

Link to forked repository-

https://github.com/muktabhumkar/Grant-Writing-Support-Tool/blob/main/error_detection.ipynb

Approach: Steps Taken to Analyse and Implement the Task

1. Analysis of the Problem:

- The task involves creating an error detection feature for grant proposals. The goal was to identify common issues such as inconsistent language, unclear objectives, and missing sections within a grant proposal.

2. Identifying Key Areas to Analyse:

- I identified three key areas where errors are commonly found in grant proposals:
 1. **Inconsistent Language:** Using different terms for the same concept, which can confuse readers.
 2. **Unclear Objectives:** Objectives that are vague or unclear, making it difficult for reviewers to understand the goals of the proposal.
 3. **Missing Information:** Essential sections such as budget, timeline, and goals that may be missing in the proposal.

3. Implementation Plan:

- To detect these issues, I used a combination of Natural Language Processing (NLP) techniques and simple heuristics. Specifically:
 - **Inconsistent Language:** I analysed the frequency of key terms and flagged overuse or inconsistent use of related terms.
 - **Unclear Objectives:** I used a list of vague words and phrases (e.g., "improve", "increase") to identify unclear objectives.
 - **Missing Information:** I checked for the presence of critical sections like budget, timeline, and objectives.

4. Development:

- I created a Python script that uses spaCy for NLP tasks and NLTK for sentence tokenization to process the text. I defined functions to check for each of the error categories and implemented a main function to compile the results.

AI Tools Used:

- **spaCy:**
 - **Purpose:** Used for tokenizing the input text and identifying key terms for the analysis. It also helped in part-of-speech tagging, which was used for detecting inconsistencies in language.
 - **Application:** Loaded the pre-trained English model (en_core_web_sm) to process and analyse the text.

- **NLTK:**
 - **Purpose:** Utilized for sentence tokenization to break down the grant proposal into smaller parts for more granular analysis.
 - **Application:** The `sent_tokenize` function from NLTK was used to split the text into individual sentences and check for unclear objectives based on specific phrases.
- **Regular Expressions (re):**
 - **Purpose:** Used for pattern matching and identifying whether certain key sections (e.g., "budget", "timeline") were mentioned in the proposal.
 - **Application:** Applied regular expressions to search for these terms and check if the content was missing.

Challenges and Learnings:

1. **Challenge:** Missing NLTK Resources
 - The initial error was due to missing the NLTK resource for sentence tokenization (punkt), which caused the script to fail.
 - **Solution:** I resolved this by downloading the punkt resource using `nltk.download('punkt')` and ensuring all required components were available before running the script.
2. **Challenge:** Handling Vague Objectives
 - It was tricky to define a comprehensive list of vague phrases that would consistently identify unclear objectives. There's no one-size-fits-all solution for identifying vague goals in grant proposals.
 - **Solution:** I used a set of common vague words as a starting point (e.g., "improve", "better"), but this can be expanded based on the domain of the grant proposals or customized by the user for better accuracy.
3. **Learning:**
 - The importance of NLP in analysing textual data was reinforced, and I learned how to apply simple NLP techniques, such as tokenization and part-of-speech tagging, to process and analyse text effectively.
 - Also, combining simple heuristics (like checking for missing sections) with NLP tools provided a straightforward yet effective method for detecting errors in the grant proposal.