```
!pip install pyspellchecker
```

```
Collecting pyspellchecker
  Downloading pyspellchecker-0.8.2-py3-none-any.whl.metadata (9.4 kB)
Downloading pyspellchecker-0.8.2-py3-none-any.whl (7.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.1/7.1 MB 58.5 MB/s eta 0:00:00
Installing collected packages: pyspellchecker
Successfully installed pyspellchecker-0.8.2
```

```
from transformers import pipeline
from spellchecker import SpellChecker
import nltk

# Download necessary resources (run this in your Colab notebook)
nltk.download('punkt')

spell = SpellChecker()
text_gen = pipeline("text-generation", model="gpt2")

def check_spelling(text):
    words = text.split()
    misspelled = spell.unknown(words)
    return list(misspelled)

def correct_spelling(text):
    words = text.split()
    corrected_words = [spell.correction(word) if word in spell.unknown([word]) else word for word in words]
    return ' '.join(corrected_words)

def autocomplete_text(prefix, max_length=50):
    try:
        completions = text_gen(prefix, max_length=max_length, num_return_sequences=1, do_sample=True)
        return completions[0]['generated_text']
    except Exception as e:
        return f"Error during autocompletion: {e}"

def colab_interface():
    """Provides a command-line interface for the NLTP tools in Google Colab."""
    print("Welcome to the NLTP Spelling and Autocomplete Tool in Google Colab!")
    while True:
        print("\nChoose an option:")
        print("1. Check Spelling")
        print("2. Correct Spelling")
        print("3. Autocomplete Text")
        print("4. Exit")

        choice = input("Enter your choice (1-4): ")

        if choice == '1':
            text = input("Enter your text: ")
            result = check_spelling(text)
            print("Misspelled Words: " + ", ".join(result))
        elif choice == '2':
            text = input("Enter your text: ")
            result = correct_spelling(text)
            print("Corrected Text:\n" + result)
        elif choice == '3':
            text = input("Enter your text: ")
            result = autocomplete_text(text)
            print("Autocomplete Result:\n" + result)
        elif choice == '4':
            print("Exiting...")
            break
        else:
            print("Invalid choice. Please try again.")

if __name__ == "__main__":
    colab_interface()
```

```
ata] Downloading package punkt to /root/nltk_data...
ata]   Unzipping tokenizers/punkt.zip.
cal/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
ret `HF_TOKEN` does not exist in your Colab secrets.
enticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it as secre
l be able to reuse this secret in all of your notebooks.
note that authentication is recommended but still optional to access public models or datasets.
ngs.warn(
on: 100%                                                  665/665 [00:00<00:00, 54.3kB/s]
rage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to regular HTTP download. For better perfo
:huggingface_hub.file_download:Xet Storage is enabled for this repo, but the 'hf_xet' package is not installed. Falling back to r
fetensors: 100%                                           548M/548M [00:02<00:00, 255MB/s]
on_config.json: 100%                                      124/124 [00:00<00:00, 11.1kB/s]
_config.json: 100%                                        26.0/26.0 [00:00<00:00, 2.29kB/s]
on: 100%                                                  1.04M/1.04M [00:00<00:00, 4.77MB/s]
xt: 100%                                                  456k/456k [00:00<00:00, 3.17MB/s]
.json: 100%                                               1.36M/1.36M [00:00<00:00, 6.39MB/s]
set to use cpu
 to the NLTP Spelling and Autocomplete Tool in Google Colab!

an option:
k Spelling
ect Spelling
complete Text

led Words: misssing

an option:
k Spelling
ect Spelling
complete Text

led Words:

an option:
k Spelling
ect Spelling
complete Text

our choice (1-4):
```