WEB SEARCH AND MINING

PROJECT REPORT
DESIGN AND IMPLEMENT
HINDI SPELL CHECKER

Guided By : Dr. Rakesh Chandra Balabantaray

By Group 8 :

Preeti Dey(B118040)

Pritish Priyatosh Nayak(B118041)

Priya Pradeep Kumar(B118042)

Priyabrat Mahapatro(B118043)

ABSTRACT

Spell checker is one of the most common features that are used across text editors and search engines that are regularly used by people.

Normally a spell correction technique consists of two process :

1. **Error detection** : The first part consists of identifying the errors in the typed text. This part uses a language model which accounts for the words allowed in the language.

2. **Error Correction**: This involves rectifying the wrong spelled words using a set of possibilities from the dictionary.
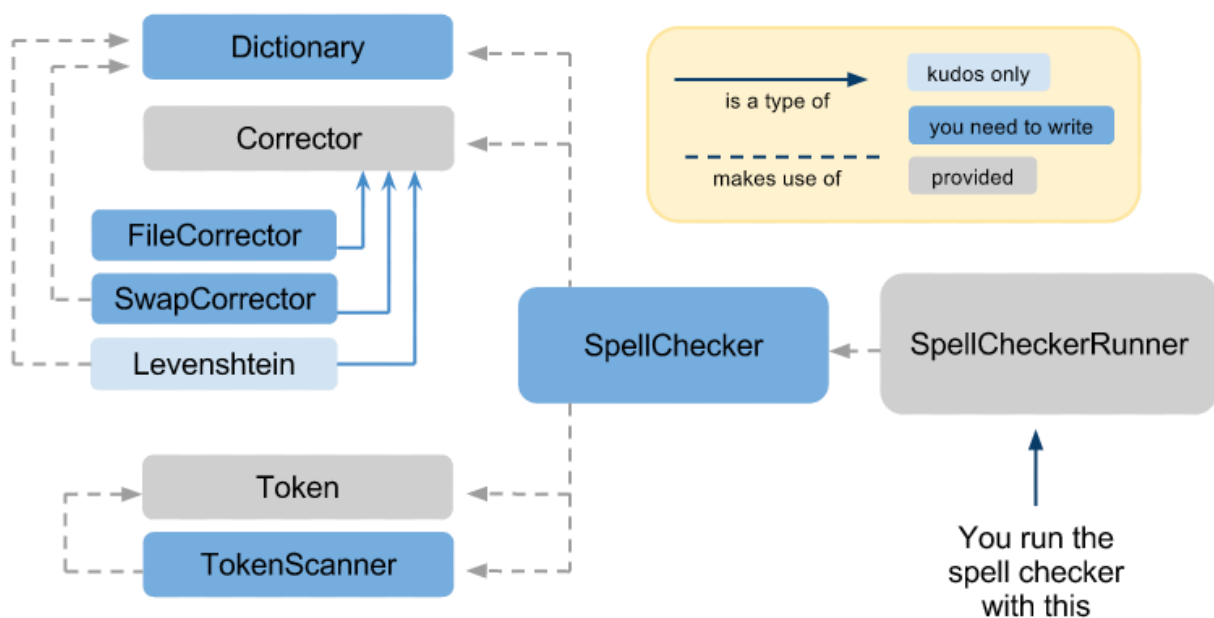

Figure 1: Diagram represting working of a simple Spell Checker.

In this project, we have built a simple spell checker for **Hindi language**.

Method used is : **Edit distance based.**
Algorithm used is : **Levenshtein distance.**

LEVENSHTEIN DISTANCE :

It is one of the most popular algorithms to calculate edit distance. It is named after the Soviet mathematician Vladimir Levenshtein, who considered this distance in 1965.

The main work of this algorithm is –given two words(strings) ,it calculates the minimum single character changes required to convert one of the word to another.

The three basic operations involved in this algorithm are:
1. Insert
2. Delete
3. Replace

It is a Dynamic Programming based algorithm whose complexity is **O(**|s1||s2|**)** ,where |s1| and |s2| denote the lengths of the two input strings.

CODE IMPLEMENTATION DETAILS:

This project uses Python 3.7.0 as the programming language.
External Libraries Used : None

1. Our corpus consists of nearly 20,000 Hindi words which were extracted from news articles. The source to the full corpus can be found : <u>Here.</u>

2. Our python program named spell_checker.py loads this corpus (file: hindi_corpus.txt) into memory and stores it in a global list (array).

3. The program then loads the input (file : input.txt) which contains the Hindi sentenses which are to be checked.

4. The program then checks for each word if it is present directly in the current loaded corpus. If it is present then there's no need to correct this word. Otherwise we start searching for the correct word.

5. For searching the correct word , we simply loop through all the words in the corpus and look for word with the minimum edit distance. In case of words having equal edit distance, we chose the one which is present earlier in the corpus (the corpus was pre-sorted according to the frequency of occurance of each word).

6. Upon finding the possible match for the correct word, we simply print the location of the mispelled word along with the corrected version of the word.

Note: Further optimization heuristics for speeding up the query time were not implemented as it was somewhat beyond the scope of the project.

Input Format:
Paste the Hindi string into "input.txt" in the root directory of the project.

Output Format
The program will create a "output.txt" file in the root directory of the project where for each wrong word a line will be printed in the format:

"At Line:" <line_number> "Word No. " <word_number> ":" <wrong_word> "->" <correct_word>

Project Source Code Link :
https://github.com/pritish-n/Hindi-Spell-Checker

External Resources used :
- Theory :
  - https://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf
  - https://en.wikipedia.org/wiki/Levenshtein_distance
- Corpus Source : https://wortschatz.uni-leipzig.de/en/download/hindi