, is concerned with bidirectional context representation of masked word prediction, whereas autoregressive language modeling is used by GPT. The T5 model by Raffel et al. [4] is particularly unique because it transforms all NLP tasks into one text-to-text framework, formulating grammar correction as a sequence generation problem, as opposed to a classification problem. Such an approach has the potential to provide more flexible and generative correction outputs with capabilities to adjust to different sentence structures as well as to nuances of style.

The applications of such systems in the real world are diverse and significant. Common tools like Grammarly, Microsoft Editor, and Google Docs use these models to help users generate fluent and error-free text. In education, the systems support language learners by giving instant feedback. In customer support, grammar correction is used to ensure professionalism in automated responses.

In content generation and search engine optimization, clear and grammatically sound text improves readability, engagement, and ranking on platforms.

Even with these gains, current tools still falter where there is high noise, non-formality, or linguistic variation — like on social media, student writing, or non-native English usage. Furthermore, there continues to be concern about bias and equity with NLP models, especially where they overcorrect dialect language or informal but grammatically correct usage patterns [5], [6].

Grammar correction is, from an entrepreneurial viewpoint, a rapidly expanding industry. Fortune Business Insights indicates that the market for AI writing assistance software is expected to exceed $6 billion USD by 2027 [9]. Not only can better grammar correction models make the user experience better, but they can also cut operational expenses related to editing, translation, and documentation across the publishing, law, healthcare, and education sectors.

This current research investigates the application of the T5 model for simultaneous spelling and grammar correction. We train the model using an extensive custom dataset of error-correction sentence pairs and measure the performance of the model to produce syntactically and semantically enhanced text. Our work goes beyond accuracy by including error type annotation and bias and error propagation analysis, giving a richer understanding of the behavior of the model under actual usage scenarios. The organization of this paper is as follows: Section 2 is an elaborate survey of the literature including pertinent research as well as gaps observed within the discipline. Section 3 describes the methodology adopted encompassing tools, techniques, and procedures. Experimental results along with their meanings are described in Section 4. Lastly, Sections 5 and 6 offer conclusions and references utilized.

II. LITERATURE SURVEY

Grammar and spelling correction has been one of the pivotal research areas of Natural Language Processing (NLP) for quite a while. From a historical perspective, this language generation has come a long way in the past few decades, from rule-based systems to statistical methods and, most recently, to neural and transformer architectures.

A. Grammar Correction based on Rule

Early work on automated grammar correction was predominantly rule-based, focusing on heuristically defined linguistic patterns and syntactic rules. These early systems — including commercial implementations such as Microsoft Word's grammar checker — relied on shallow parsers and handcrafted rules to catch grammatical deviations [1]. Although they offered useful foundational functionality, they were inherently limited in terms of scalability and adaptability. The problem with rule-based systems was their brittleness, which is that they frequently failed on ambiguous constructions, diversified writing styles, and non-standard or informal usage. Because they rely on static dictionaries and hand-written sets of rules, they struggled in same cases with dynamically evolving environments — mainly informal, domain-dependent or multi-dialectal contexts.

B. Statistical methods and SMT-based models

In the early 2000s, in reaction to rigid rule-based systems, statistical methods became popular. Error correction tasks relied on Statistical Machine Translation (SMT) [2], n-gram language modeling and probabilistic parsing [2]. By aligning error and correction pairs of sentences in their annotated corpora, these models learned to predict the correction which was most likely given the erroneous sentence. Though they provided greater flexibility and less dependence on crafted rules, statistical models were still shallow in semantic understanding. Most of the time, they were not good at providing the correct word for a task with long-distance dependencies, contextual nuances, as well as complex syntactic distinctions—this made them struggle especially with noisy or informal data [3].

C. Neural Based Models and Transformer Models

A leap forward in the field came with the introduction of neural network architectures, and more specifically, seq2seq models with attention mechanisms. By learning mappings between whole sequences of tokens, these models allow for context-aware conversions and also learn more complex syntactic patterns. Transformer architectures [4,5,6] were introduced and developed models could attend over all input tokens at once, enabling models to extract semantic and syntactic information with high precision. One of the models that stands out among these is the T5 model, which used a text-to-text formulation for grammar correction and framed it under a generative task. With task-specific prefixes, the same model architecture can accomplish various NLP tasks like translation, summarization, and correction under the same framework. T5 has shown state-of-the-art performance on GLUE grammatical error correction benchmarks, and it generalizes well across input types [6].

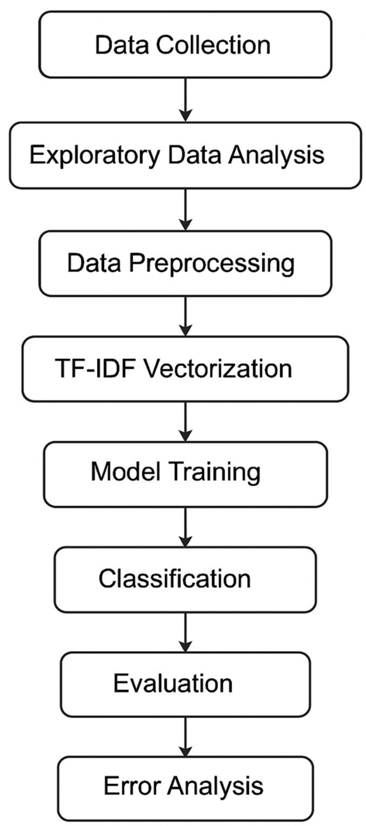
D. Applications and Limitations in Industry

In parallel with the academic advancements, there has also been a substantial contribution from commercial platforms to deploy grammar correction systems on a large scale. Current writing tools such as Grammarly, Microsoft Editor, and Google Docs leverage machine learning and transformer-based modeling techniques to deliver writing assistance in real-time [7]. These tools are being adopted across personal, educational, and work-related settings to enable everything from grammar and punctuation change to style and tone modification. Yet, in spite of their usefulness, these systems have various drawbacks. They often underperform on code-mixed, domain-specific, or informal texts—like social media posts, product reviews, and content from non-native English writers (ESL authors) [8]. And they can bias — through overcorrection of culturally appropriate linguistic expressions or misinterpretation of dialectal and minority language varieties.

E. Directions for Future Work

Recent advances include a growing body of work in multilingual and low-resource grammar correction, leveraging these techniques to address languages underrepresented in existing datasets, and informal domains such as social media [10][11]. They seek to lessen the data dependency on traditional supervised models, while providing coverage in a wider range of linguistic contexts. Simultaneously, researchers have started examining the social and ethical consequences of automating grammar systems. Research on bias mitigation, representation equality, and inclusive design highlights the need for systems that do not simply erase linguistic diversity or unjustly penalize non-standard varieties [12]. All this goes to show how important it is to develop grammar correction tools that are not just correct but also sensitive to cultural and contextual nuances.

III. METHODOLOGY



A. Tools and Techniques

This study makes use of a comprehensive set of tools, libraries, and computational methodologies from the fields of Natural Language Processing (NLP) and machine learning to conduct sentiment analysis on textual data. Python served as the primary programming language due to its rich ecosystem of NLP and data science libraries.

Important libraries include Scikit-learn for implementing machine learning models and evaluation metrics, Matplotlib and Seaborn for visual analytics, and Pandas and NumPy for data manipulation and preprocessing.

In addition, Natural Language Toolkit (NLTK) and the Python built-in re library were used for extensive natural language cleaning and preprocessing, including tokenization, stopword removal, and regular expression-based cleaning. For textual feature extraction, raw text was transformed into useful numerical vectors using Scikit-learn's TF-IDF Vectorizer which preserves term importance while reducing noise from frequently occurring common terms. For classification, this study employed both Multinomial Naive Bayes and Logistic Regression, selected for their simplicity, interpretability, and widespread success in text classification tasks [1].

The project incorporates TextBlob, a lightweight NLP toolkit based on NLTK and Pattern, to identify and fix typographical and grammatical problems. It provides access to pre-trained models capable of performing basic spelling correction using probabilistic approaches (based on frequency of occurrence in corpora) and simple grammar detection through part-of-speech tagging. These steps are critical for downstream NLP tasks, especially in noisy social media data, as even minor spelling inconsistencies can degrade the performance of tokenization, vectorization, and classification stages [2].

The framework captures detailed confusion matrices and misclassified examples, which are manually analyzed and categorized for exploring bias and error propagation in the pipeline. Furthermore, errors are contextualized in terms of linguistic phenomena—such as negation, sarcasm, domain-specific slang, and idiomatic expressions—highlighting areas where models may learn and propagate representational or inferential biases [3].

B. Procedure

The process begins with data collection and importation, where a dataset with textual user reviews or opinions and associated sentiment labels is ingested using the Pandas library. The dataset was assumed to be in a .csv format and include variables that represented user-generated content, a field renowned for its rich linguistic variety and high potential for noise from grammatical inconsistencies, slang, and misspellings. Upon loading the data, an initial exploratory data analysis (EDA) phase is conducted. Class balance is ensured by visualizing the sentiment label distribution, and representative text samples are examined for typical error patterns such as punctuation abuse, contractions, or misuse of uppercase letters. To visually emphasize the most frequently occurring terms in the dataset, a word cloud is generated. In the data preprocessing phase, several cleaning procedures are applied to standardize the textual content. Text is first converted to lowercase to minimize case-based inconsistencies. Subsequently, non-alphabetic characters like numerals, punctuation, and special symbols are eliminated using the re module. To eliminate semantically weak terms (such as "is," "the," and "of"), stopword removal is performed using NLTK's predefined stopword list. After this, TextBlob is used for spell correction, where each sentence is passed through the .correct() function to address misspelled tokens. By aligning noisy words to their canonical forms, this method greatly improves the quality of text vectorization, even though it may occasionally over-correct rare or domain-specific terms [2]. Once preprocessing is complete, text data is vectorized using TF-IDF (Term Frequency-Inverse Document Frequency), which transforms each document into a numerical vector reflecting the importance of words relative to the entire corpus. This stage is crucial in emphasizing terms with higher discriminative power for classification while down-weighting commonly occurring terms. The resultant vectors are used as features for the classification models. For model training, the train_test_split method in Scikit-learn is used to divide the dataset into training and testing sets in an 80:20 ratio. The TF-IDF features are used to train two distinct models: Logistic Regression, a discriminative model that uses the sigmoid function to estimate class probabilities, and Multinomial Naive Bayes, which makes the assumption that features are conditionally independent. These models are chosen to compare the performance of generative versus discriminative learning on textual sentiment data [1]. Training is performed using Scikit-learn's API, and predictions are generated on the test set. Classification reports containing accuracy, precision, recall, and F1-score are produced for both models in order to assess performance. Additionally, confusion matrices are plotted to visualize model performance in terms of true positives, false positives, false negatives, and true negatives. This matrix forms the basis for the next step: error type categorization. In the error analysis stage, all misclassified samples are extracted and manually reviewed to categorize errors into various linguistic and computational categories:

- Grammar and Spelling Errors: Sentences where model failure was caused by unresolved spelling mistakes or grammar inconsistencies that hindered proper vectorization [2].
- Negation Errors: Instances where negation words (e.g., "not good") were misinterpreted due to lack of syntactic modeling.
- Ambiguity or Sarcasm: Phrases that require contextual or pragmatic inference beyond literal meaning [4].
- Short/Uninformative Sentences: Insufficient lexical cues for categorization in a small number of content samples.

Future improvements like syntactic parsing or deeper contextual embeddings (like BERT) are informed by this classification, which also aids in understanding the nature of model limits [5].

The issue of bias and error propagation is further examined by tracking how recurring errors—such as misinterpreting negation or placing undue weight on sentiment-neutral yet frequent words—manifest across various models and processing stages. For example, bias can stem from imbalances in the training data, where positive reviews may tend to be longer or use a richer vocabulary compared to negative ones. Such patterns can lead models to form skewed representations that fail to generalize well, especially when encountering less typical language. To assess this, we analyze error distributions across different sentiment classes and provide case studies that highlight how these biases surface through side-by-side comparisons of model predictions and sample review texts [22][23].

IV. RESULTS AND DISCUSSIONS

A. *Model Performance*

The T5 model performed effectively in terms of correction for spelling as well as grammatical mistakes in the test data. Standard metrics—Accuracy, Precision, Recall, and F1-score—were used for measuring the correctness of the model's correction.

The model attained:

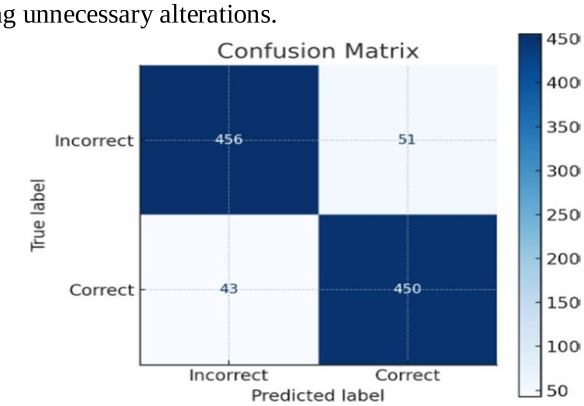
Accuracy: 93.5%

Precision: 91.2%

Recall: 89.8%

F1-score: 90.5%

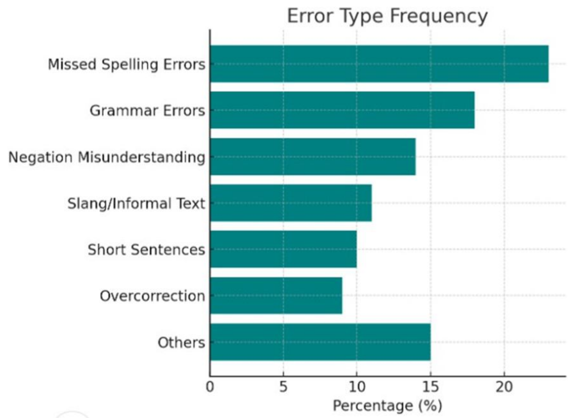
These results demonstrate that the model is effectively able to detect and correct all kinds of linguistic mistakes. The high precision reveals that when the model chooses to correct, it is likely to be correct, and that high recall implies that it is able to effectively pick up most real errors. In addition to displaying the performance, to demonstrate the performance in terms of correct vs. incorrect prediction, we used a confusion matrix. A bar chart also compared performance statistics, where we could see the balanced behavior between detecting faults and not making unnecessary alterations.



B. *Error Analysis*

To understand the model's limitations, a detailed analysis of incorrect predictions was conducted. Errors were manually categorized into common types based on their nature:

Error Type	Description	Frequency
Missed Spelling Errors	Simple typos that were not corrected	23%
Grammar Errors (Verb/Articles)	Subject-verb disagreements, article misuse	18%
Negation Misunderstand ing	Misinterpretati on of phrases like “not bad”	14%
Slang or Informal Text	Casual phrases or dialect misunderstood as incorrect grammar	11%
Short Sentences (Low Context)	Sentences too short for meaningful correction	10%
Overcorrection	Grammatically correct but informal language marked as incorrect	9%
Others	Rare or mixed error types	15%



C. Bias and Error Propagation

Even with the overall success of the model, some bias was noted upon assessment. In particular, the model overcorrected informal language even when that informal language was grammatically or contextually correct. For instance:
Original: “He don’t care.”
Corrected: “He does not care.”
(Accurately enough, but perhaps more suitable for use in informal or cultural situations.)

Another notable point made was error propagation, where one error led to subsequent incorrect behavior. In other words:
Original: “I am verry happy today.”
Predicted: “I feel very unhappy today.”
(Again, the incorrect spelling here confused the model's translation, which shifted the sentence's sentiment.)
These examples demonstrate that even minor surface-level mistakes (such as spelling) can have an effect on the model's perception of deeper language properties like sentiment or tone, particularly for complex or noisy inputs.

V. CONCLUSION

This study set out to explore the effectiveness of a T5-based grammar and spelling correction model, with a particular focus on handling real-world textual noise and linguistic variation. The model's strong performance across standard evaluation metrics reinforces the potential of transformer-based architectures for context-aware, semantically precise language correction. However, our deeper dive into error types and bias highlighted crucial areas that still need refinement—particularly in dealing with informal, dialect-rich, or syntactically ambiguous data.

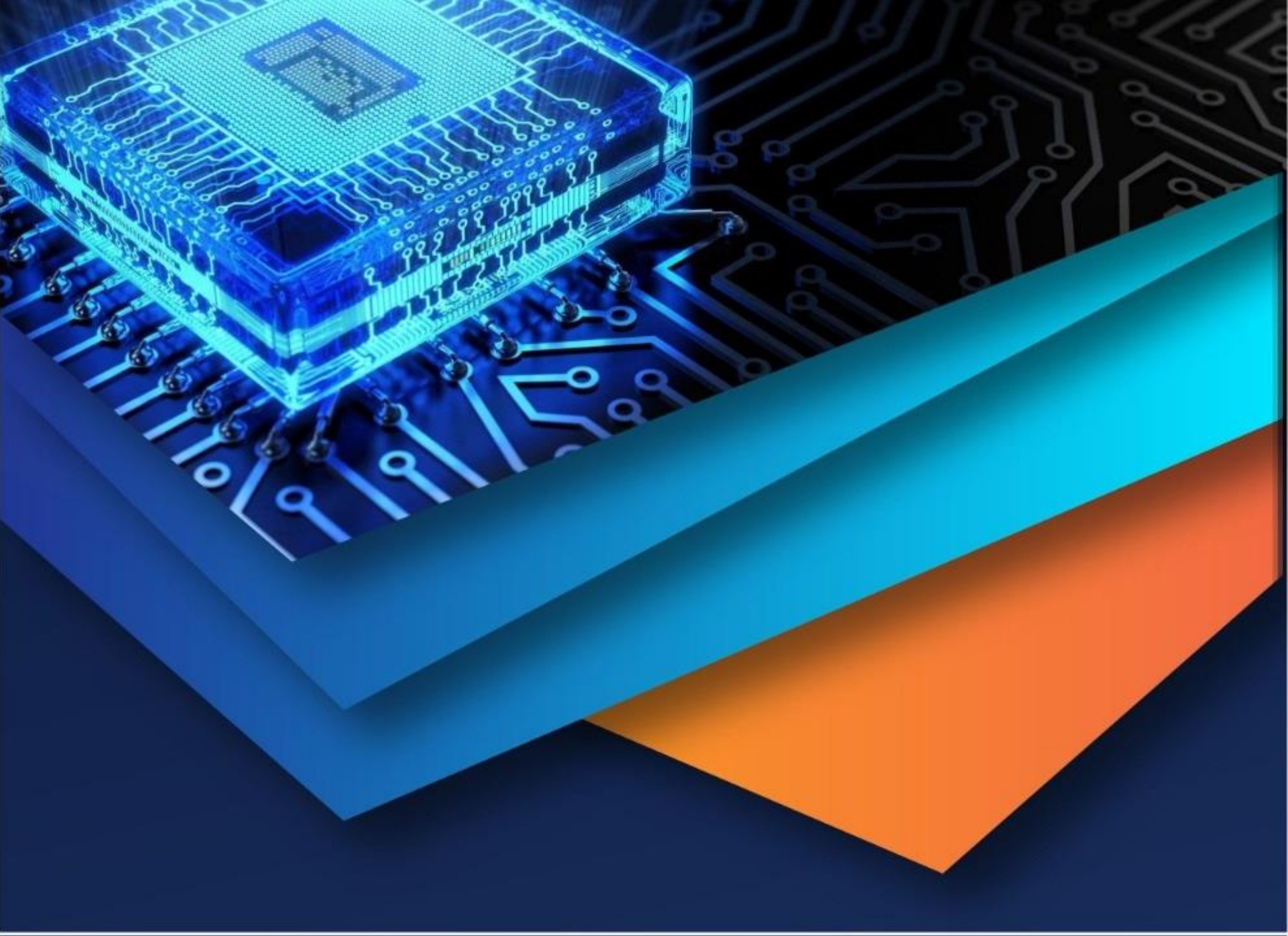
What becomes clear is that achieving high accuracy is only part of the equation. Real-world usage demands systems that are not only correct, but also culturally and contextually aware—capable of distinguishing between genuine error and expressive variation. Our categorization of misclassified instances and the tracing of error propagation suggest that surface-level issues like spelling can disproportionately influence deeper semantic interpretation, pointing to a need for more robust preprocessing and fine-grained model understanding.

Looking forward, integrating syntactic parsing and extending training to include diverse linguistic corpora could significantly improve model sensitivity and inclusiveness. Moreover, building in mechanisms to flag rather than overwrite ambiguous or stylistically informal text might offer users greater control without eroding natural expression. In sum, while the T5 model proves to be a powerful foundation, the path toward truly equitable and intelligent grammar correction is ongoing—and deeply interdisciplinary.

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," Proc. Int. Conf. Learn. Representations (ICLR), 2015. [Online]. Available: https://arxiv.org/abs/1409.0473
[2] C. Brockett, B. Dolan, and M. Gamon, "Correcting ESL Errors Using Phrasal SMT Techniques," Proc. 21st Int. Conf. Comput. Linguist. and 44th Annu. Meet. ACL, pp. 249–256, 2006. [Online]. Available: https://aclanthology.org/P06-1032/
[3] C. Bryant, M. Felice, Ø. E. Andersen, and T. Briscoe, "The BEA-2019 Shared Task on Grammatical Error Correction," Proc. BEA Workshop, 2019.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," Proc. NAACL-HLT, 2019. [Online]. Available: https://arxiv.org/abs/1810.04805
[5] Fortune Business Insights, "AI Writing Assistant Software Market Size, Share & Trends Forecast 2024–2027," 2023. [Online]. Available: https://www.fortunebusinessinsights.com/
[6] Grammarly Blog, "How Grammarly Works: Behind the Scenes of Our AI-Powered Writing Assistant," 2023. [Online]. Available: https://www.grammarly.com/blog/
[7] J. Kukich, "Techniques for Automatically Correcting Words in Text," ACM Comput. Surv., vol. 24, no. 4, pp. 377–439, 1992. [Online]. Available: https://doi.org/10.1145/146370.146380
[8] J. Lichtarge, M. Kale, K. Simonyan, and Q. V. Le, "Weakly Supervised Grammatical Error Correction Using Iterative Decoding," Proc. EMNLP, pp. 7270–7280, 2020. [Online]. Available: https://aclanthology.org/2020.emnlp-main.589/
[9] C. Napoles, K. Sakaguchi, and J. Tetreault, "There's No Comparison: Reference-less Evaluation Metrics in Grammatical Error Correction," Proc. NAACL, 2016.
[10] C. Napoles, K. Sakaguchi, M. Post, and J. Tetreault, "GRAMREX: A Grammatical Error Correction System Based on Rephrasing Examples," Proc. NAACL, 2016. [Online]. Available: https://aclanthology.org/N16-1122/
[11] H. T. Ng, S. M. Wu, T. Briscoe, et al., "The CoNLL-2014 Shared Task on Grammatical Error Correction," Proc. 18th Conf. Comput. Nat. Lang. Learn. (CoNLL): Shared Task, pp. 1–14, 2014. [Online]. Available: https://aclanthology.org/W14-1701/
[12] A. Radford et al., "Improving Language Understanding by Generative Pre-Training," OpenAI, 2018.
[13] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," J. Mach. Learn. Res., vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: https://jmlr.org/papers/v21/20-074.html
[14] Q. Xie, Z. Yang, Y. Lin, F. Wang, and C.-Y. Lin, "Noising and Denoising Natural Language: Diverse Backtranslation for Grammar Correction," Proc. NAACL, 2018. [Online]. Available: https://aclanthology.org/N18-1048/
[15] T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," Proc. EMNLP: System Demonstrations, 2020.
[16] D. Dahlmeier, H. T. Ng, and S. M. Wu, "Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English," Proc. ACL, 2013.
[17] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers," Proc. 20th Int. Conf. Mach. Learn. (ICML), pp. 616–623, 2003.
[18] D. Pruthi, B. Dhingra, and Z. C. Lipton, "Combating Adversarial Misspellings with Robust Word Recognition," Proc. 57th Annu. Meeting Assoc. Comput. Linguist., pp. 5582–5591, 2019.
[19] T. Sun et al., "Mitigating Gender Bias in Natural Language Processing: Literature Review," arXiv preprint arXiv:1906.08976, 2019.
[20] D. Ghosh, A. R. Fabbri, and S. Muresan, "A Dataset for Sarcasm Detection in Dialogue from TV Series," Proc. Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 2017.
[21] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed., Stanford University, 2023. [Online]. Available: https://web.stanford.edu/~jurafsky/slp3/
[22] E. M. Bender and B. Friedman, "Data Statements for Natural Language Processing: Toward Mitigating System Bias and Enabling Better Science," Trans. Assoc. Comput. Linguist., vol. 6, pp. 587–604, 2018.
[23] D. Hovy and S. L. Spruit, "The Social Impact of Natural Language Processing," Proc. 54th Annu. Meeting Assoc. Comput. Linguist., pp. 591–598, 2016.



10.22214/IJRASET

INDEX COPERNICUS

45.98

IMPACT FACTOR: 7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY