

Project Description

For my final project for 15-112, I would like to implement a simple text editor using python, where the user can type a series of characters and sentences. My text editor will feature a real-time English spell-checker that provides a list of word suggestions (3-5 suggestions) that the incorrectly-spelt word is likely to be, or alternatively, auto-corrects that word for you (depending on the user's chosen setting). To implement this, I will implement an algorithm for calculating the "Minimum edit distance", or the number of changes required to change the user's misspelt word to a correctly-spelt word in my stored dictionary, as well as considering word frequency for instances of words with equal minimum edit distances. The option of adding an unfamiliar/misspelt word to the dictionary will also be available.

The project will also contain manual word-complete which will continuously be running as the user is typing a word, updating the list of complete word suggestions as each new character is typed. The word-complete will be implemented using string processing and comparison of the typed string to list of correct words stored internally in a dictionary, and considering word frequency to create more accurate suggestions.

The last main feature of my text editor will be the ability to locally create and save the typed content as files, as well as open and modify local ".txt" files. I will also implement file tabs, in which the user can open multiple files (in separate tabs), navigate through them seamlessly without losing work progress, and transfer data between them using copy/paste and cut/paste shortcuts.

The interface will mainly be toolbar for options, word markup for errors, file tabs and pop-up suggestions lists (more detailed description later in 'user interface').

Some additional features for better user experience may be implemented in the final project, such as auto-capitalization, recently opened files history, interface customization (font, size, colors, style, etc), word lookup, synonyms, definitions, opening different file types (e.g. .csv files) and document edit history.

Libraries and Main Features:

The main features of the text editor as previously described are:

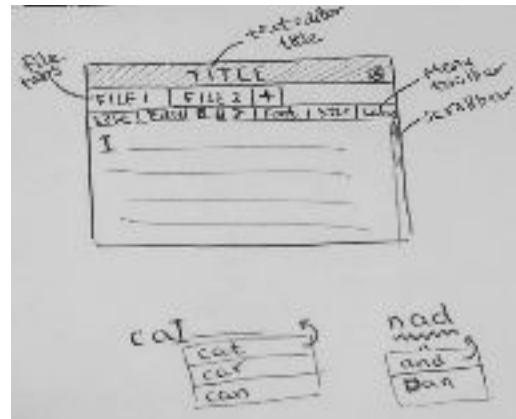
- Word-complete (updating list of word suggestions)
- Spell-checker/Auto-correct (by finding Min Edit Distance)
- File Editing/Navigation
- Markup

Main libraries that will be used to complete this project are:

- Standard Python Library (for string and file processing)
- Tkinter (for interface and interaction with user)
- TkFileDialog (saving files)

User Interface

The interface will look similar to the diagram on the right, with file tabs at the very top, the menu toolbar right under them (for customization (additional), opening local files, saving, copying/pasting) and the main typing space underneath. There will also be a scrollbar to navigate through longer text files and markup of pop-up lists for autocomplete and autocorrect and underlining for misspellings as shown.



Features for Milestone Submission (Nov 24):

-Word-complete:

- The program generates a list of word suggestions as a part of a word is being typed.
- The list of words updates as a new character is typed

-Autocorrect/Manual-correct:

- The program generates a list of word suggestions for each word typed incorrectly/ one it doesn't find in the internal dictionary of correct English words
- List of word suggestions is mainly based on minimum edit distance algorithm, with words of a smaller edit distance appearing at the top of the list in ascending order downwards.
- Option to auto-correct: the program automatically replaces the incorrect word with the top word in the list of word suggestions
- Spell-checker implemented on each word right after it is typed (i.e. it runs on the last word typed after each space is typed)

-Basic interface

- Pop up lists of word suggestions for word-complete + manual word-correct (as in diagram)
- Interacting with lists of suggestions to incorporate them into writing
 - (E.g. clicking on the desired word in the word list replaces the original word with the correct one from the list of word suggestions)
- Text editor typer where user types and the spell-checker and word-completer program runs
- Scrollbar to navigate through long documents
- Word markup for misspelled words (underlining)
- Menu-bar checkbox for autocorrect option

(check project description for further detail, if needed)

Additional Features for Final Submission (Dec 2):

- File tabs interface + opening local files + saving new files
 - Ability to open multiple local files at once, navigate between them without losing work progress, create new files within the program itself, save the files
- copy/pasting between file tabs
 - Ability to copy, cut and paste between the opened files in the file tabs

- Tool Menubar functionality (including open/save/copy/paste options)
 - Buttons for creating, opening and saving files locally under “File” menubar
 - Buttons for copying, cutting and pasting functionalities between files under “Edit” menubar

(check project description for further detail, if needed)

Complete Set of Features in Final Submission (Dec 2):

-Word-complete:

- The program generates a list of word suggestions as a part of a word is being typed.
- The list of words updates as a new character is typed

-Autocorrect/Manual-correct:

- The program generates a list of word suggestions for each word typed incorrectly/ one it doesn't find in the internal dictionary of correct English words
- List of word suggestions is mainly based on minimum edit distance algorithm, with words of a smaller edit distance appearing at the top of the list in ascending order downwards.
- Option to auto-correct: the program automatically replaces the incorrect word with the top word in the list of word suggestions
- Spell-checker implemented on each word right after it is typed (i.e. it runs on the last word typed after each space is typed)

-Basic interface

- Pop up lists of word suggestions for word-complete + manual word-correct (as in diagram)
- Interacting with lists of suggestions to incorporate them into writing
 - (E.g. clicking on the desired word in the word list replaces the original word with the correct one from the list of word suggestions)
- Text editor typer where user types and the spell-checker and word-completer program runs
- Scrollbar to navigate through long documents
- Word markup for misspelled words (underlining)
- Menu-bar checkbox for autocorrect option
 - File tabs interface + opening local files + saving new files
 - Ability to open multiple local files at once, navigate between them without losing work progress, create new files within the program itself, save the files
 - copy/pasting between file tabs
 - Ability to copy, cut and paste between the opened files in the file tabs
 - Tool Menubar functionality (including open/save/copy/paste options)
 - Buttons for creating, opening and saving files locally under “File” menubar
 - Buttons for copying, cutting and pasting functionalities between files under “Edit” menubar

(check project description for further detail, if needed)