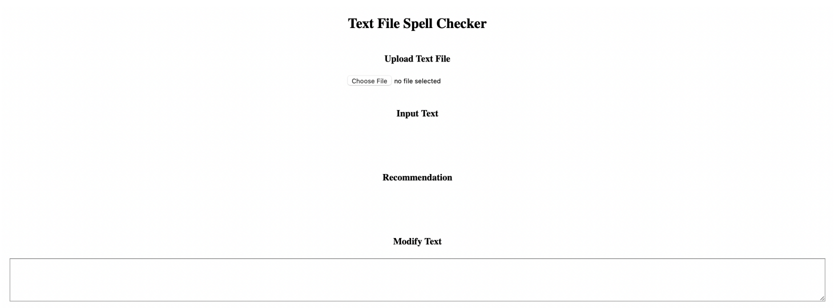
CS459-MP3-Report

Jacob Desilets, Serin Yoon

April 2022

1 Text file spell checker

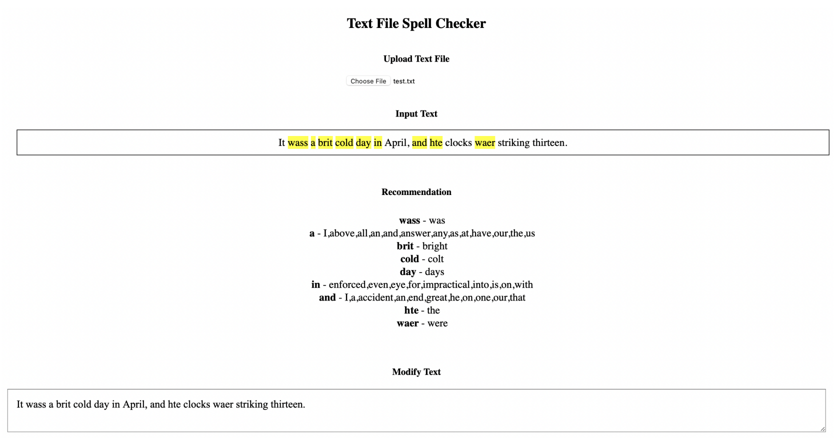


To run this program,open the `Text File Spell Checker` folder, run `server.py`, and open `index.html` in a browser. When you run the program, the screen as above appears. If you upload a text file in the 'Upload Text File' section, the original text content is displayed in the 'Input Text' section. The 'Recommendation' section provides the original word and the word to be corrected if there is a possibility of typo in the original text. In the 'Modify Text' section, an input box is provided so that the user can directly modify the content of the original text.

The figure above is an example of using this program. Words with a possibility of typo are highlighted, and recommended words of those words are provided. The user can modify his or her text based on this.

The 'missp.txt' file from https://www.dcs.bbk.ac.uk/~ROGER/missp.dat was used to implement the text file spell checker. When you see the 'missp.txt' file, the word after '$' is an error-free word, and the words below it are the words that has a possibility of may have been written the error-free word incorrectly.

For example, 'long', 'ro', 'rog', 'rong', 'ronge', 'worng', 'wrang', and 'wronge' are the words that may have been miswritten 'wrong'. In the case of 'ro', 'rog', 'rong', 'ronge', 'worng', 'wrang', and 'wronge', we are sure that the spelling is wrong, but in the case of 'long', it may have been used because the sentence



Text File Spell Checker

Upload Text File

Choose File test.txt

Input Text

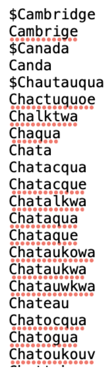
It wass a brit cold day in April, and hte clocks waer striking thirteen.

Recommendation

wass - was
a - I,above,all,an,and,answer,any,as,at,have,our,the,us
brit - bright
cold - colt
day - days
in - enforced,even,eye,for,impractical,into,is,on,with
and - I,a,accident,an,end,great,he,on,one,our,that
hte - the
waer - were

Modify Text

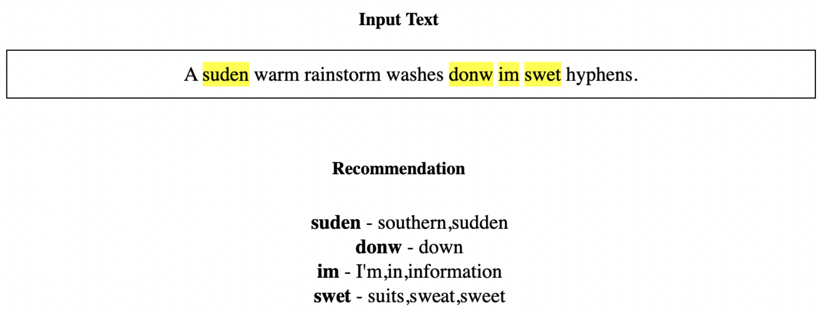
It wass a brit cold day in April, and hte clocks waer striking thirteen.



```
$Cambridge
Cambrige
$Canada
Canda
$Chautauqua
Chactuguoe
Chalktwa
Chaqua
Chata
Chatacqua
Chatacque
Chatalkwa
Chataqua
Chataque
Chataukowa
Chataukwa
Chatauwkwa
Chateau
Chatocqua
Chatogua
Chatoukouv
```

really needs ‘long’. So this spell checker is not perfect.

The following are the cases in which the program worked well and in which it did not work well.



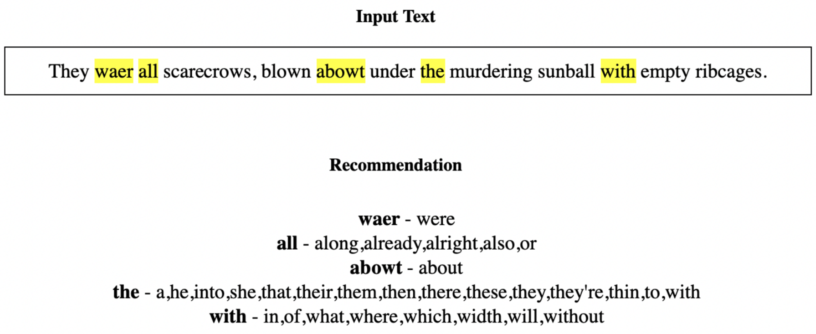
Input Text

A suden warm rainstorm washes donw im swet hyphens.

Recommendation

suden - southern,sudden
donw - down
im - I'm,in,information
swet - suits,sweat,sweet

This indicates when we type in a hurry. ‘sudden’ is incorrectly written as ‘suden’, ‘down’ as ‘donw’, ‘in’ as ‘im’, and ‘sweet’ as ‘swet’. If we look at the ‘Recommendation’ section, it provides the words that can be changed into. Therefore, by using this, we can change ‘A suden warm rainstorm washes donw im swet hyphens’ to ‘A sudden warm rainstorm washes down in sweet hyphens.’.



```
Input Text

They waer all scarecrows, blown abowt under the murdering sunball with empty ribcages.

Recommendation

waer - were
all - along,already,alright,also,or
abowt - about
the - a,he,into,she,that,their,them,then,there,these,they,they're,thin,to,with
with - in,of,what,where,which,width,will,without
```

‘waer’ and ‘abowt’ are the words that written ‘were’ and ‘about’ incorrectly. In the ‘Recommendation’ section, it recommends the word well. However, about ‘all’, ‘the’, and ‘with’, words are not written incorrectly. However, the program recommends other words. This has a problem. User might correct the text thinking that he or she wrote it incorrectly even though he or she wrote it correctly.

2 Real-time spell checker and word prediction

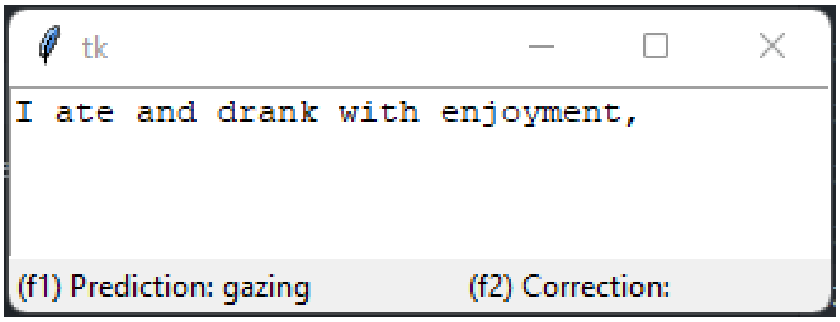
To run this program, run `live.py` using Python. This program was developed and tested using Python 3.10.4, and require the Tkinter library, which usually comes preinstalled with Python. When you run the program, a small window should appear containing a text input box and two labels along the bottom, the first for a suggestion for the next word, and the second for a correction for the current word. Press F1 and F2 respectively to append the suggested word or replace the current word with a correction.

The real time prediction works by using this corpus https://norvig.com/big.txt. The program generates a list of all pairs and triples of words in the corpus and how often they occur. The program also generates a list of each word in the corpus and the percentage of the total text that word occupies, which we called the word probability. The program first tries to generate a suggestion for the next word based on the previous two words. This is done by checking the list of word triples to see if the two last words match the first two words in any triples. We then generate a list of candidate words from the last entry in those triples. A score is generated for each candidate by multiplying the number of occurrences of the triple the word came from by the probability for that candidate word.

The candidate with the highest probability is finally suggested to the user. If no candidate words are produced by this process, the program tries again but only with last word entered. If that process doesn't produce a suggestion, the program defaults to suggesting the empty string. Also, the system will avoid recommending a word that is in the previous five words.

The real time correction works similarly to the text file correction, with an added step. We check a word against same the misspelling corpus to generate a list of possible corrections. We then use the list of common bigrams found here http://norvig.com/ngrams/count_2l.txt. The correction candidates are sorted by the commonality of the bigrams they contain, and the candidate that contains the most common bigram is the output.

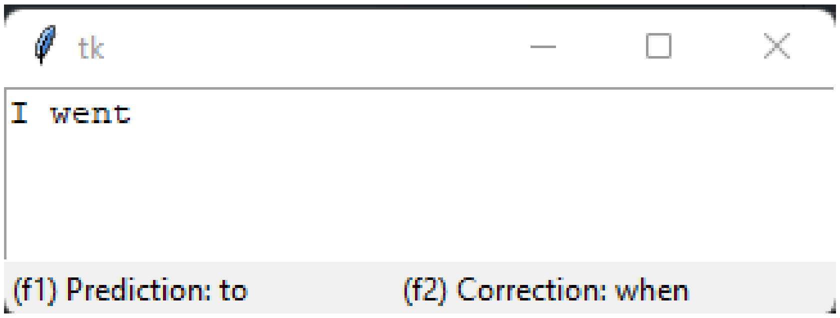
This approach to real time suggestion does not work well. The words it recommends often do not make sense. The biggest lesson we learned is that a text recommendation system should take into account more than just the last two words. It sometimes works, as seen below. That text was generated only from "I ate".



I ate and drank with enjoyment,

(f1) Prediction: gazing (f2) Correction:

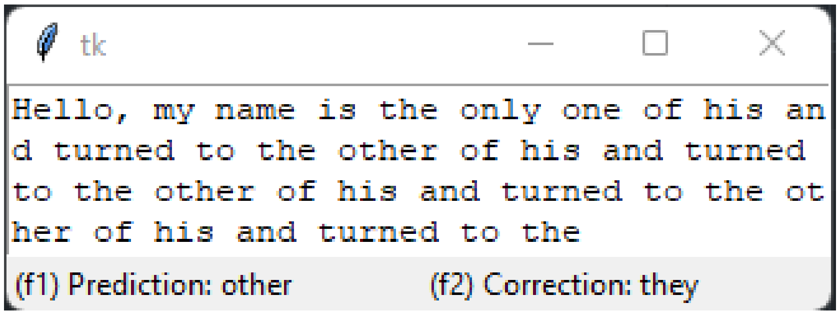
Another example of the system working well is with the input "I went". Both the prediction "to" and correction "when" are appropriate.



I went

(f1) Prediction: to (f2) Correction: when

However, the prediction system will often suggest inappropriate words or get itself stuck in loops. In the example below, the input was "Hello, my name is" and the first suggested word was "the", which doesn't make sense. Also, the system gets caught in a nonsensical suggestion loop, which often happens if you have it suggest more than one or two words at a time.



```
Hello, my name is the only one of his an
d turned to the other of his and turned
to the other of his and turned to the ot
her of his and turned to the
```

(f1) Prediction: other (f2) Correction: they