Homework 3

Yuxuan Zhang

For this model, we have the following assumptions:

- Every word in the 'count_1w.txt' file is correct.
- Only one edit is needed in each incorrect word not including transposition.
- Weighted cost for edits are derived from the emprical data (additions.csv, deletions.csv, substitutions.csv).
- Words that do not appear in the dictionary ('count_1w.txt') are considered incorrect.

Here are some cases where the correction works:

In [ ]:
```
from HW3 import correct
# Deletion
print(correct('buesiness'))
print(correct('intellligence'))
print(correct('bpble'))
print(correct('homwork'))
print(correct('basket ball'))
# Substitution
print(correct('superheio'))
print(correct('correct'))
print(correct('beausiful'))
# Insertion
print(correct('condesending'))
print(correct('populatio'))
print(correct('amazn'))
```

```
business
intelligence
bible
homework
basketball
superhero
correct
beautiful
condescending
population
amazon
```

Here are some cases where the correction does not work

Words with multiple valid edits

In [ ]:
```
print(correct('speeling'))
print("I meant to spell speeding.")
```

```
spelling
I meant to spell speeding.
```

Words with error that needs more than two edits

In [ ]:
```
print(correct('beyasian'))
print(correct('condscanding'))
```

```
beyasian
condscanding
```

Words that are not included in the dictionary.

In [ ]:
```
print(correct('taikont'))
print('This should be taikonaut\n')

print(correct('pneumonoultramicroscopicsilcovolcanoconiosis'))
print('This should be pneumonoultramicroscopicsilicovolcanoconiosis, a term in b
```

```
trikont
This should be taikonaut

pneumonoultramicroscopicsilcovolcanoconiosis
This should be pneumonoultramicroscopicsilicovolcanoconiosis, a term in biology
```

The actually incorrect word is included in the dictionary

In [ ]:
```
freq_dict = {}
with open('count_1w.txt', mode='r') as f:
    for line in f:
            parts = line.split()
            freq_dict[parts[0]] = int(parts[1])
print('inteligence' in freq_dict)
print('heirarchy' in freq_dict)

print(correct('inteligence'))
print(correct('heirarchy'))
```

```
True
True
inteligence
heirarchy
```

Explaining the poor judgements and future improvements of this model:

- For words with multiple valid edits, the poor judgements is because our model does not take the surrounding text into consideration. We will need a more complicated model that can take the surrounding text as extra information and use this as another layer of prior to calculate the probabilities of each candidates.
- Words that does not have the correct form in the dictionary. To improve this, we will have to use a larger dictionary with a comprehensive set of words.
- For edits with more than one edits, it is not supported by this model. To implement a model with multiple edits, we will need to rewrite the generate candidate part, where candidates with multiple edits will be added.

- For incorrect words in the dictionary, we will need to make sure the source of the dictionary and each word inside is correct