Context-Aware Autocorrect and Autocomplete: Enhancing Mobile Typing with NLP

Faheem Arif*, M.S.R Siddarth Reddy*
*Mathematics and Computing, Epoch, IIT Hyderabad, Hyderabad, India
Email: ma23btech11010@iith.ac.in
*Mathematics and Computing, Epoch, IIT Hyderabad, Hyderabad, India
Email: ma23btech11017@iith.ac.in

***Abstract*—This paper presents a comprehensive study on developing context-aware autocorrect and autocomplete models to enhance mobile typing experiences using Natural Language Processing (NLP). Our work explores two major approaches: a BERT-based model for autocorrect and a Seq2Seq model leveraging T5-small for paragraph-level corrections. We detail the challenges faced, solutions implemented, and experimental findings. The project also outlines the design of a user interface to integrate these models effectively and provides future directions for improvement.**

***Index Terms*—NLP, Autocorrect, Autocomplete, BERT, T5-small, Curriculum Learning.**

I. Introduction

Mobile typing often suffers from errors caused by typos, missing spaces, or incorrect slang usage. These issues can significantly impact user experience. This project focuses on building context-aware autocorrect and autocomplete models to address these challenges effectively. The goal was to create a robust system capable of handling real-world typing scenarios by leveraging state-of-the-art NLP techniques.

The initial objective was twofold:

- Develop an autocorrect system that considers contextual information.
- Implement an autocomplete system to predict subsequent words or phrases.

However, due to time constraints, much of the work focused on autocorrect functionality.

II. Literature Survey

Our literature survey highlighted several existing methods and datasets for spelling correction and autocomplete tasks:

- Pre-trained models like BERT have been used for masked word prediction tasks but face limitations in handling multi-token errors and niche vocabulary.
- Seq2Seq models such as T5 excel in text-to-text tasks by converting input-output pairs into a unified format.
- Curriculum learning has shown promise in improving model performance by structuring training data from simple to complex instances.
- Existing datasets often lack realistic slang usage or sufficient diversity in error types.

III. Dataset Acquisition

To train our models effectively, we required a dataset that simulated real-world typing errors:

- The dataset needed to include keyboard proximity typos, missing spaces, and added spaces.
- Slang vocabulary was identified as a critical gap in existing datasets.

Initially, we planned to synthetically generate such a dataset. However, we discovered an existing dataset with multiple levels of error generation for IMDB and Amazon reviews. Despite its lack of slang coverage, this dataset saved significant time and effort.

IV. Methodology

Our methodology evolved through two primary approaches:

A. BERT-Based Approach

The initial approach used a BERT-type model for autocorrect:

1) After each space is entered, the model runs inference k times on the last k-words (where $k = 2 - 3$).
2) Each word is masked sequentially, and top K predictions are generated.
3) The word with the least edit distance from the current word is selected unless it exceeds a threshold (e.g., 2-3 edit distances).
4) Named Entity Recognition (NER) was used to skip corrections for named entities.
5) A reranking mechanism was proposed to refine predictions based on contextual relevance.

Challenges faced:

- Niche words often failed to appear in predictions due to limited vocabulary.
- Multi-token replacements were problematic for longer words.
- Extra or missing spaces disrupted masking-based predictions.
- Contextual errors involving adjacent words were difficult to resolve.

B. Seq2Seq Approach with T5-Small

To overcome these limitations, we pivoted to a Seq2Seq approach using T5-small:

1) The input consisted of complete sentences or paragraphs with errors.
2) The target labels were corrected versions of the input text.
3) Curriculum learning was employed by fine-tuning models on datasets with increasing error complexity over three epochs each.
4) Surprisingly, the model trained exclusively on the hardest dataset performed best across metrics such as Word Error Rate (WER), Character Error Rate (CER), and Exact Match.

V. User Interface Design

The user interface (UI) was designed with usability in mind:

- Corrections were applied only to the last entered sentence to preserve user intent.
- For sentences with differing word counts between input and output (e.g., missing/extra spaces), entire sentence suggestions were provided.
- Pop-up suggestions appeared above individual words for simpler corrections.
- Real-time feedback ensured seamless integration into typing workflows.

VI. Future Work

Several enhancements are planned:

1) Develop a custom error-generation system to incorporate slang vocabulary into training data.
2) Fine-tune larger models like T5-base or T5-large while employing quantization techniques for efficiency.
3) Implement autocomplete functionality by predicting the next 2-3 words of incomplete sentences using similar Seq2Seq principles.

VII. Conclusion

This project demonstrates how NLP techniques can enhance mobile typing experiences through context-aware autocorrect and autocomplete functionalities. While challenges remain, our findings provide valuable insights for future advancements.

For implementation details and codebase access, visit our GitHub repository:

https://github.com/FA0206/ContextualAutoCorrect/blob/main/ML/autocorrectfinetunet5.ipynb