

ABSTRACT

Natural language processing (NLP) is a field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages. Many challenges in NLP involve natural language understanding, that is, enabling computers to derive meaning from the human or natural language input. An important application of NLP is a Text Processor or Editor. Many well-known Text Editors already exist in the market, like Gedit, Notepad, Leafpad, MS Word, TexShop, AkelPad, etc.

The “Integrated Text Editor” is a simple live text editor that is intended to implement few features that most text/word editing processors (E.g. MS-Word) do not encompass. Some of the features include Spell checking and auto correction, Word Segmentation, Auto Complete, Regex Find, Topic extraction, etc. All these features are the work of crafty algorithms, and neat ideas to explore speed and user friendliness. The spellchecker continuously scans for misspelt words. The AutoCorrect feature, quickly changes correctable words, to the most appropriate word. This saves lot of time and unwanted revisits. Word Segmentation splits two words automatically, if the user forgets to put spaces between them. Autocomplete feature allows the user to type big essays in quick time. The regular expression find option is a very versatile search option for the users. The Topic Extraction algorithms tries to learn from the text that is being typed, and provides quick Google search results on the topic. It is an innovative concept that involves recognizing the theme/context that the user is currently attempting to type. On recognition, search links are provided to the user, to enable the user in easily searching for additional information on text being typed.

The GUI of the text-editor has been implemented using Java Swing. The editor is kept simple, without a lot of fancy options to annoy the user. Spell checking involves three concepts of non-word error detection, isolated word correction and word segmentation. The Bayesian Model, Levenshtein’s Edit Distance Algorithm, etc. are some of the algorithms that are used for the purpose. The Auto complete feature is implemented using efficient data structures, and search mechanisms. Topic Extraction has been implemented using simple tag analysis, and the concepts of keyword extraction. Dynamic find uses the Java regex class.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without the mention of the people who made it possible. Many are responsible for the knowledge and experience that we have gained during our project and throughout the course. Hence, we feel that expressing our deepest gratitude is just not formality but a part and parcel of the project.

We would also like to express our gratitude towards our honourable principal **Dr. M.K. Venkatesha** for facilitating all of us to pursue a project as per our choice and giving us all the inspiration and support.

We are highly indebted to **Dr. G T Raju**, Professor and Head, Dept. of CSE, for his consent and wholehearted cooperation in providing all the facilities and resources that we had required for successful implementation of this project.

We would first like to express our earnest thanks towards our project guide, **B I Khodandpur**, Asst. Professor, Dept of CSE. He is the motivator, guide and constant source of knowledge and inspiration for us towards the preparation of this project. We would also like to thank **Mr. Devaraju B. M.**, Asst. Professor, Dept. of CSE for his encouragement and support throughout this project work.

Last but not the least, we thank all our friends who helped us directly or indirectly during this project and made it successful. At the same time, we thank all our faculty and lab assistants of the Computer Science and Engineering Dept., for their kind co-operation.

Tushar K Naik

Vamanan T S

Suhas V