International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 06 | June - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Autocorrect Misspelled Word Search Engine in Python

Naznin Shaikh, Mr. Ali Karim Sayed
Department of B.Voc (Artificial Intelligence and Data Science)
Anjuman-I-Islam's Abdul Razzak Kalsekar Polytechnic, New Panvel

Abstract
This paper presents the development of an intelligent autocorrect system designed to enhance search engine performance by correcting misspelled user queries. Built using Python, the system leverages NLP techniques like Levenshtein Distance, fuzzy matching (via fuzzywuzzy), and probabilistic models to suggest accurate alternatives. The autocorrect engine processes queries by tokenizing them, comparing against a vocabulary, and selecting the most likely correction. Designed to be lightweight and easily integrable, the solution is ideal for real-time applications in education, e-commerce, and healthcare. The results demonstrate high accuracy for common typographical errors with potential for multilingual and domain-specific adaptations.

Key Words:
Autocorrect, Natural Language Processing, Python, Levenshtein Distance, Fuzzy Matching, Search Engine.

1. INTRODUCTION

Misspellings in user input are a major obstacle to accurate information retrieval in search engines. Traditional systems often return poor or no results when queries contain typographical errors. This project proposes a Python-based autocorrect engine that intelligently detects and corrects such mistakes in real-time. It uses NLP techniques to compare user inputs against a known corpus and suggest the most likely correction. This system is lightweight and can be adapted for various applications including academic portals, customer support bots, and mobile apps.

2. METHODOLOGY

The system is implemented in Python using libraries such as `re`, `collections.Counter`, `string`, and `fuzzywuzzy`. It starts by tokenizing a large text corpus to form a vocabulary. The core autocorrect logic generates candidate corrections through edit operations and ranks them using word frequency probability. Known words from these candidates are compared and the most probable correction is selected. The design is efficient and performs well in resource-constrained environments.

3. RESULTS

The system successfully corrected a variety of common spelling errors. For example:
- speling → spelling
- writting → writing
- corect → correct

Accuracy exceeded 90% for single-edit misspellings. The tool also allows for domain-specific vocabulary integration, improving relevance in specialized fields.

4. CONCLUSION

The developed autocorrect engine is a practical, lightweight solution for enhancing the usability of search systems. It uses basic NLP methods to provide real-time corrections without relying on heavy ML models. Future improvements could include context-aware corrections and multilingual support.

REFERENCES

1. Peter Norvig. How to Write a Spelling Corrector, 2007.
2. FuzzyWuzzy Python Library. https://github.com/seatgeek/fuzzywuzzy
3. TextBlob Documentation. https://textblob.readthedocs.io
4. Python re, string, collections modules - Official Documentation.