Lesson Plan: Finding Patterns in Spelling Errors and History

Standards

- **RL.8.5.** Compare and contrast the structure of two or more texts and analyze how the differing structure of each text contributes to its meaning and style.
- **L.9-10.4.** Determine or clarify the meaning of unknown and multiple-meaning words and phrases based on *grades 9–10 reading and content*, choosing flexibly from a range of strategies.
- **RH.6-8.4.** Determine the meaning of words and phrases as they are used in a text, including vocabulary specific to domains related to history/social studies.

Overview

Have you ever wondered how your computer or phone is able to correct your spelling and grammar? It would not be possible without recognizing patterns in correct and incorrect spellings as well as understanding language structure and context. The same algorithms that can predict if "surprise" is correctly spelled, can also shed light on patterns and themes in our culture. In this lesson, students will analyze spelling errors and large sets of data to find patterns, develop abstractions, and discover how large amounts of data can tell us much about our society.

Prerequisites

- None

Materials

- Install Python 2.7
- Google Spreadsheet with a form - optional, but saves a lot of time collecting data

Suggested Time Frame

- Looking for Patterns in Spelling Errors
  - Collecting Spelling Data - Grades 6-12 (20 min)
  - Finding Patterns and Creating Abstractions from the Data - Grades 6-12 (30 - 45 min)
- Using n-grams to Explore Patterns in Culture and History - Grades 6-12 (30 - 45 min)

Terminology

- **n-gram:** Text or items with a length of n. The term can be applied to genetic strands, molecule chains, etc. (Wikipedia). For example: 'Hello how are you today?' can be split by letters, words, or sentences to create n-grams such as:

  'how' - 3-gram (letters)
  'Hello how are you' - 4 gram (words)
  'Hello how are you today' - 1-gram (sentence)
- **corpus:** A body of information. For example, in the Google Ngram viewer, the English corpus consists of all the books in English that Google has scanned. Another examples are the Shakespearean corpus or Project Gutenberg, the free e-book repository.

Activity: Looking for Patterns in Spelling Errors

1. When we look at a word, we can often tell whether or not it is spelled correctly even if we don't know the correct spelling. Humans sense of sight has evolved to discern if something is out of place. This is why reading is one of the best ways to become better at spelling, it trains your visual senses to know what patterns to expect.

Student Question:
This paragraph has circulated the Internet many times:

It deosn't mttaer in waht oredr the ltteers in a wrod are, olny taht the frist and lsat ltteres are at the rghit pcleas. The rset can be a toatl mses and you can sitll raed it wouthit a porbelm. Tihs is bcuseae we do not raed ervey lteter by ilstef, but the wrod as a wlohe.

Why might a person from a non-English speaking country have difficulty reading this?
A: If you do not speak English, then you have not seen these words often enough to recognize the patterns and figure out the words from the context. This would likewise apply to English speakers if the language was foreign to them. However it is possible to discover patterns as seen below.

2. To a computer, these words are raw data (whether considered correct or misspelled).
3. There are so many ways in which each word can be spelled incorrectly that to tell the computer what to do in each instance would be very difficult. Instead, computers have dictionaries or lists of correctly spelled words and apply heuristics and rules to suggest corrections for misspelled words.

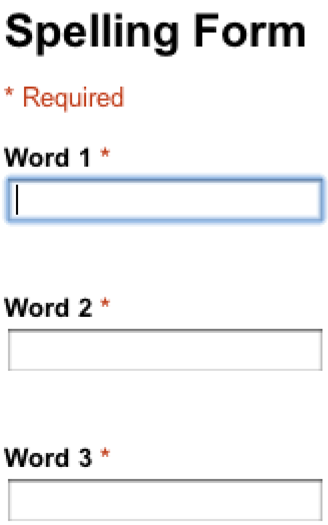
Subactivity: Collecting Spelling Data

Before class:

1. Create a form to collect the students' data.
2. Pick five to ten words for students to spell. These words could be selected from a book, recent exam or Internet search for [`Commonly Misspelled Words`].

During class:

3. Have the students open the form.
4. Say the words to students one at a time and have them write down how they believe it is spelled.
   a. An easy way to do this is using a Google Form so the data can be collected quickly, but the data can be collected via paper.



Spelling Form

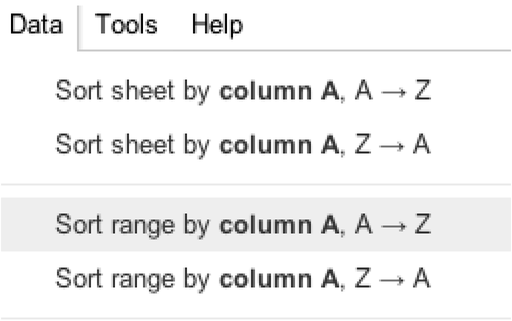
* Required

Word 1 *

Word 2 *

Word 3 *

5. After all students have submitted their forms, share the results with the students and have them sort each column alphabetically.


Data
Tools
Help
Sort sheet by **column A**, A → Z
Sort sheet by **column A**, Z → A
Sort range by **column A**, A → Z
Sort range by **column A**, Z → A

Paperless versions:

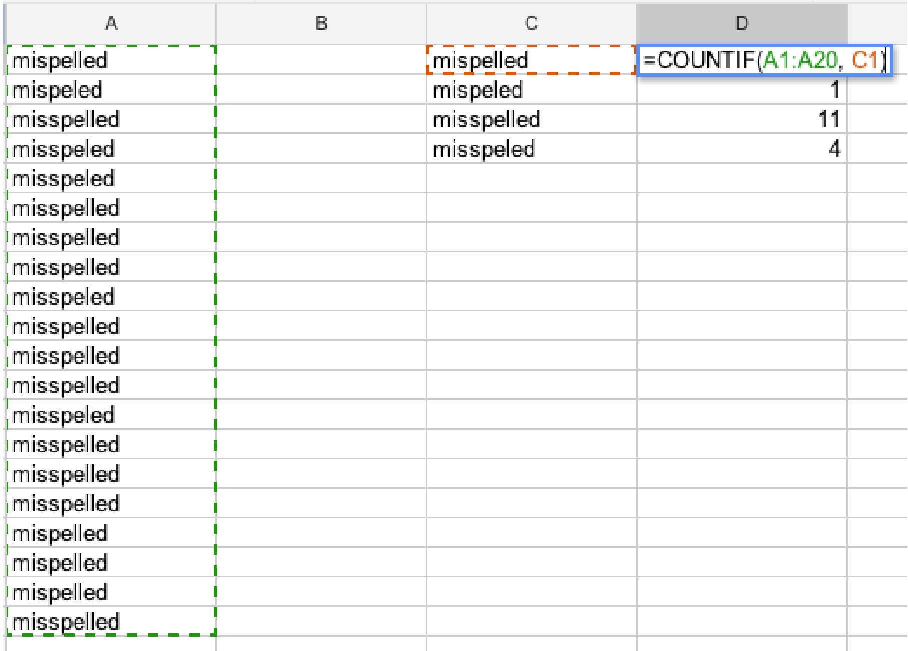
- Have students pass up their papers or call out their results and place check marks next to the duplicate results on a board.
- Create a table of data on the board from an already existing set of data.
- The advantage of the paperless Google Form is it preserves the anonymity of the students and the data collection is faster.

Subactivity: Finding Patterns and Creating Abstractions from the Data

1. Have students find the misspelled words and mark them by coloring the cells.
    a. Initially they can be word specific (e.g. misspelled vs mispelled, because of the omission of a letter and likely because it was spelled phonetically).
2. Identify the most common errors in each column.
3. See if the students can abstract the patterns they find to be applicable to more than just one word.
    b. Some common ways spelling errors are made:
        i. omitting one or more letters - hello vs helo
        ii. using the wrong letter - readable vs readible
        iii. transposing/swapping letters - believe vs beleive

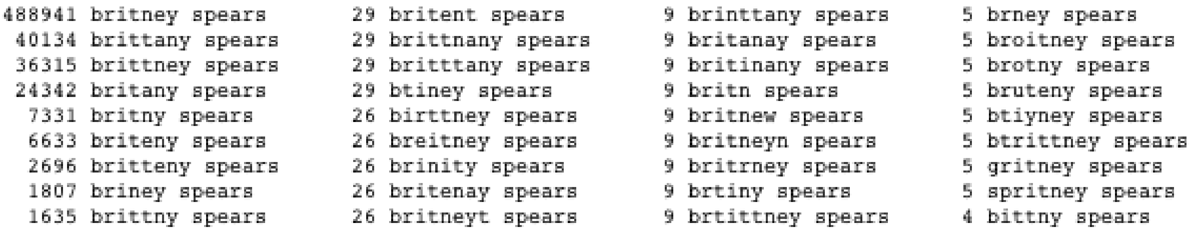
4. Students can determine which spelling errors occur more often than others. Here is an example spreadsheet and a screenshot can be found on the next page.
    a. Choose a blank column in the data collection spreadsheet used earlier, and into one of the blank cells type =UNIQUE(X:Y), where X and Y is the data collected earlier. (e.g. A1:A20).
        i. The UNIQUE function fills the blank column with one of each type of response.
    b. In the next column, in each cell next to enter =COUNTIF(X:Y, Z), where Z references the cell with the pattern to count.
    c. Even though this is a very small text sample, it does exemplify several types of spelling errors and how common they are.

Teacher Note:
In practice, this type of spreadsheet could be used to identify patterns in spelling tests. For students, it can help them identify the types of errors they tend to make. For teachers, it might suggest focus areas for instruction or practice.



A	B	C	D
mispelled		mispelled	=COUNTIF(A1:A20, C1)
mispeled		mispeled	1
misspelled		misspelled	11
misspeled		misspeled	4
misspeled			
misspelled			
misspelled			
misspelled			
misspeled			
misspelled			
misspelled			
misspelled			
misspeled			
misspelled			
misspelled			
misspelled			
mispelled			
mispelled			
mispelled			
misspelled			

5. Google receives billions of queries a day. This type of data helps improve the suggestions for spellings and results. Without misspellings, it would be much more difficult to find patterns and develop algorithms to guess at what people might mean with a query.
6. Below is a sample of queries from people searching on Google for [`Britney Spears`]. This shows how often people used each type of spelling.



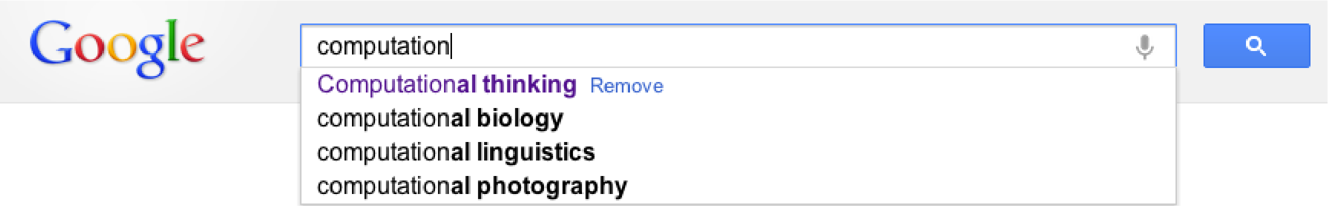
488941	britney spears	29	britent spears	9	brinttany spears	5	brney spears
40134	brittany spears	29	brittnany spears	9	britanay spears	5	broitney spears
36315	brittney spears	29	britttany spears	9	britinany spears	5	brotny spears
24342	britany spears	29	btiney spears	9	britn spears	5	bruteny spears
7331	britny spears	26	birttney spears	9	britnew spears	5	btiyney spears
6633	briteny spears	26	breitney spears	9	britneyn spears	5	btrittney spears
2696	britteny spears	26	brinity spears	9	britrney spears	5	gritney spears
1807	briney spears	26	britenay spears	9	brtiny spears	5	spritney spears
1635	brittny spears	26	britneyt spears	9	brtittney spears	4	bittny spears

7. From the data, it is possible to determine what the most likely correct spelling of Britney Spears might be. Google can return the expected results even if her name is misspelled. Note that the algorithms doing the suggestions, do not actually have to know who Britney Spears is.

8. On the other hand, it is entirely possible that the most frequent query uses incorrect spelling. Therefore, algorithms are refined based on more incoming data but the process is automated.

Activity: Using n-grams to Explore Patterns in Culture and History

1. How does Google Instant work? As you type letters or words into the search box, Google uses probability to calculate the most likely next word or letter and suggests it to you. Google uses frequency tables (as we did above), n-grams, and other methods.



computation

Computational **thinking** Remove

computational **biology**

computational **linguistics**

computational **photography**

2. An n-gram is text or items with a length of n. The sentence, 'Hello how are you?’ can be broken down in multiple ways, such as:
    a. 1-gram by letter: ‘H’, ‘e’, ‘l’, ‘l’, ‘o’...
    b. 1-gram by word: ‘Hello’, ‘how’, ‘are’...
    c. 1-gram by sentence 'Hello, how are you?’
    d. 2-gram by letter: ‘He’, ‘ll’, ‘o_‘...
    e. 2-gram by word: ‘Hello how’, ‘are you’
3. N-grams are only useful for autocompletion if you have a large corpus of words and sentences to compare against. If you had a large database of words and sentences, you could predict how likely it is that they will type, ‘how’ if they previously typed ‘Hello’.
4. The code below separates a text file (sample taken from the Gutenberg library) into n-grams by word, and determines the frequency of how often it occurs.

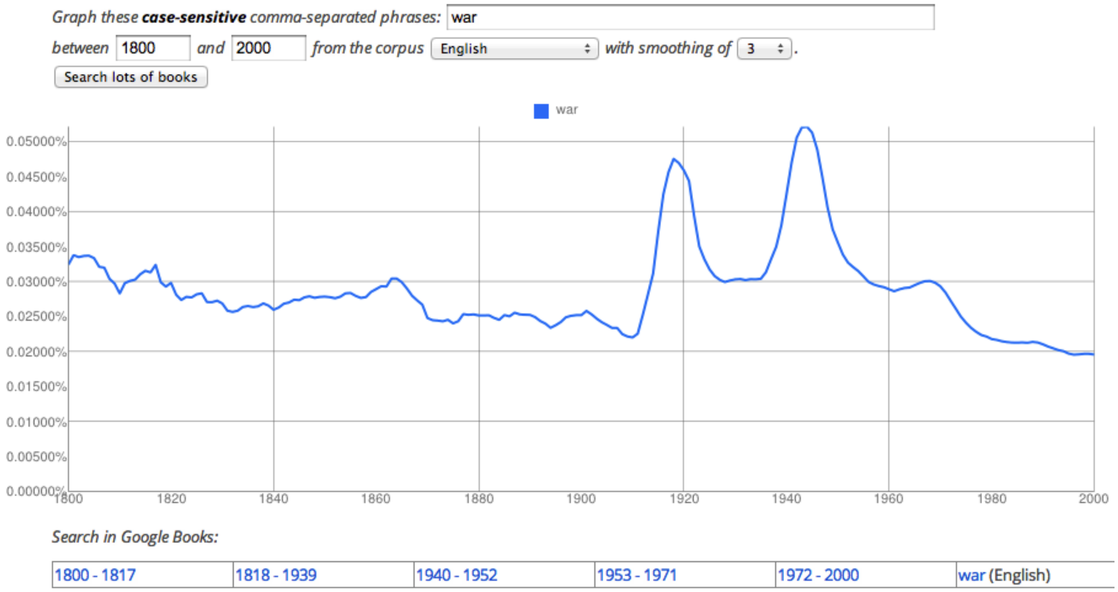
```
import urllib2, re
from collections import Counter

n = 1 #size of n-gram

#You can substitute the link below with any web page with plain text
url = 'http://www.gutenberg.org/cache/epub/76/pg76.txt'

page = urllib2.urlopen(url)

contents = page.read()
#contents = contents[:len(contents)/500] #smaller data/less memory option

words = re.findall(r'\w+', contents) #Regular expression to find all words

ngrams = [] #A list to store the ngrams
i = 0

while i < len(words):
    ngrams.append(tuple(words[i:i+n])) #Splits "words" into n sized tuples
    i += n

print Counter(ngrams) #Calculates the frequency of each n-gram
```

Sample Output:

```
Counter({('and',): 6195, ('the',): 4673, ('I',): 3666, ('a',): 3155, ('to',):
3005, ('it',): 2310, ('t',): 2083, ('was',): 2037, ('of',): 1739, ('he',):
1560, ('in',): 1461, ('you',): 1377, ('that',): 1146, ('s',): 1097, ('on',):
885, ('up',): 860,...
```

Student Question:
Why might it be useful to perform an n-gram analysis of a single book?
A: Answers may vary, but it provides insight into an author or the theme of the book. You may find that the author uses highly descriptive imagery or focuses more on dialog. One could also infer the type of book by the n-grams (e.g. technical, fiction, biography).

5. This is useful information for one book, but what if you wanted to know the probability of an n-gram coming up during a historical event or today? Google has made large (approximately 900 MB each) sets of n-grams available for free that you can use parse.
6. It might be easier to use the Ngram viewer, a tool that incorporates the data from the Google Books scanning project that contains ngram data from approximately 4% of the world's books. One example comes from searching for the term [`war`]:



*Graph these **case-sensitive** comma-separated phrases:* war

between 1800 *and* 2000 *from the corpus* English *with smoothing of* 3.

Search in Google Books:

1800 - 1817 | 1818 - 1939 | 1940 - 1952 | 1953 - 1971 | 1972 - 2000 | war (English)

7. As might be expected, the graph has two large spikes around the times of World War I and II.
8. So many interesting permutations are possible that they are impossible to list, but here are a few more examples to get you thinking:
    a. [`car, horse`]
    b. [`capitalism, communism`]
    c. [`planet, sun`]
    d. [`plato`]
    e. [`Shakespeare`]

    f. [`television, newspaper`]

9. Click on the date ranges below the graph to see the excerpts from where the data came from.

Teacher Note:
These examples are all from the English corpus and therefore are culturally biased. Other corpa can be searched to see their perspective on these and other n-grams.

Extension

- Change the n-gram code above to break a text down by letters or entire sentences. Note: you may want to look at the regular expression documentation here, here, and examples here.
- Peter Norvig provides code for a Spelling Checker in Python that can be adapted or added to - http://norvig.com/spell-correct.html
- In order to use Peter Norvig's aforementioned Spelling Checker code, download this text and place it in the same directory to train the algorithms - http://norvig.com/big.txt
- Students can find and explore more deeply the results from various n-grams and correlate them with cultural or historical events (e.g. why does [`carrot`] become much more prominent in the 1940's? Hint: Search for [`carrots and radar`]).

More lessons and examples can be found at Google's Exploring Computational Thinking website.