



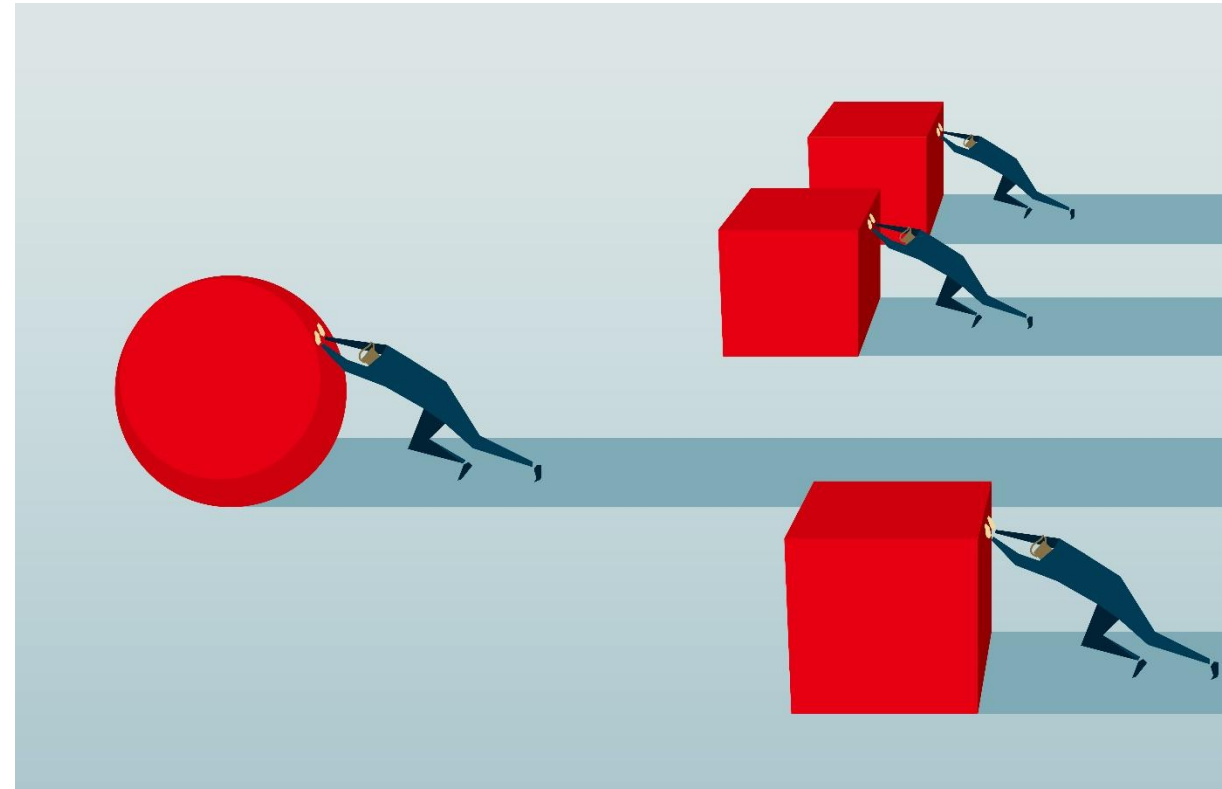
AUTOCORRECT

ESC101 PROJECT

BY- UMANG MALIK

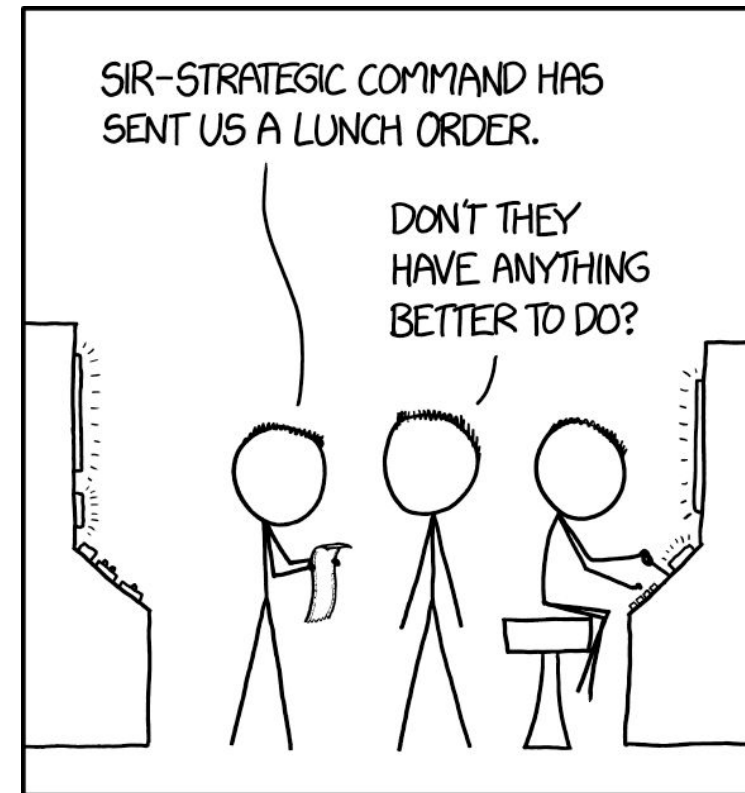
The broad problem

- ▶ A good Autocorrect is of prime importance in today's world.
- ▶ Useful for everyone, from authors to bloggers to anyone you can think of.
- ▶ An amazing tool for saving time and effort.



But...

- ▶ A satisfactory autocorrect is hard to find.
- ▶ Wrong corrections mess things up, a lot.



EVERYONE COMPLAINS ABOUT AUTOCORRECT,
BUT WE FORGET ABOUT THE TIME IT
PREVENTED A NUCLEAR WAR.

Scope of work

- ▶ Project was divided in phases-
 - ▶ First phase- Develop a spelling corrector.
 - ▶ Second phase- Develop a basic model of Autocorrect based on the developed spelling corrector.
 - ▶ Third phase- Develop a GUI and add various other features.
- ▶ Improvements on the error model were made throughout the duration of the project.
- ▶ Model based on errors during typing.

Working on the software-

- ▶ Process a large collection of text, previously typed data, and personal dictionary.
- ▶ Take in the input during runtime.
- ▶ Propose the top 3 corrections if the word is wrong.
- ▶ Person chooses what he wants. If he doesn't, most probable word chosen.

Work done so far

- ▶ The error model has been developed and furnished.
 - ▶ Based on the frequency of words in the a big dataset of text files.
- ▶ The Autocorrect based on the model has been implemented.
- ▶ The GUI for the Autocorrect has been developed.



The screenshot shows a window titled "AutoCorrect" with a yellow background. At the top, there is a green header bar with the text "ENTER THE SENTENCE" in bold. Below this is a large white text area for input. To the right of the text area, there is a "Suggestions" section with three input fields labeled "option1", "option2", and "option3", and a "correct" button. At the bottom, there is a section for adding words to the dictionary, with a "Want to add a word to Dictionary?" label, an input field, and an "Add to Dictionary" button. Below this, there is a section for adding text replacements, with a label "Add Text Replacements-", a "Replace" input field, a "with" input field, and an "Add" button.

Work done so far

- ▶ Various other features have been added, namely-
 - ▶ Placement of keys on the keyboard.
 - ▶ Personal dictionary
 - ▶ Self improving model on the basis of typing habits
 - ▶ Text Replacement for abbreviations etc.
 - ▶ Alternative corrections

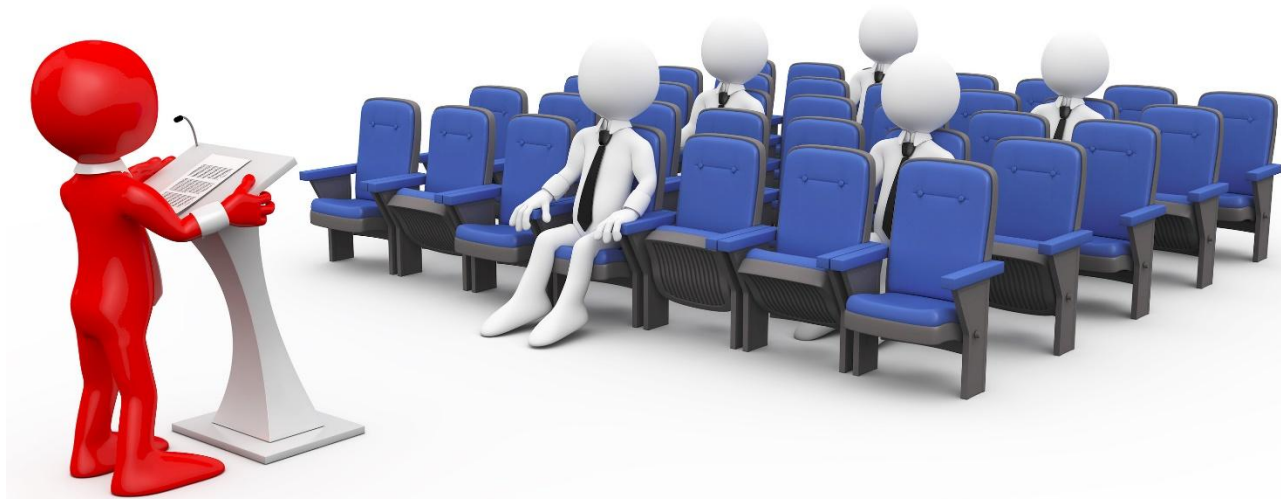
ESC101 Concepts used

- ▶ Working with strings
- ▶ Structures and classes
- ▶ Pointers and iterators
- ▶ Data structures like map/vector
- ▶ File i/o
- ▶ Arrays, loops, functions etc.

Testing of the software

- ▶ Model tested against standard data from GNU aspell and Wikipedia.
- ▶ Software tested by friends, mentor and other students.
- ▶ For a test sample consisting of 4000+ words, ~75% accuracy yielded at 15 words per second.
- ▶ Accuracy can be improved by collecting more data.

Demonstration



Issues faced

- ▶ Error Model
- ▶ Development of GUI
- ▶ Implementation of various features

Future Work

- ▶ Incorporating the autocorrect into various applications through plug-ins, extensions etc.
- ▶ Contest sensitive model required if accuracy over 85% needed.

Conclusion

- ▶ Had an experience of software development
- ▶ Learned to work in a phase-by-phase manner
- ▶ Strengthened coding skills and learned a plethora of new things about C++