Semantic Search

By Brayton Hall

Purpose

- To build a specialized *semantic search engine*, which can return the most similar paragraphs to an input string

- A 'search-between-lines' app, to search by connotations and misremembered quotes, rather than exact fragments

- e.g., an excerpt from a novel such as *War and Peace*, or an important statement in a legal documents

Why?

If you've ever used a Kindle or a website-specific search engine (like Reddit), they often require you to be **annoyingly exact** to find what you're looking for, especially if you can't quite remember the words.

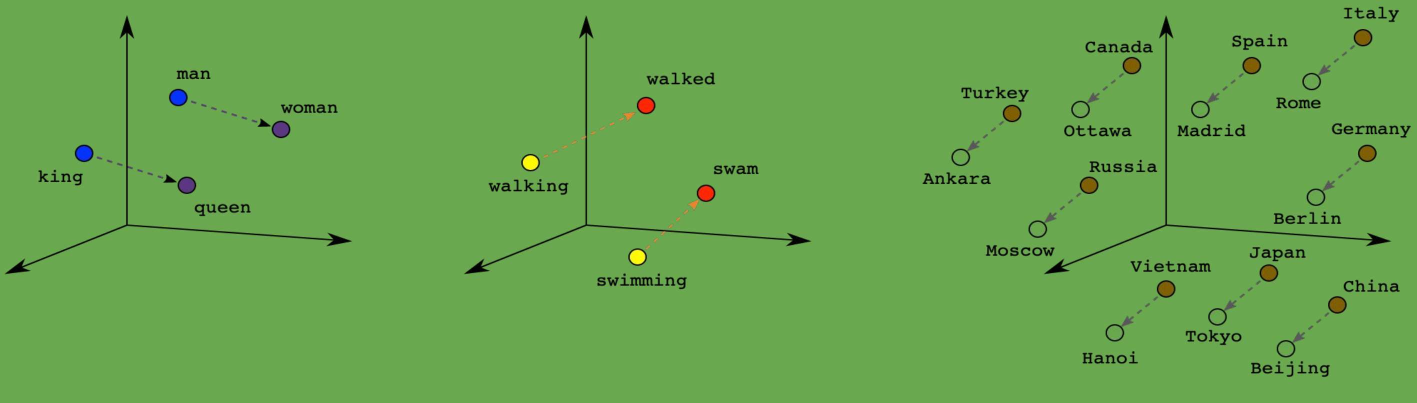
My idea for Semantic Search was inspired by that frustration, in order to create a search function by *related* words and connotations, somewhat like Google, but ***within the scope of desired documents***.

TRAINING DATA AND EDA

- 100 of the top 'free ebooks' from Project Gutenberg, scraped with BeautifulSoup
- approximately 12 million words, cleaned and tokenized
- approximately 57,000 paragraphs, each 12,000 *characters* long
- features such as 'lexicon count' and 'lexicon ratio', i.e. engineered representations of *unique words* and *unique word:total word* ratio, respectively (useful for identifying 'literary' or difficult prose)

Word2Vec

- Word2Vec is a two-layer neural net which creates unique representations of words using vectorization, i.e. it creates representations of words using matrices based on the entire set of training data (the 'corpus' in NLP)
- Those representations actually contain *meaningful semantic information*, such that, using Word2Vec, we can actually perform the following classic operation: [king + woman - man] = [queen]



Male-Female

Verb Tense

Country-Capital

MODEL PREPARATION

- Just like Word2Vec, Doc2Vec creates vector representations of words

- Doc2Vec also creates vector representations of *documents*, which are paragraphs here (I arbitrarily decided 1200 characters == 1 paragraph)

- In order for Doc2Vec to vectorize tokenized docs, each doc (paragraph) must be 'Tagged'. For example, the first paragraph of Anna Karenina:

```
TaggedDocument(words=['happy', 'families', 'are', 'all', 'alike', 'every', 'unhappy', 'family', 'is', 'unhappy',
'in', 'its', 'own', 'way'...'after', 'the', 'quarrel', 'prince'], tags=['47831'])
```

This is paragraph 47,831 in my entire corpus of approx. 57, 000 paragraphs.

MODEL CREATION

Doc2Vec parameters:

- Vector size of 300 (each paragraph is represented as a vector with 300 dimension)
- Learning rate of .01
- 200 Epochs
- Window size of 20 (for tokens, since word order in paragraphs matters)

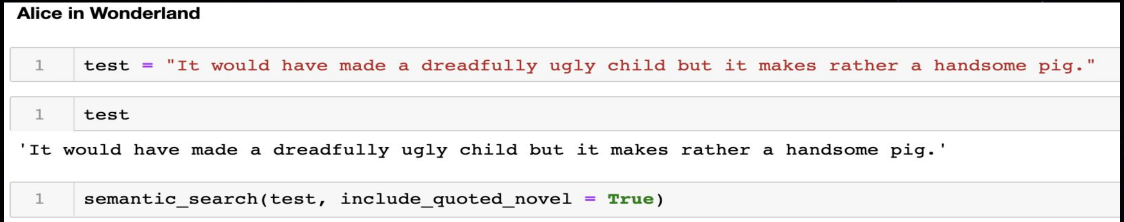
MODEL APPLICATION

Storing and application:

- I pickled the model, so it can be quickly loaded (150 mb size, 20 min to build otherwise) and stored it on an AWS bucket on S3
- Used in the 'Semantic Search Function', shown on next slide ---->

EXAMPLE with Alice in Wonderland

QUERY:



Alice in Wonderland

```
test = "It would have made a dreadfully ugly child but it makes rather a handsome pig."
```

```
test
```

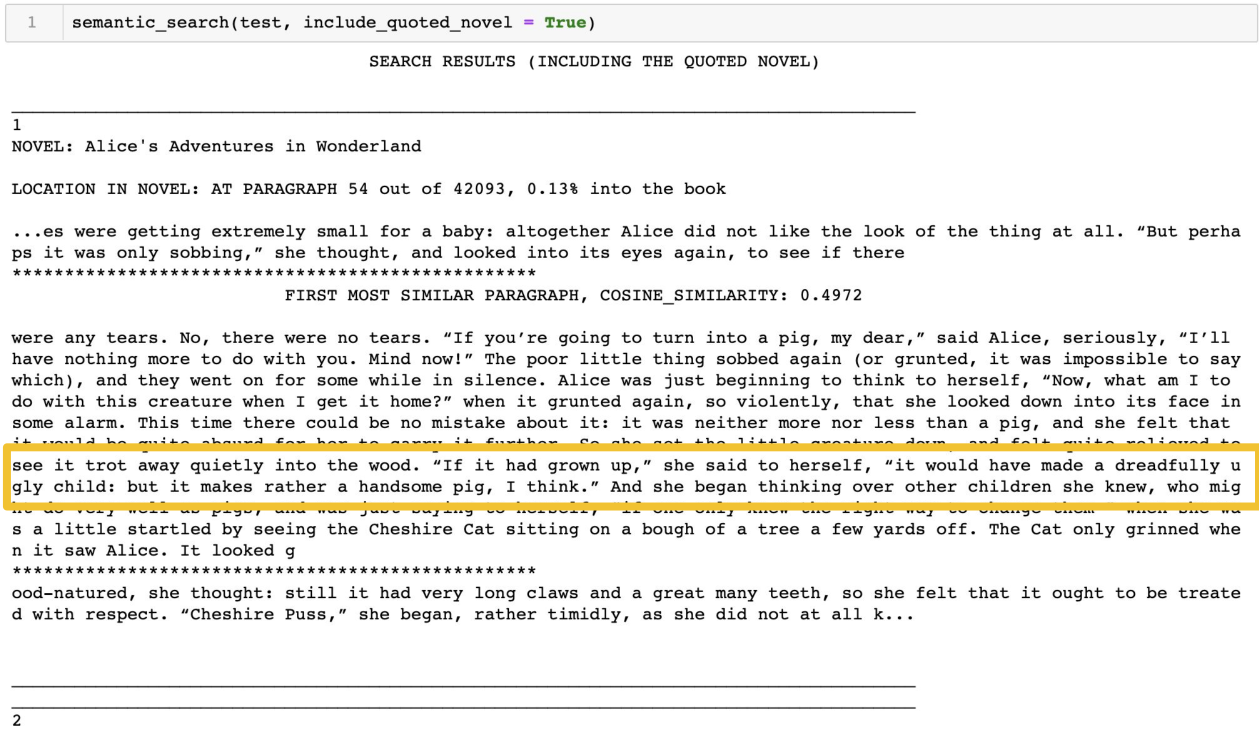
```
'It would have made a dreadfully ugly child but it makes rather a handsome pig.'
```

```
semantic_search(test, include_quoted_novel = True)
```

Imperfect match is still found in the search results below!

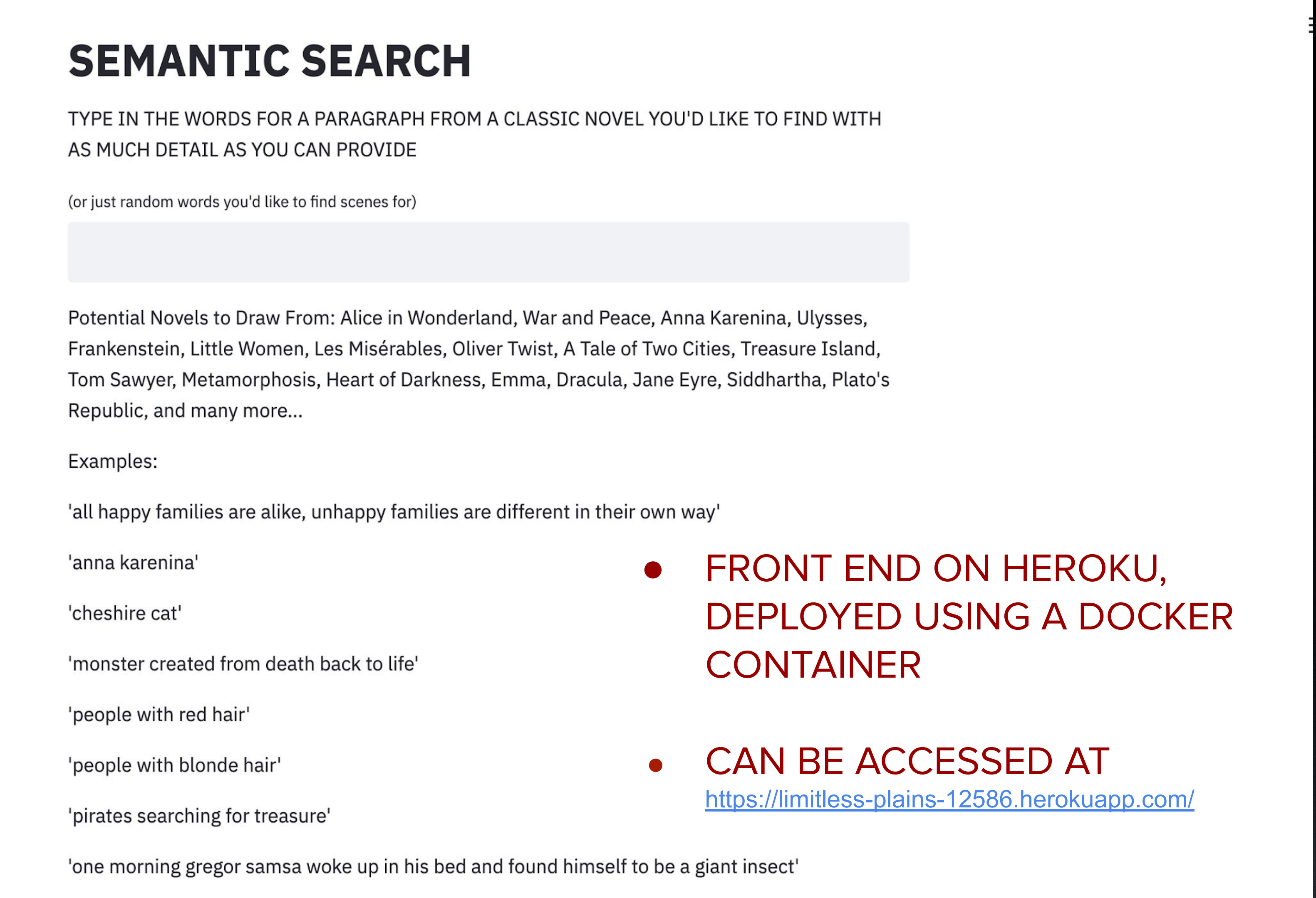
RESULTS:

(with location and cosine similarity of the closest five paragraphs)



```
semantic_search(test, include_quoted_novel = True)
```

```
                              SEARCH RESULTS (INCLUDING THE QUOTED NOVEL)

____________________________________________________________________________________________________
1
NOVEL: Alice's Adventures in Wonderland

LOCATION IN NOVEL: AT PARAGRAPH 54 out of 42093, 0.13% into the book

...es were getting extremely small for a baby: altogether Alice did not like the look of the thing at all. “But perha
ps it was only sobbing,” she thought, and looked into its eyes again, to see if there
*****************************************************
                         FIRST MOST SIMILAR PARAGRAPH, COSINE_SIMILARITY: 0.4972

were any tears. No, there were no tears. “If you’re going to turn into a pig, my dear,” said Alice, seriously, “I’ll
have nothing more to do with you. Mind now!” The poor little thing sobbed again (or grunted, it was impossible to say
which), and they went on for some while in silence. Alice was just beginning to think to herself, “Now, what am I to
do with this creature when I get it home?” when it grunted again, so violently, that she looked down into its face in
some alarm. This time there could be no mistake about it: it was neither more nor less than a pig, and she felt that
it would be quite absurd for her to carry it further. So she set the little creature down, and felt quite relieved to
see it trot away quietly into the wood. “If it had grown up,” she said to herself, “it would have made a dreadfully u
gly child: but it makes rather a handsome pig, I think.” And she began thinking over other children she knew, who mig
ht do very well as pigs, and was just saying to herself, “if one only knew the right way to change them—” when she wa
s a little startled by seeing the Cheshire Cat sitting on a bough of a tree a few yards off. The Cat only grinned whe
n it saw Alice. It looked g
*****************************************************
ood-natured, she thought: still it had very long claws and a great many teeth, so she felt that it ought to be treate
d with respect. “Cheshire Puss,” she began, rather timidly, as she did not at all k...


____________________________________________________________________________________________________
____________________________________________________________________________________________________
2
```



SEMANTIC SEARCH

TYPE IN THE WORDS FOR A PARAGRAPH FROM A CLASSIC NOVEL YOU'D LIKE TO FIND WITH AS MUCH DETAIL AS YOU CAN PROVIDE

(or just random words you'd like to find scenes for)

Potential Novels to Draw From: Alice in Wonderland, War and Peace, Anna Karenina, Ulysses, Frankenstein, Little Women, Les Misérables, Oliver Twist, A Tale of Two Cities, Treasure Island, Tom Sawyer, Metamorphosis, Heart of Darkness, Emma, Dracula, Jane Eyre, Siddhartha, Plato's Republic, and many more...

Examples:

'all happy families are alike, unhappy families are different in their own way'

'anna karenina'

'cheshire cat'

'monster created from death back to life'

'people with red hair'

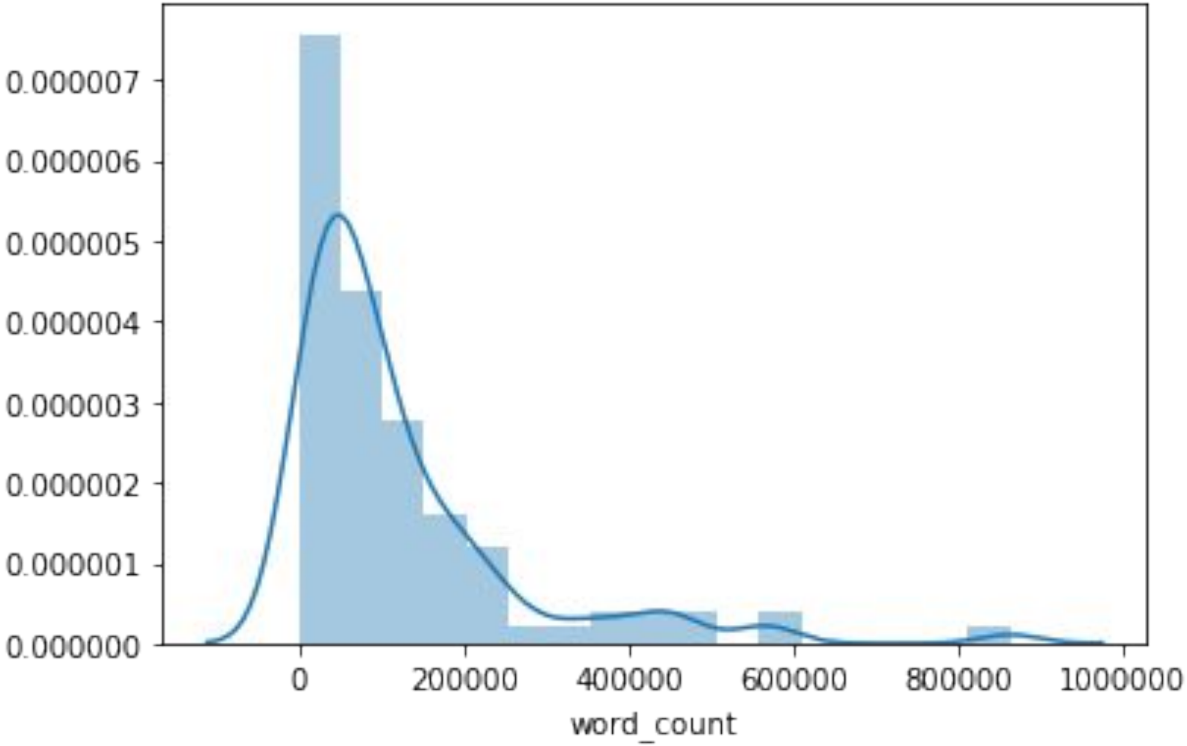
'people with blonde hair'

'pirates searching for treasure'

'one morning gregor samsa woke up in his bed and found himself to be a giant insect'

- FRONT END ON HEROKU, DEPLOYED USING A DOCKER CONTAINER
- CAN BE ACCESSED AT https://limitless-plains-12586.herokuapp.com/

WORDS BY NOVEL



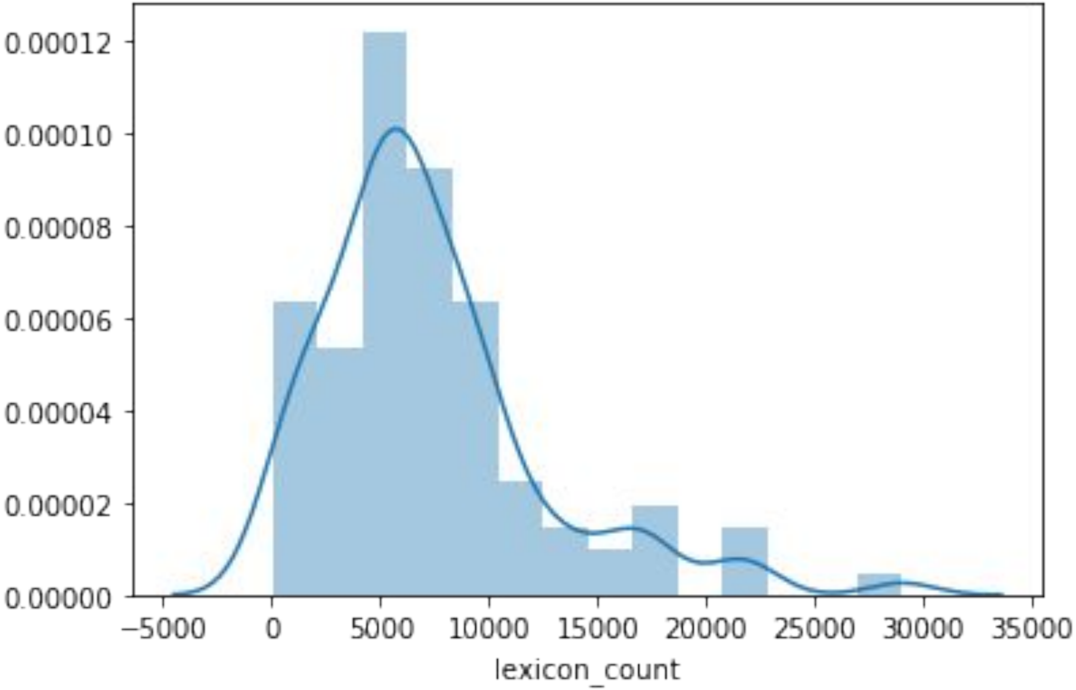
DATA
100 NOVELS

AVERAGE WORD COUNT
121,642 WORDS

MEDIAN WORD COUNT
70,054 WORDS

LONGEST NOVEL
862,471 WORDS

WORDS BY PARAGRAPH

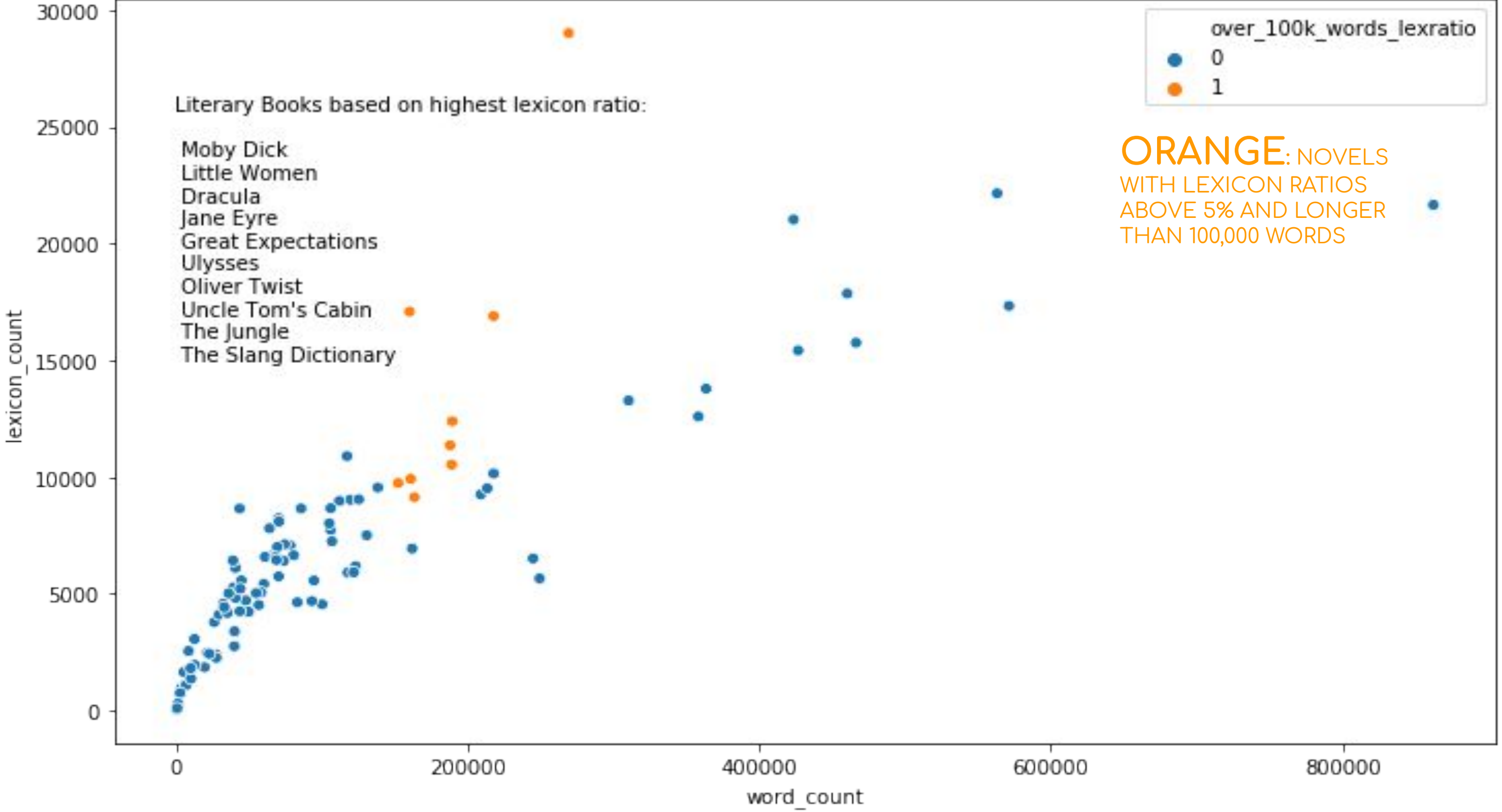


DATA
99 NOVELS

AVERAGE LEXICON
7267 UNIQUE WORDS

MEDIAN LEXICON
6,412 UNIQUE WORDS

LONGEST LEXICON
29,012 UNIQUE WORDS

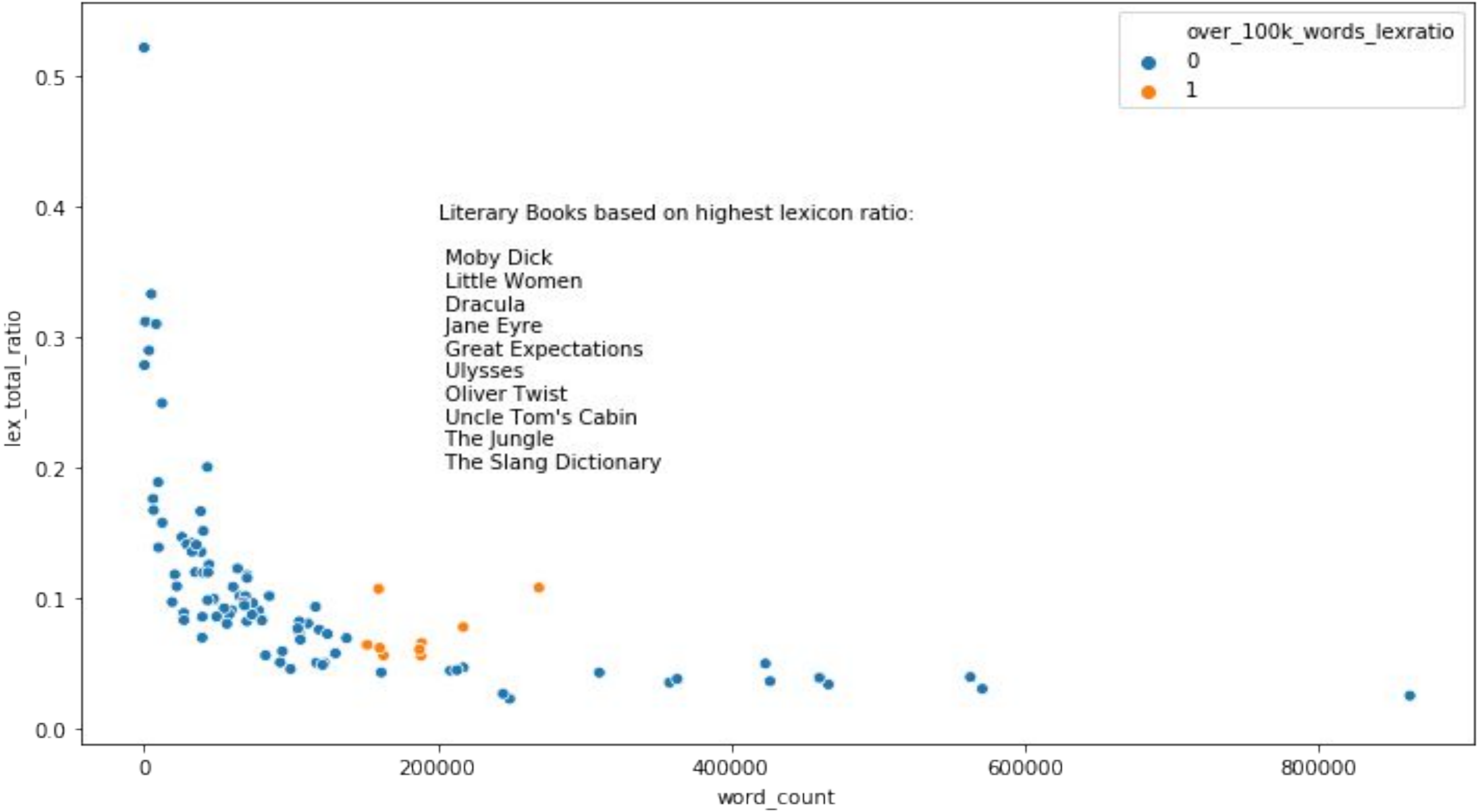


Literary Books based on highest lexicon ratio:

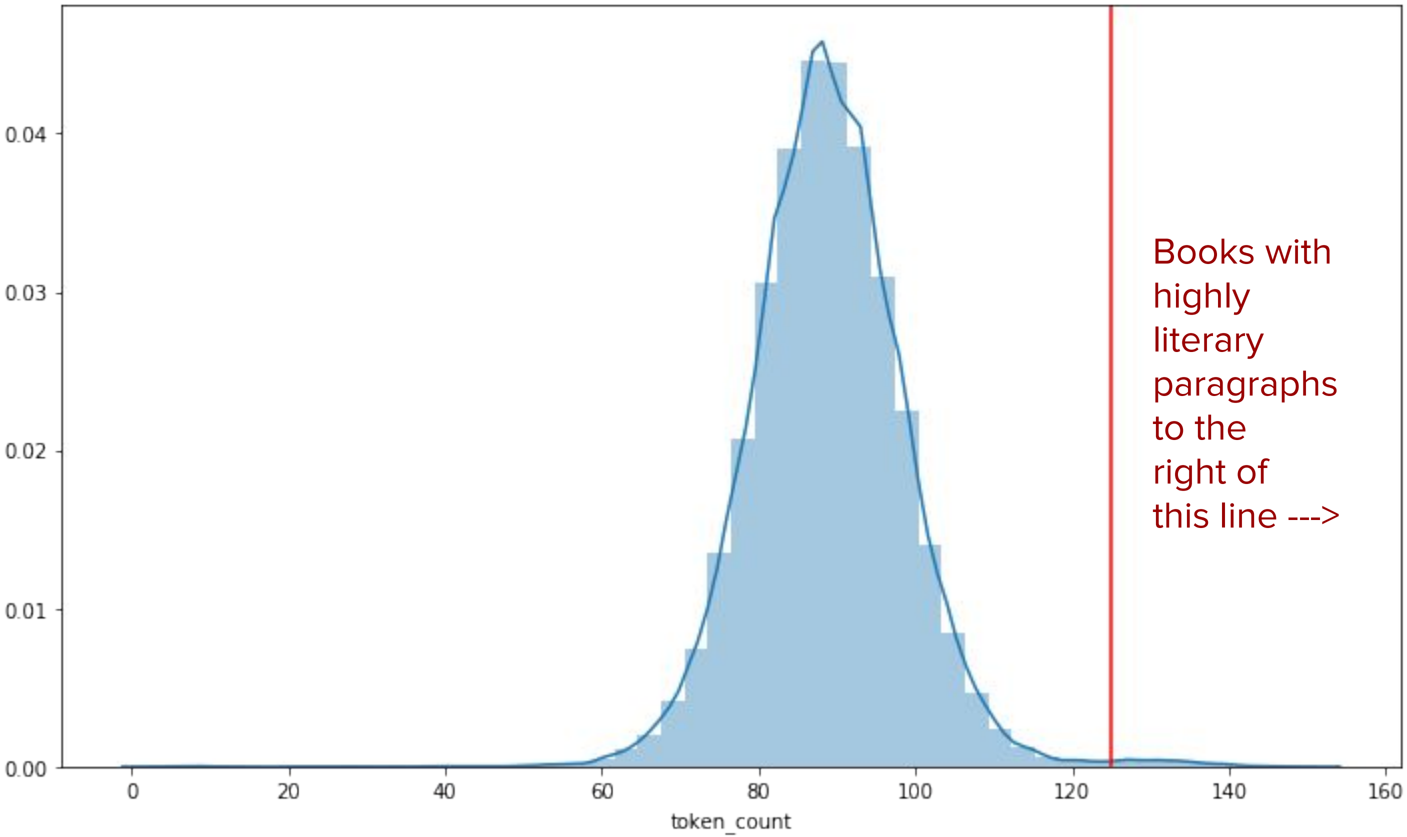
Moby Dick
Little Women
Dracula
Jane Eyre
Great Expectations
Ulysses
Oliver Twist
Uncle Tom's Cabin
The Jungle
The Slang Dictionary

ORANGE: NOVELS WITH LEXICON RATIOS ABOVE 5% AND LONGER THAN 100,000 WORDS

- 'unique words' goes up with the length of a book, naturally, but the *ratio* of lexicon length to total words tends to go down (presumably because longer books repeat the same words over and over).
- But the books with the *highest* lexicon length AND words above a certain threshold include highly representative 'difficult' or wordy classics, Ulysses being one of the prototypical difficult English novels.



- Same findings and books, but with a different y-axis.
- We can see that longer books tend to have a lower lexical uniqueness, but if we look at books above a particular threshold of uniqueness (5%), once again they represent our idea of difficult or wordy books (or a dictionary).



Books with highly literary paragraphs to the right of this line --->

'Wuthering Heights'

'The Works of Edgar Allan Poe'

'**The Slang Dictionary**'

'The Iliad'

'**Ulysses**'

'**Little Women**'

'The Picture of Dorian Gray'

'Second Treatise of Government'

- Distribution of the PARAGRAPHS (57,000) which make up the 100 books
- By looking at the paragraphs with the longest unique words, we once again find 'literary' or difficult prose in the associated books, some which **overlap with those** before (Little Women, Ulysses, Slang Dictionary).