



VIT[®]

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

DATA STRUCTURES AND ALGORITHMS PROJECT REPORT

Title:

Paragraph Classification and Spell Checker – A text processing tool

1. V. Ganesh Kumar Reddy(22BEC0674)

2. Y. Vignesh Reddy (22BEC0658)

Abstract:

This project presents a user-friendly application for finding and correcting the spelling mistakes and categorizing those paragraphs into categories that are already defined. This tool uses a dictionary-based approach, the misspelled word will be corrected by first nearest word from the pre-defined dictionary, it will also give autosuggestions for the misspelled words with the first three nearest words from the dictionary. It will analyse the content and categorize the certain paragraph to classify it into their respective domains such as finance, technology, sports, history, science, or politics. It is also implemented using Python and Tkinter, it provides a user-friendly and interactive GUI (Graphical User Interface) for efficient text processing.

Introduction:

In present digital era, written communication plays a crucial role in various fields, like sending email, reports, research papers. In those written communication, spelling errors and less structured content can reduce the professionalism. To address this spelling mistake issue, we developed a application which helps for spell checking and paragraph classification, this is a tool to improve our written content by auto-correcting spelling mistakes and categorizing paragraph based on their context.

The main reason behind this project or application is to help individuals, like students, in improving their writing quality. Our application utilizes a dictionary-based approach for spell checking, along with auto suggestions for the misspelled words. Additionally, it also categorizes the paragraph into domains such as finance, technology, sports, history, science, and politics.

Our application implements Levenshtein distance for spell checking using Python's "difflib" library. We used set-based keyword matching for paragraph categorization. We also used Python and tkinter, which is a graphical user interface (GUI) for user friendly interaction and easily accessible.

This is also useful for real-world applications like these spell checkers are used in word processors, search engines, and chat applications. Also, categorization is essential for content recommendation, filtering, and topic-based search engines.

Literature Review:

Text processing is an important area of natural language processing (NLP) that includes things like spell checking and paragraph categorization. Researchers have studied many ways to improve the accuracy and efficiency of text processing processes. Paragraph classification is the sorting text into predefined categories based on its content. Traditional approaches traditionally used rule-based approaches that required human labor to write linguistic rules (Jurafsky & Martin, 2021) [3]. Newer approaches, as machine learning and deep learning have advanced, use algorithms, including deep neural networks, Support Vector Machines (SVM), and Naïve Bayes (Le & Mikolov, 2014) [6].

Recent studies have shown that transformer-based models like GPT (Generative Pre-trained Transformer) and BERT (Bidirectional Encoder Representations from Transformers) yield significantly better performance than classical machine learning methods in paragraph classification tasks (Devlin et al., 2019) [2]. While classical methods capture word meaning as independent of context, transformer models address position and context with embeddings, resulting in more accurate semantic distinctions. Recurrent neural networks (RNNs) and convolutional neural networks (CNNs) also have been effective for text classification (Kim, 2014) [4].

Spell Checking:

Spell checking is another basic text processing task that involves detecting and correcting spelling errors. Early spell checkers used dictionary-based methods that compared words against a pre-specified dictionary. Their approaches struggled with grammatical context challenges and words outside of the dictionary (Kukich, 1992) [5]. Modern spell-checking systems integrate probabilistic models, such as Hidden Markov Models (HMMs), and machine-learning models, such as edit distance algorithms, like Levenshtein Distances (Brill & Moore, 2000) [1]. Notably, deep-learning models like sequence-to-sequence models and attention-based models have significantly improved spelling correction accuracy by considering the broader context of the text (Sakaguchi et al., 2017) [7]. Spell checkers for Google and Microsoft employ neural network technology to continually learn and improve their systems in correcting spelling mistakes.

Integration in Text Processing Tools:

Combining spell checking and paragraph classification into a single text processing tool leads to improvements in content moderation, search engine optimization, and automated document processing (Schuster et al., 2019) [8]. TensorFlow and spaCy are two of the recent advancements in NLP frameworks that provide powerful services for combining these functionalities within a single platform. Because of advances in text-processing technology, the effectiveness of both spell-checking and paragraph classification has greatly increased. These tasks are now more reliable and effective because of the advancements in probabilistic methods, contextual embeddings, and deep learning models. Future research may focus on enhancing multilingual capabilities, lowering processing costs, and increasing real-time performance.

Proposed Methodology:

For this Paragraph Classification and Spell Checker – A text processing tool, it has certain order. If we enter the paragraph from the collected dataset then,

The function reads a dictionary text file that contains correctly spelled words. It uses regex to extract the words and saves them to a collection for convenient access. It will terminate with an error if the file is not present.

It also auto suggests and determines the three nearest terms in the dictionary using `difflib.get_close_matches`. For Auto-Correction the term stays the same if it has numbers or is listed a dictionary. If it is misspelled, the nearest correct term replaces it. The GUI shows separating corrections and suggestions.

It also splits content into fixed categories like finance, technology, sport and sports, history, science, and politics by identifying them as keyword. Lowers text to lowercase, tokenizes words and checks them against keywords for each category. chooses the category that matched the most words.

It also obtains input from a text box and processes it splits the paragraph into words and addresses punctuation. each word is spell-checked, and any erroneous words are fixed reconstructs the paragraph that has been corrected. assigns a content classification for the improved paragraph. displays the category identified, the text that has been corrected, and suggestions for spell-checking.

It also provides a field for entering text. a button for processing the text. Returns the resulting output, including what category the paragraph fell into, the revised text, and any suggested spell-check changes. uses Scrolled Text, which makes reading long text entries easier.

The system integrates spell-checking and paragraph categorization features in one software program. utilizes regex and a dictionary-based spell-checking approach. provides categorization of paragraphs and edits in real time. For more advanced paragraph categorization, you can use some NLP models (e.g. BERT). Additionally, use deep learning to design a more advanced spell-checking application.

Graphical User Interface:

Spell Checker & Paragraph Categorizer

Enter Paragraph:

Check & Categorize

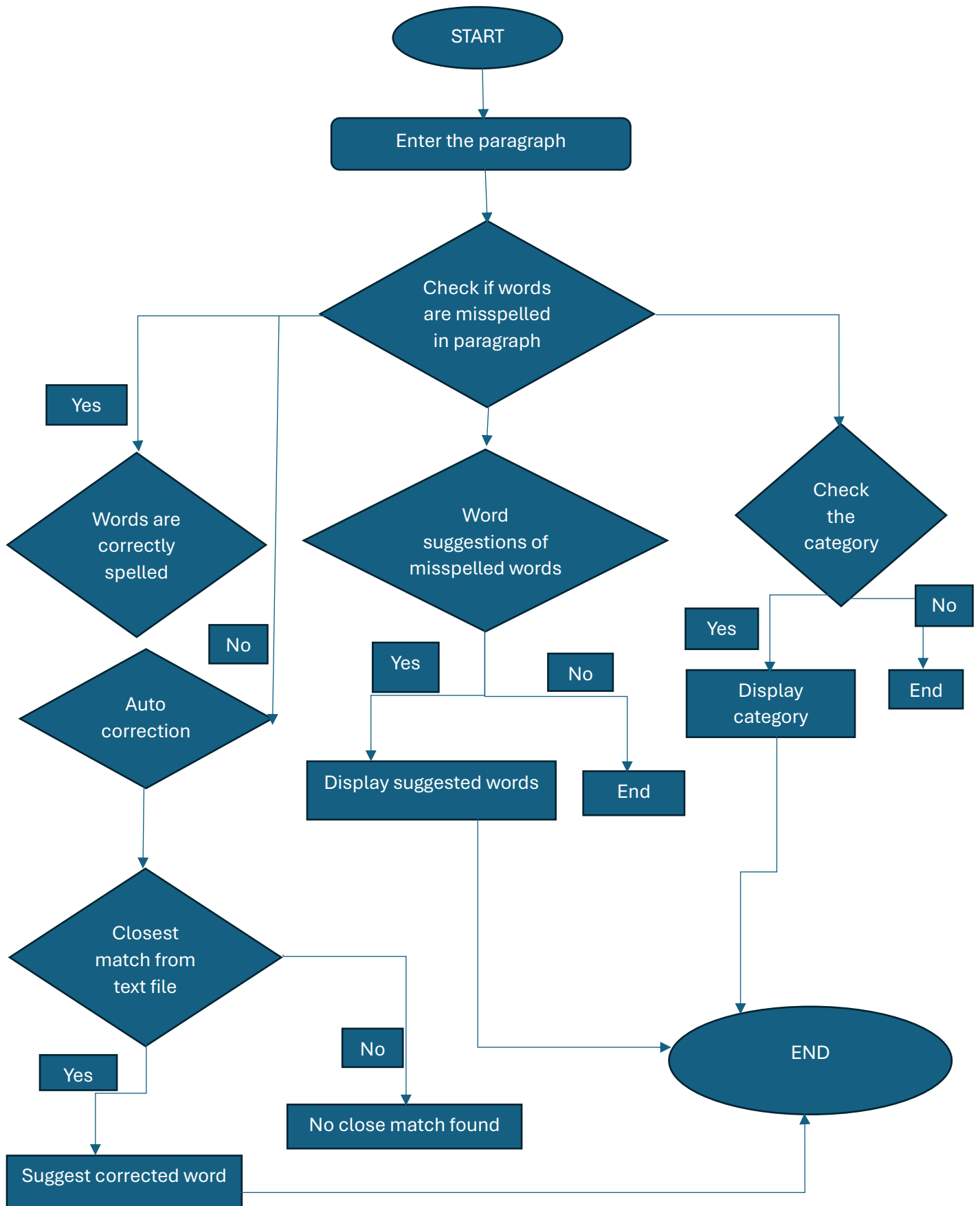
Spelling Corrections & Suggestions:

Corrected Paragraph:

Paragraph Category:

The below flow diagram explains that if we enter a paragraph from a dataset the prior thing is it will check the misspelled words in the paragraph, if the words are spelled incorrectly then it suggests three closest matched words. Then it displays the suggested words and finally it categorises the paragraph by finding some of the keywords like finance, technology, sport and sports, history, science, and politics

Flow chart:



Code:

```
import re
import sys
import tkinter as tk
from tkinter import filedialog, scrolledtext, messagebox
from difflib import get_close_matches
from collections import Counter

# Load dictionary from the text file
def load_dictionary(file_path):
    dictionary = set()
    try:
        with open(file_path, "r", encoding="utf-8") as file:
            for line in file:
                words = re.findall(r'\b[a-zA-Z]+\b', line.lower()) # Extract words
                dictionary.update(words)
        print(f'Dictionary Loaded: {len(dictionary)} words.')
    except FileNotFoundError:
        print("Error: Dictionary file not found.")
        sys.exit(1)
    return dictionary

# Suggest words for a misspelled word
def suggest_words(word, dictionary):
    suggestions = get_close_matches(word.lower(), dictionary, n=3, cutoff=0.7) # Top 3 suggestions
    return suggestions if suggestions else ["No suggestions found"]

# Auto-correct a word using the closest match from the dictionary
def auto_correct_word(word, dictionary):
```

```

if word.lower() in dictionary or any(char.isdigit() for char in word):
    return word # Keep numbers and correctly spelled words unchanged

suggestions = get_close_matches(word.lower(), dictionary, n=1, cutoff=0.8) # Best match
corrected_word = suggestions[0] if suggestions else word

if corrected_word != word:
    correction_text.insert(tk.END, f"Misspelled Word: {word} → Auto-corrected to: {corrected_word}\n")
    correction_text.insert(tk.END, f"Suggestions: {' '.join(suggest_words(word, dictionary))}\n\n")

return corrected_word

# Paragraph Categorization based on keywords
def categorize_paragraph(paragraph):
    categories = {
        "finance": {"money", "economy", "investment", "market", "bank", "stock", "financial"},
        "technology": {"computer", "software", "hardware", "AI", "technology", "internet", "cyber"},
        "sports": {"football", "cricket", "basketball", "tennis", "athlete", "olympics"},
        "history": {"war", "revolution", "historical", "empire", "ancient", "battle"},
        "science": {"physics", "chemistry", "biology", "science", "research", "experiment"},
        "politics": {"government", "election", "policy", "law", "democracy", "political"},
    }

    words = set(paragraph.lower().split()) # Convert paragraph to a set of words
    category_counts = {category: len(words & keywords) for category, keywords in categories.items()}

    best_category = max(category_counts, key=category_counts.get)

```

```

return best_category if category_counts[best_category] > 0 else "Uncategorized"

# Process the paragraph to correct spelling mistakes and suggest words
def process_paragraph():
    paragraph = input_text.get("1.0", tk.END).strip()

    if not paragraph:
        messagebox.showerror("Error", "Please enter a paragraph!")
        return

    corrected_paragraph = []
    correction_text.delete("1.0", tk.END) # Clear previous results

    # Split paragraph into words while preserving punctuation
    words = re.findall(r"[\w']+|[.,!?:\\""])", paragraph)

    for word in words:
        clean_word = re.sub(r"^[^\\w]", "", word) # Remove punctuation for checking
        corrected_word = auto_correct_word(clean_word, dictionary)
        corrected_paragraph.append(word.replace(clean_word, corrected_word)) # Maintain
punctuation

    corrected_text = " ".join(corrected_paragraph)
    category = categorize_paragraph(corrected_text) # Categorize the corrected paragraph

    corrected_paragraph_text.delete("1.0", tk.END)
    corrected_paragraph_text.insert(tk.END, corrected_text)

    category_label.config(text=f"Paragraph Category: {category}")

# GUI Setup

```

```
root = tk.Tk()
root.title("Spell Checker & Paragraph Categorizer")
root.geometry("800x600")
root.configure(bg="#f0f0f0")

# Load dictionary
dictionary_path = "C:\\Users\\Ganesh\\OneDrive\\Desktop\\paragraphs.txt" # Using the
uploaded file
dictionary = load_dictionary(dictionary_path)

# Input Text
tk.Label(root, text="Enter Paragraph:", font=("Arial", 12, "bold"),
bg="#f0f0f0").pack(anchor="w", padx=10, pady=5)
input_text = scrolledtext.ScrolledText(root, height=5, wrap=tk.WORD, font=("Arial", 12))
input_text.pack(fill="both", padx=10, pady=5)

# Process Button
process_button = tk.Button(root, text="Check & Categorize", font=("Arial", 12, "bold"),
command=process_paragraph, bg="#4CAF50", fg="white", padx=10, pady=5)
process_button.pack(pady=10)

# Correction Output
tk.Label(root, text="Spelling Corrections & Suggestions:", font=("Arial", 12, "bold"),
bg="#f0f0f0").pack(anchor="w", padx=10, pady=5)
correction_text = scrolledtext.ScrolledText(root, height=6, wrap=tk.WORD, font=("Arial",
12))
correction_text.pack(fill="both", padx=10, pady=5)

# Corrected Paragraph Output
tk.Label(root, text="Corrected Paragraph:", font=("Arial", 12, "bold"),
bg="#f0f0f0").pack(anchor="w", padx=10, pady=5)
```

```
corrected_paragraph_text = scrolledtext.ScrolledText(root, height=5, wrap=tk.WORD,  
font=("Arial", 12))
```

```
corrected_paragraph_text.pack(fill="both", padx=10, pady=5)
```

```
# Category Label
```

```
category_label = tk.Label(root, text="Paragraph Category: ", font=("Arial", 14, "bold"),  
fg="#333", bg="#f0f0f0")
```

```
category_label.pack(pady=10)
```

```
# Run GUI
```

```
root.mainloop()
```

Experimental Results:

1.

The screenshot shows a web application titled "Spell Checker & Paragraph Categorizer". It has a light gray background and a white text input area at the top. The input area contains the text: "This biographical article related to French artistic gymnastics is a stub. You can help Wikipedia by expanding it". Below the input area is a green button labeled "Check & Categorize". Underneath the button is a section titled "Spelling Corrections & Suggestions:". This section contains two lines of text: "Misspelled Word: article → Auto-corrected to: article" and "Suggestions: article, particle, articles". Below this is another line of text: "Misspelled Word: gymnastics → Auto-corrected to: gymnastics" and "Suggestions: gymnastics, gymnastic, gymnasts". Below the suggestions is a section titled "Corrected Paragraph:". This section contains the same text as the input area, but with the corrections applied: "This biographical article related to French artistic gymnastics is a stub . You can help Wikipedia by expanding it .". At the bottom of the application is a gray bar with the text "Paragraph Category: Uncategorized".

Explanation:

1. Here from the paragraph, we can see that the word spelled “artcle” is changed into “article” and it also gives some suggestions like article, articles, particle and it chooses nearest appropriate word.
2. Similarly, from the paragraph we can that the word spelled “gymnstics” is changed into “gymnastics” and it also gives some suggestions like gymnastic, gymnastics, gymnasts and chooses nearest appropriate word.
3. Based on the given categories it will check the word given in the defined categories.

2.

Spell Checker & Paragraph Categorizer

Enter Paragraph:

InsideAR was the largst Augmented Reality evnt in Europe. It was organized and supported by metaio GmbH every year. The first event was held in 2010, had since expanded globally and was run at multiple locations around the world. However, after Apple purchased metaio in May 2015, metaio cancelled the InsideAR conference 2015 without any statements about the conference's future.

Check & Categorize

Spelling Corrections & Suggestions:

Misspelled Word: largst → Auto-corrected to: largest
Suggestions: largest, largs, last

Misspelled Word: evnt → Auto-corrected to: event
Suggestions: event, levant, events

Corrected Paragraph:

InsideAR was the largest Augmented Reality event in Europe. It was organized and supported by metaio GmbH every year. The first event was held in 2010, had since expanded globally and was run at multiple locations around the world. However, after Apple purchased metaio in May 2015, metaio cancelled the InsideAR conference 2015 without any statements about the conference's future.

Paragraph Category: Uncategorized

Explanation:

1. Here from the paragraph, we can see that the word spelled “largst” is changed into “largest” and it also gives some suggestions like largest, larges, last and it chooses nearest appropriate word.
2. Similarly, from the paragraph we can that the word spelled “evnt” is changed into “event” and it also gives some suggestions like event, levant, events and chooses nearest appropriate word.
3. Based on the given categories it will check the word given in the defined categories.

3.

Spell Checker & Paragraph Categorizer

Enter Paragraph:

North Korea conducts fishing in its own EEZ, the extnt of which is unknown becaus North Korea has not passed a law on it, mainly for the industrial sector. Some fishing for the artisanal sector takes place, too.

Check & Categorize

Spelling Corrections & Suggestions:

Misspelled Word: extnt → Auto-corrected to: extent
Suggestions: extent, extant, extinct

Misspelled Word: becaus → Auto-corrected to: because
Suggestions: because, beau, belarus

Corrected Paragraph:

North Korea conducts fishing in its own EEZ , the extent of which is unknown because North Korea has not passed a law on it , mainly for the industrial sector . Some fishing for the artisanal sector takes place , too .

Paragraph Category: politics

Explanation:

1. Here from the paragraph, we can see that the word spelled “extnt” is changed into “extent” and it also gives some suggestions like extent, extant, extinct and it chooses nearest appropriate word.
2. Similarly, from the paragraph we can that the word spelled “becaus” is changed into “because” and it also gives some suggestions like because, beau, belarus and chooses nearest appropriate word.
3. Based on the given categories it will check the word given in the defined categories.

4.

Spell Checker & Paragraph Categorizer

Enter Paragraph:

The first joint ventre North Korea estblshd with China, in 1989, was a maine fishey proucts firm locatd in Chongjin that had an initial capitalization of US\$1 million.

Check & Categorize

Spelling Corrections & Suggestions:

Misspelled Word: ventre → Auto-corrected to: venture
Suggestions: venture, venstre, ventures

Misspelled Word: estblshd → Auto-corrected to: established
Suggestions: established, establish, reestablished

Corrected Paragraph:

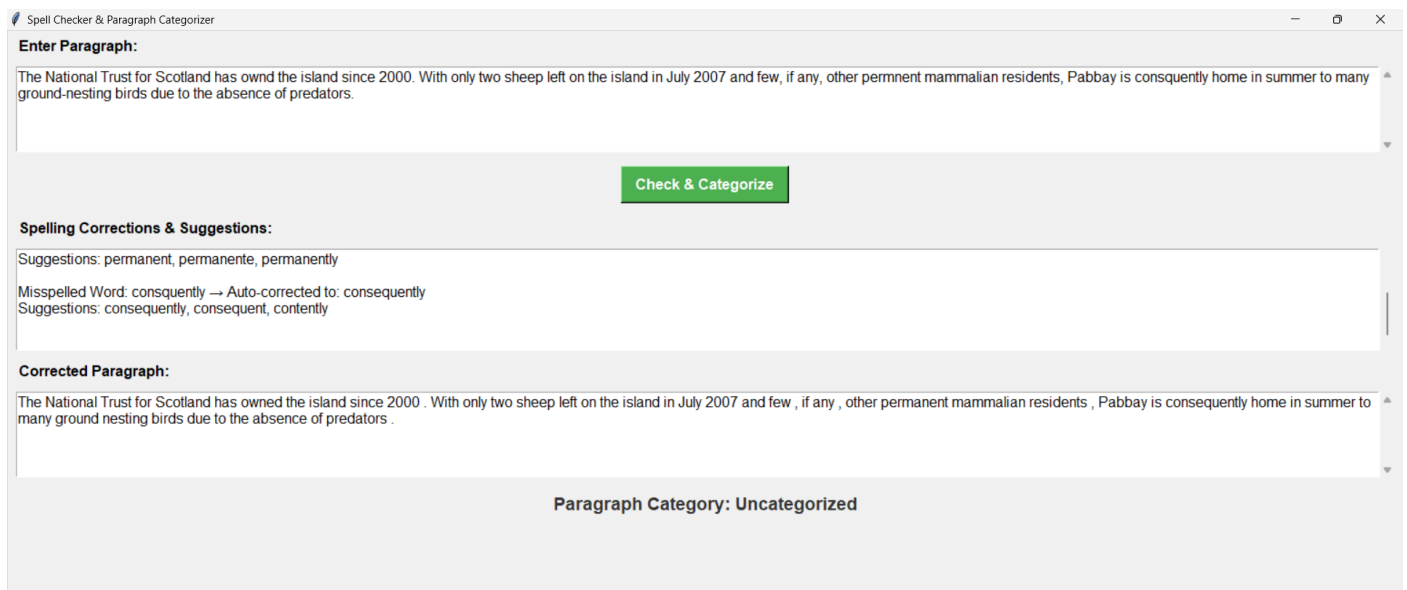
The first joint venture North Korea established with China , in 1989 , was a maine fishery products firm located in Chongjin that had an initial capitalization of US 1 million .

Paragraph Category: Uncategorized

Explanation:

1. Here from the paragraph, we can see that the word spelled “ventre” is changed into “venture” and it also gives some suggestions like venture, venstre, ventures and it chooses nearest appropriate word.
2. Similarly, from the paragraph we can that the word spelled “estblshd” is changed into “established” and it also gives some suggestions like established, establish, reestablished and chooses nearest appropriate word.
3. Similarly, from the paragraph we can that the word spelled “fishey” is changed into “fishery” and it also gives some suggestions like fishery, fishy, fishes and chooses nearest appropriate word.
4. Similarly, from the paragraph we can that the word spelled “proucts” is changed into “products” and it also gives some suggestions like products, product, prout and chooses nearest appropriate word.
5. Similarly, from the paragraph we can that the word spelled “locatd” is changed into “located” and it also gives some suggestions like located, locate, relocated and chooses nearest appropriate word.
6. Based on the given categories it will check the word given in the defined categories.

5.



Spell Checker & Paragraph Categorizer

Enter Paragraph:

The National Trust for Scotland has ownd the island since 2000. With only two sheep left on the island in July 2007 and few, if any, other permnent mammalian residents, Pabbay is consquently home in summer to many ground-nesting birds due to the absense of predators.

Check & Categorize

Spelling Corrections & Suggestions:

Suggestions: permanent, permanente, permanently

Misspelled Word: consquently → Auto-corrected to: consequently

Suggestions: consequently, consequent, contently

Corrected Paragraph:

The National Trust for Scotland has owned the island since 2000 . With only two sheep left on the island in July 2007 and few , if any , other permanent mammalian residents , Pabbay is consequently home in summer to many ground nesting birds due to the absence of predators .

Paragraph Category: Uncategorized

Explanation:

1. Here from the paragraph, we can see that the word spelled “ownd” is changed into “owned” and it also gives some suggestions like owned, own, wynd and it chooses nearest appropriate word.
2. Similarly, from the paragraph we can that the word spelled “permnent” is changed into “permanent” and it also gives some suggestions like permanent, permanente, permanently and chooses nearest appropriate word.
3. Similarly, from the paragraph we can that the word spelled “consquently” is changed into “consequently” and it also gives some suggestions like consequently, consequent, contently and chooses nearest appropriate word.
4. Based on the given categories it will check the word given in the defined categories.

6.

Spell Checker & Paragraph Categorizer

Enter Paragraph:

Due to its adption of the Proletarian Military Policy, the WIL argued that its members should go through the experience of the war with other membrs of their class by joining the army when called-up. But if this was applied to the whole membership it meant they could be disprsed and provide no real leadership and therefore the organisation took measures to presrve the leading cadres outside the forces.

Check & Categorize

Spelling Corrections & Suggestions:

Misspelled Word: adption → Auto-corrected to: adoption
Suggestions: adoption, adsorption, adaptation

Misspelled Word: membrs → Auto-corrected to: members
Suggestions: members, member, remembers

Corrected Paragraph:

Due to its adoption of the Proletarian Military Policy , the WIL argued that its members should go through the experience of the war with other members of their class by joining the army when called up . But if this was applied to the whole membership it meant they could be dispersed and provide no real leadership and therefore the organisation took measures to preserve the leading cadres outside the forces .

Paragraph Category: history

Explanation:

1. Here from the paragraph, we can see that the word spelled “adption” is changed into “adoption” and it also gives some suggestions like adoption, adsorption, adaptation and it chooses nearest appropriate word.
2. Similarly, from the paragraph we can that the word spelled “membrs” is changed into “members” and it also gives some suggestions like members, member, remembers and chooses nearest appropriate word.
3. Similarly, it also changes the words like “disprsed” to dispersed and “presrve” to preserve.
4. Based on the given categories it will check the word given in the defined categories.

Conclusion:

This Python script is a useful spell checker and paragraph classifier with an intuitive Tkinter GUI. It accurately identifies and fixes spelling errors through dictionary-based detection and provides word suggestions for enhancement. It also classifies paragraphs according to keyword matching, useful for text analysis.

This software can be further improved with the incorporation of machine learning for more intelligent categorization, dictionary expansion for enhanced accuracy, or real-time spell-checking. It is a useful solution in general for text quality and organization improvement.

References:

- 1.Brill, E., & Moore, R. C. (2000). An improved error model for noisy channel spelling correction. *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, 286–293.
- 2.Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- 3.Jurafsky, D., & Martin, J. H. (2021). *Speech and language processing* (3rd ed.). Pearson.
- 4.Kim, Y. (2014). Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.
- 5.Kukich, K. (1992). Techniques for automatically correcting words in text. *ACM Computing Surveys (CSUR)*, 24(4), 377–439.
- 6.Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. *International Conference on Machine Learning*, 1188–1196.
- 7.Sakaguchi, K., Post, M., & Van Durme, B. (2017). Grammatical error correction with neural reinforcement learning. *arXiv preprint arXiv:1707.09127*.
- 8.Schuster, T., Ram, O., Barzilay, R., & Globerson, A. (2019). Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 1599–1613.
- 9.Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 5998–6008.
- 10.Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems (NeurIPS)*, 3111–3119.
- 11.Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- 12.Church, K. W., & Gale, W. A. (1991). Probability scoring for spelling correction. *Statistics and Computing*, 1(2), 93–103.

13. Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to information retrieval*. Cambridge University Press.
14. Brown, P. F., Della Pietra, V. J., Mercer, R. L., Della Pietra, S. A., & Lai, J. C. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2), 263–311.
15. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.
16. Dataset for the project from Kaggle
<https://www.kaggle.com/datasets/nikitricky/random-paragraphs>