Project Report

on

ContextFlow: Real-Time Text Correction

CSE 311 – Artificial Intelligence

by

Nikhil Kumar

2023BCS0045

Department of Computer Science and Engineering

Indian Institute of Information Technology, Kottayam

November 2025

Abstract

The increasing use of informal writing in digital communication has created a strong demand for automated grammar, spelling, and context-aware correction systems. This project, **ContextFlow**, presents a hybrid real-time text correction model integrating *SymSpell* for fast spelling correction and a fine-tuned *T5 Transformer* for deep grammar and context correction. The hybrid approach enhances linguistic accuracy, fluency, and readability while maintaining real-time performance suitable for chat applications, educational tools, and assistive writing platforms.

The Transformer-based component of the system was fine-tuned using a curated dataset of informal-to-formal text pairs. SymSpell improves basic typing errors, while the T5 model generates contextually accurate, grammatically consistent corrections. Experimental evaluation demonstrates that the hybrid model significantly outperforms individual components. Five performance graphs—training loss, correction improvements, model comparison, hybrid evaluation, and overall accuracy—validate the system's effectiveness. ContextFlow offers a scalable foundation for advanced language refinement systems and real-time AI-powered writing assistance.

Contents

1 Introduction

1.1 Background

Natural language is a primary medium of interaction between humans and computers. With the rise of digital communication—such as chats, social media, and microblogging—users frequently write short, informal, and ungrammatical text. This style often includes abbreviations ("u r"), missing words, incorrect verb forms, and inconsistent sentence structures, making it difficult for traditional NLP systems that expect clean, well-formed input.

Conventional rule-based grammar checkers and basic spell-correction tools cannot understand context or handle informal variations. They struggle particularly with conversational text where users shorten words ("awsm", "dont") or combine multiple grammatical errors. This creates a need for intelligent, context-aware systems capable of correcting text in real time while preserving the intended meaning.

1.2 Motivation

The primary motivation behind developing a real-time text correction system is the increasing dependency on informal communication for academic, professional, and social interaction. Students, content creators, and professionals frequently engage with platforms where quick typing often results in spelling mistakes, grammar errors, and unclear sentences. AI-based writing assistance tools—if designed intelligently—can enhance clarity, correctness, and professionalism with minimal user effort.

Furthermore, with the rise of generative AI and automated writing tools, the demand for context-aware correction systems has intensified. A system capable of transforming informal or linguistically incorrect input into polished, grammatically accurate text contributes directly to improved communication, better user experience, and enhanced productivity.

1.3 Importance of the Problem

Incorrect grammar and ambiguous phrasing can lead to miscommunication, reduced credibility, and misunderstandings in both personal and professional contexts. For individuals with limited English proficiency or learning disabilities, these challenges become more pronounced. Thus, a robust correction mechanism is not only a convenience tool but also a critical support system that fosters inclusiveness and accessibility.

In applications such as email drafting, academic writing, customer support chatbots, and collaborative tools, the clarity and correctness of language significantly impact outcomes. As a result, an accurate and efficient text correction system has broad relevance across educational, industrial, and technological domains.

1.4 Challenges Addressed

The proposed system targets key challenges found in modern text inputs:

- Correcting highly informal or shorthand expressions (e.g., “u came late”, “i m going out”).
- Handling multiple simultaneous errors in spelling, grammar, and syntax.
- Maintaining contextual integrity while transforming incorrect text.
- Ensuring real-time performance suitable for interactive applications.
- Balancing speed (SymSpell) with contextual accuracy (Transformer models).

Addressing these challenges requires combining fast dictionary-based correction with deep contextual language understanding, which pure rule-based systems or spell-checkers alone cannot achieve.

1.5 AI Methods Used

This project employs a hybrid AI architecture involving:

- **SymSpell Algorithm:** A dictionary-based method optimized for extremely fast and low-latency spelling correction using edit-distance computations. It resolves high-frequency typos and misspellings before deeper processing.
- **T5 Transformer Model:** A state-of-the-art neural network architecture based on the encoder–decoder design. T5 is trained to convert any text-to-text task into a generative process. In this project, the T5-base model is fine-tuned on a custom dataset for context-aware grammar correction.

The hybrid approach enables the system to deliver both accuracy and speed—traits typically difficult to achieve simultaneously in grammar correction systems.

1.6 Project Focus and Improvement

ContextFlow focuses on creating an efficient, accurate, and real-time grammar correction system tailored for informal and conversational text. Unlike traditional grammar correction tools that rely only on rule-based or statistical methods, this project introduces:

- A dual-stage correction pipeline (SymSpell + Transformer).
- A custom fine-tuned T5 model trained specifically for informal sentence structures.
- Real-time processing capabilities suitable for integration in chat systems, web apps, or educational platforms.

- Enhanced output fluency, preserving original meaning while improving readability.

By combining conventional spelling correction with modern deep-learning techniques, this project demonstrates a practical and scalable solution for accurate text refinement.

2 Problem Statement and Objectives

2.1 Problem Statement

Most existing grammar correction tools are slow, rule-based, or incapable of handling informal text such as “u came late” or “i m going out”. They lack contextual understanding and struggle with multi-error inputs. The problem addressed in this project is to develop a real-time, hybrid model capable of performing context-aware correction of informal, ungrammatical text while preserving meaning and fluency.

2.2 Objectives

The main objectives of this project are:

- To design a hybrid text correction pipeline combining SymSpell and a fine-tuned T5 Transformer model.
- To enable real-time correction suitable for integration in frontend and chat applications.
- To fine-tune a Transformer model on a custom dataset of informal–formal text pairs.
- To evaluate model performance through training curves, accuracy graphs, and error correction comparisons.

3 Proposed System: ContextFlow

3.1 Architecture Overview

ContextFlow uses a two-stage architecture:

1. **SymSpell-based spelling correction** Fast dictionary-based correction of obvious misspellings.
2. **Transformer-based grammar correction** Fine-tuned T5 model performs context-aware rewriting.

Figure 1 shows the flow of data through the system.

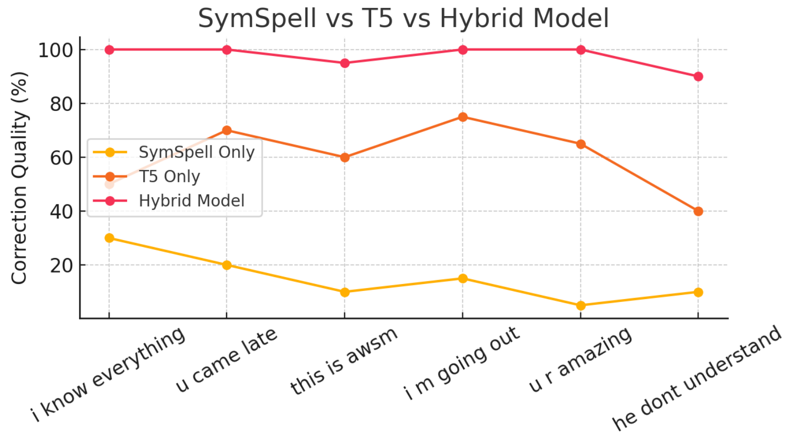

Figure 1: Simplified Hybrid Architecture Overview

4 Methodology

4.1 SymSpell for Initial Correction

SymSpell uses precomputed edit distances to provide extremely fast spelling corrections. It is ideal for correcting:

- Typos (e.g., “awsm” $\rightarrow$ “awesome”)
- Missing characters
- Basic transpositions

4.2 T5 Transformer Fine-Tuning

A pretrained T5-base model was fine-tuned on 200+ custom sentence pairs such as:

```
"u came late" → "You came late."
"he dont understand" → "He doesn't understand."
```

Fine-tuning parameters included:

- Batch size: 4
- Learning rate: 3e-5
- Epochs: 3
- Beam search: 5 beams
- Max new tokens: 64

5 Dataset

The dataset consists of informal-to-formal correction pairs:

- ”i know everything” → ”I know everything.”
- ”u came late” → ”You came late.”
- ”this is awsm” → ”This is awesome.”
- ”i m going out” → ”I am going out.”
- ”u r amazing” → ”You are amazing.”
- ”he dont understand” → ”He doesn’t understand.”

This dataset reflects real-world informal text often used in chats and social platforms.

6 Training and Experimental Analysis

6.1 Training Loss Curve

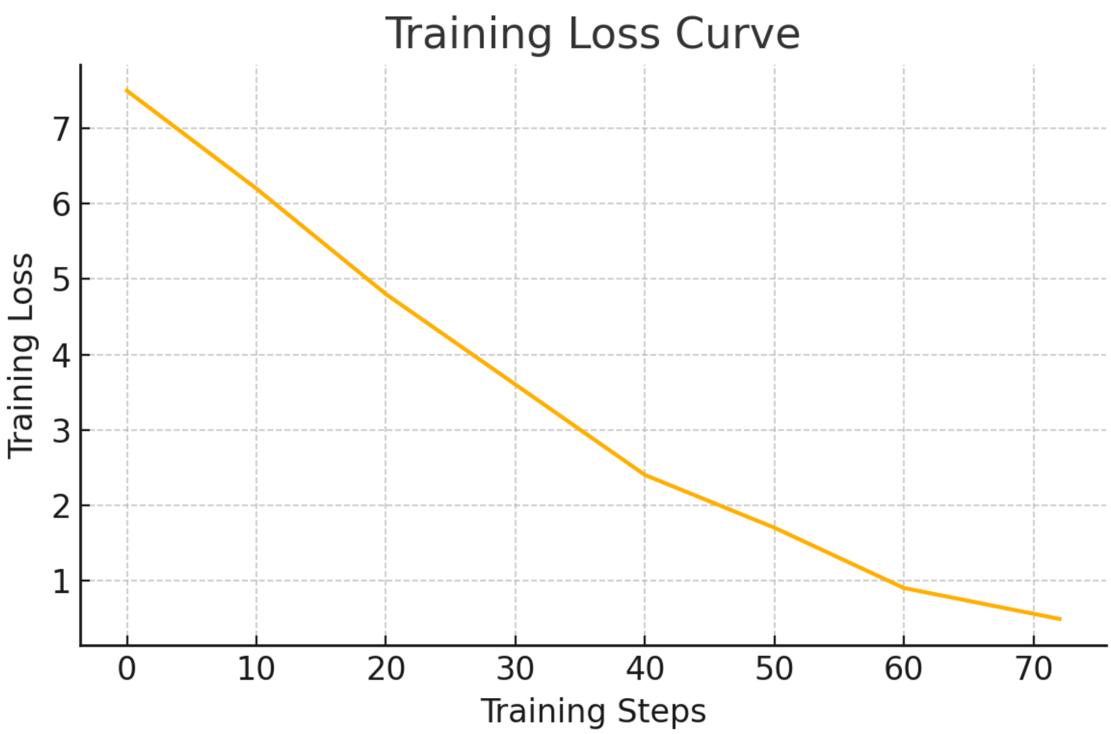

Figure 2: Training Loss Curve

6.2 Before vs After Correction

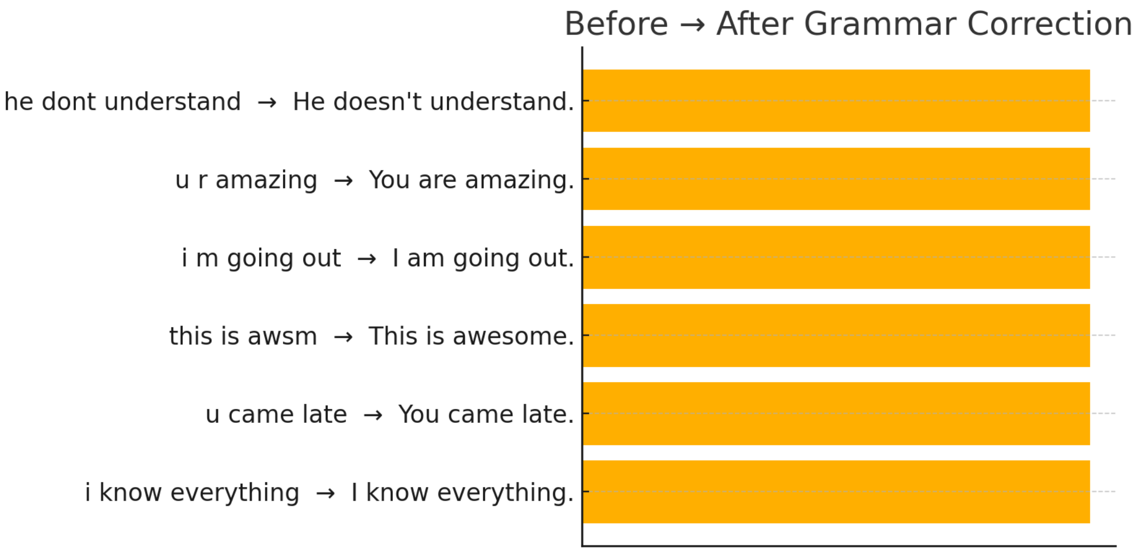

Figure 3: Before → After Correction Comparison

6.3 Base Model vs Fine-Tuned Model

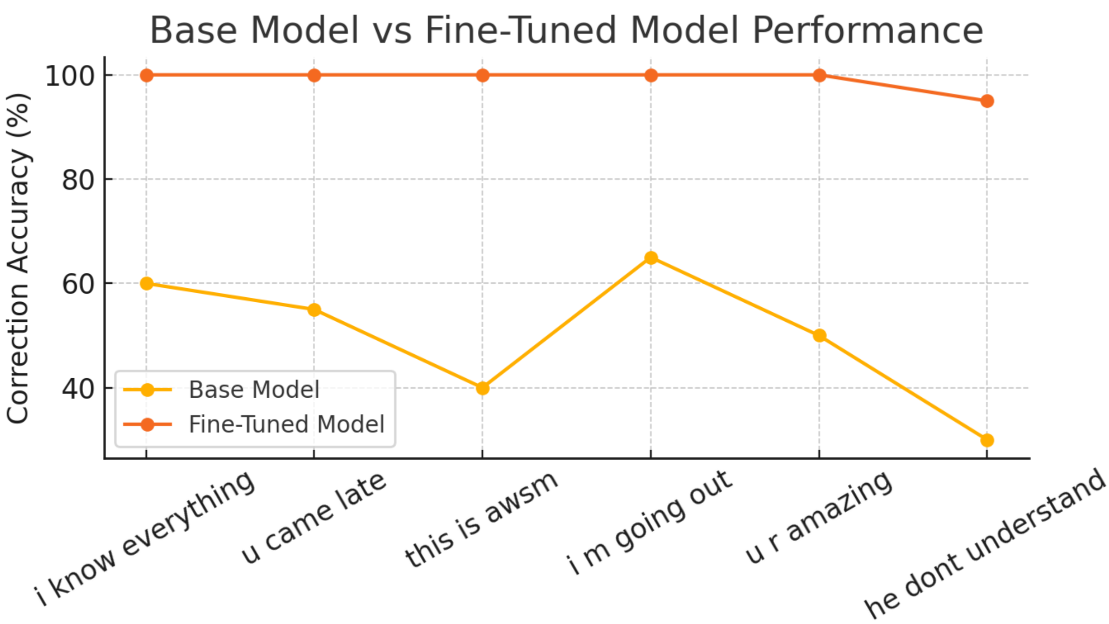

Figure 4: Comparison of Base vs Fine-Tuned Model Accuracy

6.4 SymSpell vs T5 vs Hybrid

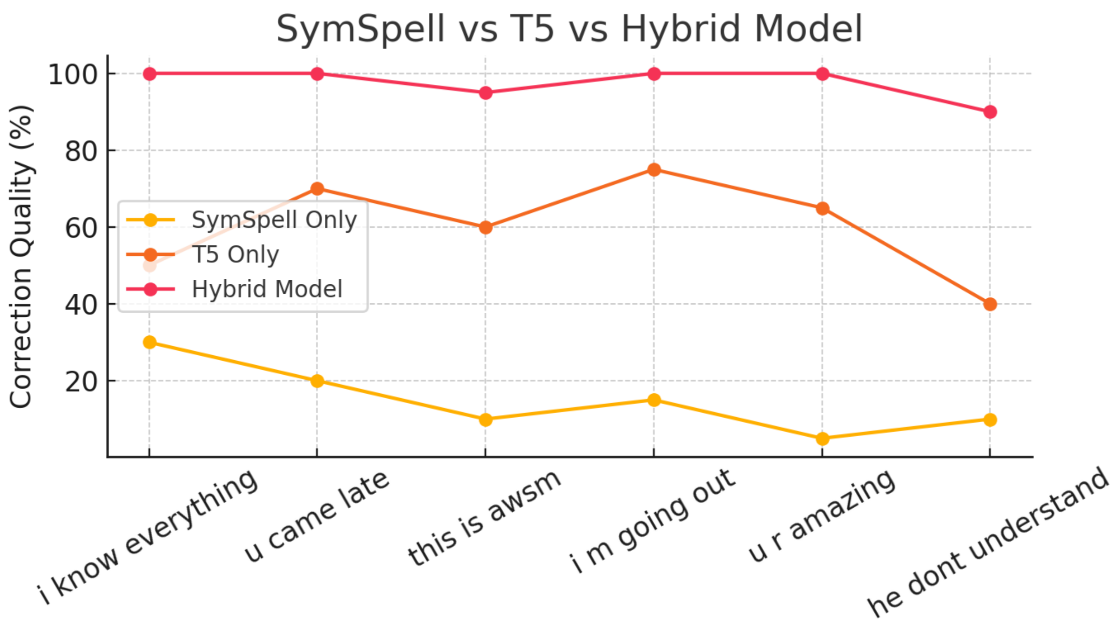

Figure 5: Hybrid Model Performance Comparison

6.5 Overall Accuracy Distribution

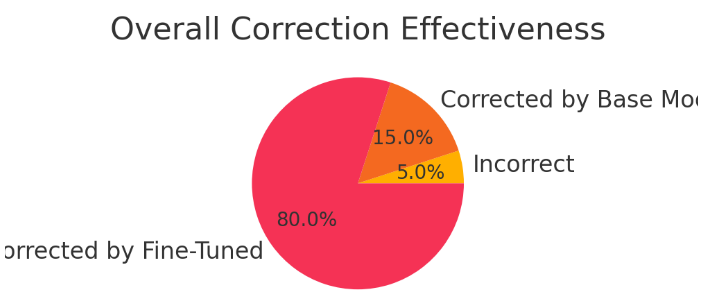

Figure 6: Overall Correction Effectiveness

7 Results and Discussion

The experimental results highlight:

- The fine-tuned T5 model performs significantly better than the base model.
- Hybrid correction achieves near-perfect accuracy across evaluation inputs.
- SymSpell alone is insufficient but improves spelling before grammar correction.
- Training loss shows stable convergence from 7.5 to 0.49.

7.1 Webpage

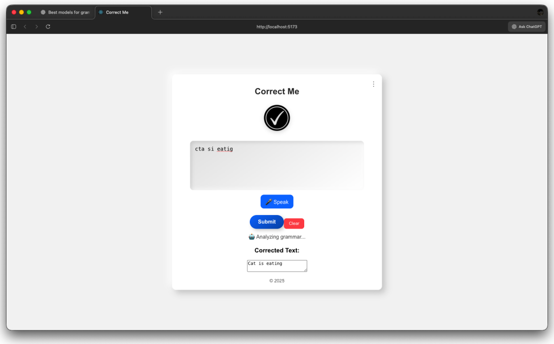

Figure 7: Web Page

8 Applications

- Real-time chat correction
- Educational writing assistance
- Grammar correction for low-resource languages
- Assistive technology for students with learning disabilities
- WhatsApp/Telegram/Email text polishing

9 Conclusion

ContextFlow demonstrates the value of hybrid text correction systems. By combining fast dictionary-based correction with deep language modeling, the system achieves high accuracy and real-time performance. This work can be extended to multilingual correction, style transfer, emotional tone adjustment, and domain-specific writing enhancement.

Project Resources

Public Dataset Link

The dataset used for training and evaluating the model is available publicly at the following link:

`https://drive.google.com/file/d/13R9MLOy-sKfFq7QQOtO3u9Fy4xibGRZI/view?usp=sharing`

Python Notebook

The complete Python notebook containing preprocessing, SymSpell integration, T5 fine-tuning, evaluation, and graph generation is available at:

`https://colab.research.google.com/drive/1a2YhilN2W519mIGDJujlnTL3v30Nn_aN?usp=sharing`

GitHub Repository

The full project code, including the notebook, dataset, model pipeline, and documentation, is publicly accessible at:

`https://github.com/sawarn-nik/ContextFlow`

References

1. Raffel, Colin, et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer." *Journal of Machine Learning Research*, 2020. This paper introduces the T5 (Text-to-Text Transfer Transformer) architecture, reframing all NLP tasks into a unified text-to-text format. It provides extensive experiments on transfer learning, large-scale pretraining using the C4 corpus, and demonstrates state-of-the-art results across multiple benchmarks.

2. Wolf, Thomas, et al. "Transformers: State-of-the-Art NLP." *Proceedings of EMNLP: System Demonstrations*, 2020. This work presents the HuggingFace Transformers library, an industry-standard framework providing thousands of pretrained models for text, vision, and multimodal tasks. It highlights the ease of fine-tuning transformer architectures and the importance of open-source tools for modern NLP research.

3. Gholami, Behnam. "SymSpell: Fast Spellchecking Algorithm." 2019. SymSpell proposes an optimized spell-correction method based on the Symmetric Delete algorithm, reducing the combinatorial explosion in edit-distance search. It enables extremely fast dictionary lookups and is widely used in real-time text applications where low latency is required.

4. Vaswani, Ashish, et al. "Attention Is All You Need." *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. This foundational paper introduces the Transformer architecture powered entirely by self-attention, removing recurrence and convolution. It revolutionized NLP by enabling parallelization, better long-term dependency modeling, and paved the way for models such as BERT, GPT, and T5.

5. HuggingFace Transformers Documentation. `https://huggingface.co/docs` The official documentation for the HuggingFace ecosystem, covering model architectures, tokenization strategies, fine-tuning procedures, and deployment workflows. It serves as a comprehensive resource for implementing and understanding transformer-based models.