**SPELLING GRAMMAR CHECK APP**

**by**

**Shruti Andraskar (MST03-0067)**

**Submitted to Scifor Technologies**



**UNDER GUIDANCE OF**

**Urooj Khan**

## TABLE OF CONTENTS

1. Abstract

2. Introduction

3. Technology Used

4. Dataset Information

5. Methodology

6. Code Snippet

7. Results and Discussion

8. Conclusion

9. Reference

## Abstract

The Streamlit application described in this report provides a user-friendly interface for checking and correcting spelling and grammar errors in both plain text and DOCX files. Leveraging TextBlob for spell checking and LanguageTool for grammar correction, the application offers an integrated solution for enhancing text quality. Users can input text directly or upload DOCX files for automated correction, with the corrected results available for download. This report outlines the technologies used, methodology, and results of the application.

## Introduction

In the digital age, accurate and error-free text is crucial for effective communication. Automated tools for spelling and grammar correction can significantly enhance the quality of written content. This Streamlit application aims to address the need for such tools by providing a seamless interface for text input and file upload, allowing users to correct spelling and grammar errors in their documents. The application utilizes advanced algorithms and libraries to deliver reliable corrections and suggestions.

## Technology Used

 **Streamlit**: An open-source app framework for creating web applications with Python. It simplifies the development of interactive applications and visualizations.

 **TextBlob**: A Python library for processing textual data, including spelling correction and sentiment analysis. It uses probabilistic models to correct spelling errors.

 **LanguageTool**: An open-source grammar checker that supports multiple languages. It employs rule-based algorithms to detect and suggest corrections for grammatical errors.

 **python-docx**: A library for reading and writing DOCX files in Python. It is used to handle file uploads and generate downloadable corrected documents

## Dataset Information

 **Text Data**: The dataset consists of plain text input from users and text extracted from DOCX files. The application does not utilize a pre-defined dataset but processes user-provided content in real-time.

 **DOCX Files**: Users can upload DOCX files containing text that will be read, processed, and corrected by the application

## Methodology

 **Text Input Processing**:

* Users input plain text into a text area.
* The text is first processed by TextBlob for spell checking.
* The corrected text is then passed to LanguageTool for grammar checking.
* The application displays the corrected text and lists grammar mistakes with suggestions.

 **DOCX File Processing**:

* Users upload DOCX files through a file uploader.
* The file's content is extracted using python-docx.
* The extracted text is corrected for spelling and grammar using the same processes applied to plain text.
* The corrected content is saved into a new DOCX file and made available for download

# Code Snippet

Below is the complete code used for this project:

import streamlit as st

from textblob import TextBlob

import language\_tool\_python

from io import StringIO

from io import BytesIO

from docx import Document

class SpellCheckerModule:

def \_\_init\_\_(self):

self.tool = language\_tool\_python.LanguageTool('en-US')

def correct\_spell(self, text):

words = text.split()

corrected\_words = [str(TextBlob(word).correct()) for word in words]

return " ".join(corrected\_words)

def correct\_grammar(self, text):

matches = self.tool.check(text)

corrections = [{'incorrect': match.context, 'suggestions': match.replacements} for match in matches]

corrected\_text = language\_tool\_python.utils.correct(text, matches)

return corrected\_text, corrections, len(matches)

st.title("Spell and Grammar Checker")

spell\_checker\_module = SpellCheckerModule()

st.header("Text Input Checker")

input\_text = st.text\_area("Enter text for spell and grammar checking:")

if st.button("Check Spelling and Grammar"):

corrected\_text = spell\_checker\_module.correct\_spell(input\_text)

corrected\_text, grammar\_corrections, grammar\_mistakes = spell\_checker\_module.correct\_grammar(corrected\_text)

st.write("Corrected Text:")

st.write(corrected\_text)

st.write(f"Grammar Mistakes Found: {grammar\_mistakes}")

for correction in grammar\_corrections:

st.write(f"Incorrect: {correction['incorrect']}")

st.write(f"Suggestions: {', '.join(correction['suggestions'])}")

st.header("File Upload Checker")

uploaded\_file = st.file\_uploader("Upload a Word file (.docx) for spell and grammar checking:", type=["docx"])

if uploaded\_file is not None:

doc = Document(uploaded\_file)

full\_text = [para.text for para in doc.paragraphs]

file\_content = '\n'.join(full\_text)

corrected\_file\_text = spell\_checker\_module.correct\_spell(file\_content)

corrected\_file\_text, corrected\_file\_grammar, grammar\_mistakes = spell\_checker\_module.correct\_grammar(corrected\_file\_text)

st.write("Corrected File Text:")

st.write(corrected\_file\_text)

st.write(f"Grammar Mistakes Found: {grammar\_mistakes}")

for correction in corrected\_file\_grammar:

st.write(f"Incorrect: {correction['incorrect']}")

st.write(f"Suggestions: {', '.join(correction['suggestions'])}")

output\_doc = Document()

output\_doc.add\_paragraph(corrected\_file\_text)

output\_io = BytesIO()

output\_doc.save(output\_io)

st.download\_button(

label="Download Corrected File",

data=output\_io.getvalue(),

file\_name="corrected\_file.docx",

mime="application/vnd.openxmlformats-officedocument.wordprocessingml.document"

)

## Results and Discussion

* **Text Input Results**: The application effectively corrects spelling errors using TextBlob and provides detailed grammar corrections through LanguageTool. Users receive immediate feedback on spelling and grammar issues.
* **DOCX File Results**: The application handles DOCX files by extracting text, applying corrections, and allowing users to download the corrected document. This feature is useful for users needing to proofread longer documents.

**Discussion**:

* The spell-checking accuracy of TextBlob is generally reliable for common words but may struggle with domain-specific or unusual terms.
* LanguageTool provides comprehensive grammar checking but may not capture every context-specific error.
* The application’s ability to handle both text input and DOCX files ensures flexibility and usability for various user needs.

## Conclusion

The Streamlit application provides an efficient solution for correcting spelling and grammar errors in both plain text and DOCX files. By integrating TextBlob and LanguageTool, the application delivers accurate corrections and user-friendly functionality. Future enhancements could include support for additional languages, improved context-aware corrections, and a more customizable user interface.

**REFERENCE**

 **Streamlit Documentation**: https://docs.streamlit.io/

 **TextBlob Documentation**: <https://textblob.readthedocs.io/>

 **LanguageTool Documentation**: <https://languagetool.org/>

 **python-docx Documentation**: <https://python-docx.readthedocs.io/>