**Basic concepts**

A basic spell checker carries out the following processes:

* It compares the word with a known list of correctly spelled words (i.e. a dictionary). This might contain just a list of words, or it might also contain additional information, such as hyphenation points or lexical and grammatical attributes.
* An additional step is a language-dependent algorithm for handling [morphology](https://en.wikipedia.org/wiki/Morphology_(linguistics)). Even for a lightly inflected language like [English](https://en.wikipedia.org/wiki/English_language), the spell checker will need to consider different forms of the same word, such as plurals, verbal forms, [contractions](https://en.wikipedia.org/wiki/Contraction_(grammar)), and [possessives](https://en.wikipedia.org/wiki/Possessive_(linguistics)). For many other languages, such as those featuring agglutination and more complex declension and conjugation, this part of the process is more complicated.

Spell checkers can use [approximate string matching](https://en.wikipedia.org/wiki/Approximate_string_matching) algorithms such as [Levenshtein distance](https://en.wikipedia.org/wiki/Levenshtein_distance" \o "Levenshtein distance) to find correct spellings of misspelled words. An alternative type of spell checker uses solely statistical information, such as [n-grams](https://en.wikipedia.org/wiki/N-gram), to recognize errors instead of correctly-spelled words.

Levenshtein distance:

Levenshtein distance is a [string metric](https://en.wikipedia.org/wiki/String_metric) for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.

[N-grams](https://en.wikipedia.org/wiki/N-gram):

N-grams are continuous sequences of words or symbols or tokens in a document. In technical terms, they can be defined as the neighbouring sequences of items in a document.

Textblob:

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

Word:

**Word()** is a simple word representation from the textblob library which has many useful methods, especially for correcting the spelling.

Tkinter:

Tkinter is the de facto way in Python to create [Graphical User interfaces (GUIs)](https://www.activestate.com/blog/top-10-python-gui-frameworks-compared/) and is included in all standard Python Distributions. In fact, it’s the only framework built into the Python standard library.

This Python framework provides an interface to the Tk toolkit and works as a thin object-oriented layer on top of Tk. The Tk toolkit is a cross-platform collection of  ‘graphical control elements’, aka widgets, for building application interfaces.

This framework provides Python users with a simple way to create GUI elements using the widgets found in the Tk toolkit. Tk widgets can be used to construct buttons, menus, data fields, etc. in a Python application. Once created, these graphical elements can be associated with or interact with features, functionality, methods, data or even other widgets.

For example, a button widget can accept mouse clicks, and can also be programmed to perform some kind of action, such as exiting the application.

**6. Implementations**

**6.1 Methodology**

Here we have used Tkinter module to create a GUI to get the input for user input data with a submit button to execute a function which give us the correct text using the TextBlob module from the textblob library.

Function from tkinter library such as Tk(), Label(), Entry(), Button(), etc are used to create the GUI needed for the Spell Checker project.

Get function is used to extract the text from the Entry box in the checkSpelling() function.

The function then sends the data received from the get function ie. The user input and sends it for pre-processing to the TextBlob module and use the .correct() function from the textblob library to get the corrected output.

The .set() function is then used to display the output.

**6.2 Testing**

from tkinter import \*

from textblob import TextBlob

def checkSpelling():

    word = text.get()

    nword = TextBlob(word)

    result = nword.correct()

    correctedText.set("Correct word: "+str(result))

master = Tk()

master.title("NLP Project")

master.geometry('750x250')

master.config(bg='Light Blue')

text=StringVar(master)

correctedText =StringVar(master)

Label(master, text='Spell Checker',font=("Times",20,"bold"), bg="red", width=90, relief="solid", bd=2, fg='black').pack(side=TOP, fill=X)

Label(master, text='Please enter the word:',bg='Light Blue',font=('calibre',13,'normal'), anchor="e", justify=LEFT).place(x=100, y=70)

Entry(master,textvariable=text, width=35,font=('calibre',13,'normal')).place(x=280,y=70)

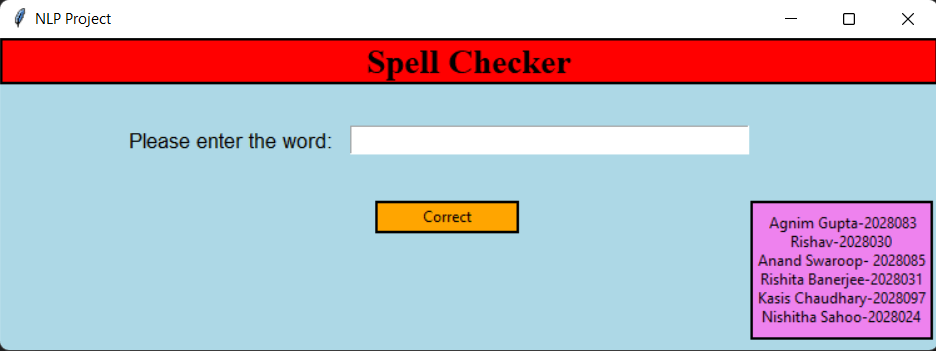
opLabel = Label(master, textvariable=correctedText, bg='Light Blue',anchor="e",font=('calibre',13,'normal'), justify=LEFT).place(x=280, y=100)

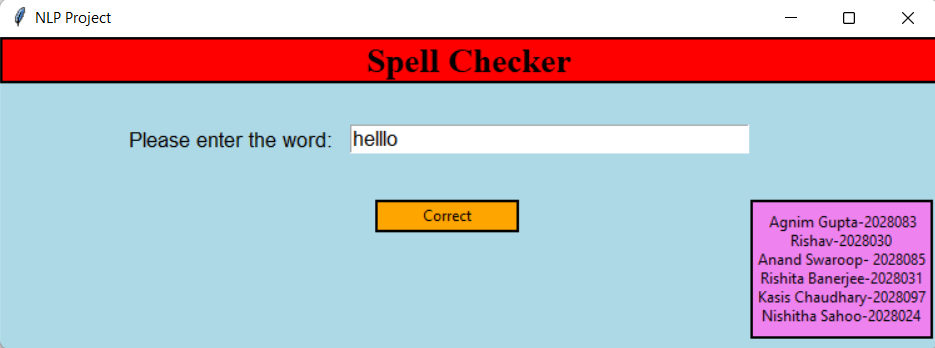
Button(master, height=1, width=15, text="Correct", relief="solid", bd=2, bg="orange",command=checkSpelling).place(x=300, y=130)

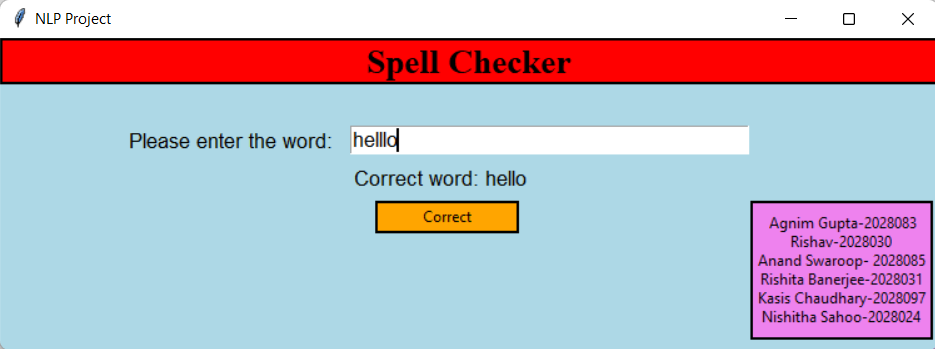
tablem4 = Label(master, text=" Agnim Gupta-2028083\nRishav-2028030\nAnand Swaroop- 2028085\nRishita Banerjee-2028031\nKasis Chaudhary-2028097\nNishitha Sahoo-2028024", width=20, height=7, bg="violet", anchor=CENTER, relief="solid").place(x=600,y=130)

master.mainloop()

**6.3 Result Analysis**







**Conclusion and Future Scope**

We started from very simple model and iteratively increase it’s capabilities, and finally we get a strong, production level spell checker. Still, it’s not the end, there is a lot of steps on the long road to our goal — making the best possible spell checker in the world.

Spell checker has a lot of scope in the future, as a good spell checker will gradually improve the writing skills and speed of every person whosoever uses it. It will be very useful in writing books, newspaper, articles and many other things. It will make the job of writer much easier

**References**

<https://medium.com/spark-nlp/applying-context-aware-spell-checking-in-spark-nlp-3c29c46963bc>

**Individual Contribution**