Project report of computational sociology

TOPIC ANALYSIS BASED ON FIFTY VICTORIAN ERA NOVELISTS AUTHORSHIP ATTRIBUTION DATA SET

Rheina Trudy Williams
School of Computer Science, Northwestern Polytechnical University, Xi'an, China

Abstract Topic analysis is a technique in machine learning to organize and assign topics for large groups of text data. Latent Dirichlet allocation (LDA) is used to model the data observations in distinct categories. This project focuses to do a topic analysis using Latent Dirichlet allocation and create word clouds with the Jupyter Notebook platform. In this project, I have used the Fifty Victorian Era Novelists Authorship Attribution data set extracted from the University of California Irvine Machine Learning repository to determine the common topics and visualize them through word clouds. The entities are 19th century English language authors with at least five books published. I have found several common themes in the text and created some word clouds by importing the data, cleaning the data, visualizing the word cloud, and modeling the data with Latent Dirichlet allocation. The results of this project present the common topics of the Fifty Victorian Era Novelists Authorship Attribution data.

Keywords Jupyter Notebook, latent dirichlet allocation, topic analysis, word cloud

1 Background

Knowing the topics of certain texts helps greatly in research. It can be done manually by reading and analyzing, but what if the text that needs to be analyzed has an immense amount of information. Checking every text manually will take a lot of time and is not effective. This is where computation can be used to assist in analyzing topics.

Topic analysis, also known as topic detection, topic modeling, or topic extraction is a machine-learning technique that organizes and analyzes large groups of textual data, by giving tags based on common keywords of each topic. Topic analysis aims to discover the common themes and uncover unpopular structures that run through a corpus. It is useful for text clustering, information retrieval, and feature selection. General tasks of topic modeling are dimensionality reduction, unsupervised learning, and tagging.

A way of doing topic analysis is by using LDA (Latent Dirichlet Analysis), a generative probabilistic model that supposes each topic is a combination over an underlying group of words, and each document is a combination of over a group of topic probabilities. After modeling the topics with LDA, visualization can be done to have a better presentation. A great visual presentation for texts is word cloud since the frequency of each keyword can be detected from the sizes of each word.

An example of text data set is information collected from books. Books contain many text data with different themes. To analyze the topics, the topic analysis technique can be used.

2 Goal

This project does topic analysis on a text data set that is Fifty Victorian Era Novelists Authorship Attribution data set involving works of fifty well-known authors from the 19th century. The general objectives of this project are to obtain one text data set with more than 30,000 instances, find the topics of each instance using LDA, and visualize the common topics using word cloud.

The specific objectives of this project are to:

1) Find mainstream topics from the works of fifty

Victorian-era authors.

2) Create word clouds from the discovered common topics.

3 Solutions

3.1 Data Set Description

The data set used in this project is Fifty Victorian Era Novelists Authorship Attribution Data. It is used to decrease the bias and create a reliable authorship attribution data set, extracted from the works of fifty well-known authors.

The data set was collected from the GDELT (Global Database of Events, Language and Tone) database, an open platform for research and analysis of global society. It is filtered with three criteria: English language authors, authors that have at least five books published (at least 5), and 19th century authors. Thus fifty authors are found with all the criteria fulfilled. The instances are 1000-word sequences that are divided from the works of every author's book.

The statistics of Fifty Victorian Era Novelists Authorship Attribution data set:

Characteristics: Text

Number of Instances: 93600

Attribute Characteristics: N/A

Number of Attributes: 1000

The data set was downloaded in CSV (Comma Separated Values) format from the UCI (University of California Irvine) Machine Learning repository [1], donated by Abdulmecit Gungor in 2018 [2], and was used for a paper titled A Survey of Modern Authorship Attribution Method written by E. Stamatatos [3].

3.2 Preparing and Loading Data

To process the data set, this project has used Jupyter Notebook, a website-based interactive development environment for notebooks, code, and data [4]. Fifty Victorian Era Novelists Authorship Attribution data set is imported into Jupyter Notebook with Python 3 as the kernel. Before importing the data, I installed some libraries and imported some modules into the environment.

Libraries installed through pip:

- nltk (Natural Language Toolkit)
- wordcloud
- pyLDAvis

Modules imported:

- pandas
- numPy
- string
- nltk
- os
- re
- WordCloud from wordcloud
- simple preprocess from gensim.utils
- gensim
- stopwords from nltk.corpus
- gensim.corpora
- pprint
- pyLDAvis.gensim models
- pickle
- pyLDAvis

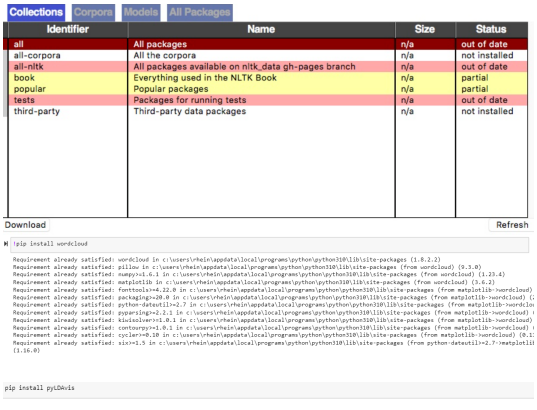


Fig.1. Installed Libraries

```
import gensim
from gensim.utils import simple_preprocess
import nltk
from nltk.corpus import stopwords
import pyLDAvis.gensim_models

import pickle
import pyLDAvis
```

```
In [6]: import re

from wordcloud import WordCloud

import gensim.corpora as corpora

from pprint import pprint
```

Fig.2. Imported modules

After preparing the environment, I imported the data set. Text is the content of the data and authors labeled from 1 to 50.

```
In [3]: mydata=pd.read_csv('/Users/rhein/Documents/VictorianTrainAuthor.csv',encoding = "Latin-1")

In [4]: mydata.head()

Out[4]:
```

	text	author
0	ou have time to listen i will give you the ent...	1
1	wish for solitude he was twenty years of age a...	1
2	and the skirt blew in perfect freedom about th...	1
3	of san and the rows of shops opposite impresse...	1
4	an hour s walk was as tiresome as three in a s...	1

Fig.3. Data set in Jupyter Notebook

3.3 Data Cleaning

Data cleaning is the process of fixing the data set by removing duplicates, incorrect formatting, and incomplete data.

3.3.1 Remove Columns

To focus on topic analysis, columns other than text data are removed. In this data set, the author column is removed.

```
In [5]: mydata=mydata.drop(columns=['author'],axis=1).sample(50)
        mydata.head()

Out[5]:
```

	text
28479	in the autumn the first month after their was ...
647	of no one else one that i would not think of a...
33397	conditions of people sympathetic and some tryi...
9781	on the u d of the golden tax of the op the fli...
34800	i should not have sung that last love song the...

Fig.4. After removing author column

3.3.2 Remove Punctuation and Lower Casing

Using a regular expression to remove punctuation, and lowercase the text for better analysis, and reliable results.

```
In [8]: string.punctuation

Out[8]: '!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'

In [9]: # Remove punctuation
        mydata['text_processed'] = \
        mydata['text'].map(lambda x: re.sub('[,\.!?]', '', x))

        # Convert the titles to lowercase
        mydata['text_processed'] = \
        mydata['text_processed'].map(lambda x: x.lower())

        # Print out the first rows of data
        mydata['text_processed'].head()

Out[9]: 28479    in the autumn the first month after their was ...
        647      of no one else one that i would not think of a...
        33397    conditions of people sympathetic and some tryi...
        9781     on the u d of the golden tax of the op the fli...
        34800    i should not have sung that last love song the...
        Name: text_processed, dtype: object
```

Fig.5. Removing punctuation and lowercasing the text data

3.3.3 Exploratory Scanning with Word Cloud

To check the data before preparing for LDA analysis, I have generated a word cloud. From the visualization, it is possible to spot some unimportant keywords that can be removed later.

```
In [13]: # Create a WordCloud object
         wordcloud = WordCloud(background_color="white", max_words=1000, contour_width=5, contour_color='black')
         # Generate a word cloud
         wordcloud.generate(long_string)

         # Visualize the word cloud
         wordcloud.to_image()

Out[13]:
```

Fig.6. Word cloud visualization before further data processing

3.3.4 Removing Stop Words

Stop words are a group of commonly used words in a language. It is used to eliminate unimportant words, granting analysis to focus on the important words.

```
print(stopwords.words('english'))

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him',
's', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom',
'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing',
'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during',
'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when',
'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too',
'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't",
'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn',
"needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]
```

Fig.7. Stop words library

Stop words can be extended based on the needs of the data set. From the exploratory scanning with word cloud, some unimportant keywords outside of the stop words library are spotted. Using the command stop words extend, common keywords are added into the library.

Stop words library extension consisting of: from, us, re, edu ,use, s, a (with an accent), ii, hy, mr, mrs, em, one, two, shall, would, could, bo, mo, till, said.

Fig.8. Removing stop words and some common keywords

3.4 Final Word Cloud Visualization

Word cloud is a visual representation of a text collection, in which the bigger and bolder the word appears, the more often the word is mentioned in the data [5]. Word clouds are great to get insights into patterns and popular words.

After processing and cleaning the data, the final form of the word cloud is visualized. These are three word cloud visualizations from three different topics of the data set:

Fig.9. Join the titles together and generate the final word clouds

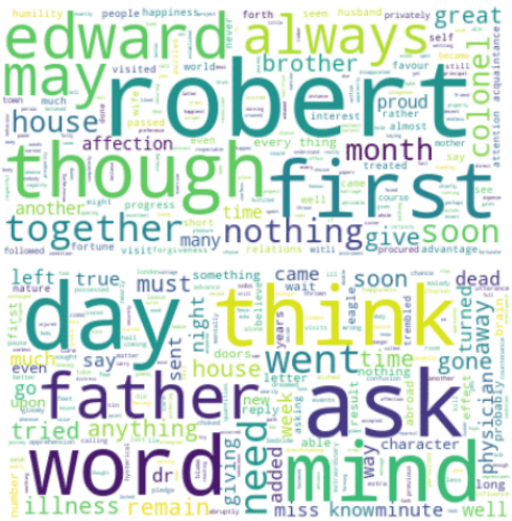
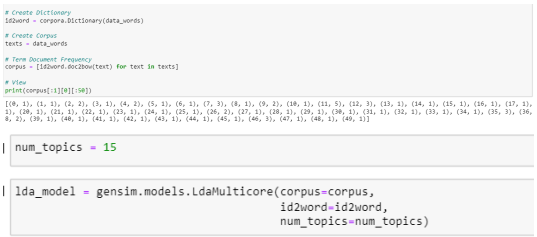


Fig.10. Word clouds from three different topics

3.5 Modeling With Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a form of statistical topic analysis from a mixture of texts [6]. LDA aims to find themes a data belongs to, based on the keywords it has.

LDA consists of two parts, the words within the data and the probability of words belonging to a topic, which are calculated through the algorithm. The algorithm attempts to determine the number of words belonging to specific topics and the number of documents belonging to a specific topic from some keywords. In this project, I have made LDA models with 15 topics.



```
pprint(lda_model.print_topics())
doc_lda = lda_model[corpus]
```

```
[(0,
  '0.004*"little" + 0.004*"man" + 0.004*"time" + 0.004*"much" + 0.004*"miss" + '
  '0.003*"sir" + 0.003*"know" + 0.003*"must" + 0.003*"upon" + 0.003*"never"'),
 (1,
  '0.005*"man" + 0.004*"time" + 0.004*"life" + 0.004*"like" + 0.004*"think" + '
  '0.003*"much" + 0.003*"say" + 0.003*"may" + 0.003*"little" + 0.003*"never"'),
 (2,
  '0.003*"life" + 0.003*"know" + 0.003*"like" + 0.003*"way" + 0.002*"see" + '
  '0.002*"never" + 0.002*"nothing" + 0.002*"even" + 0.002*"much" + '
  '0.002*"good"'),
 (3,
  '0.005*"like" + 0.005*"old" + 0.004*"man" + 0.004*"good" + 0.004*"think" + '
  '0.004*"never" + 0.004*"know" + 0.004*"little" + 0.003*"see" + 0.003*"time"'),
 (4,
  '0.006*"good" + 0.005*"well" + 0.005*"know" + 0.004*"think" + 0.004*"much" + '
  '0.004*"say" + 0.004*"like" + 0.004*"old" + 0.003*"never" + 0.003*"man"'),
 (5,
  '0.004*"man" + 0.004*"much" + 0.004*"like" + 0.003*"know" + 0.003*"time" + '
  '0.003*"well" + 0.003*"must" + 0.003*"life" + 0.003*"upon" + 0.002*"say"'),
 (6,
  '0.004*"upon" + 0.003*"men" + 0.003*"life" + 0.003*"see" + 0.003*"white" + '
  '0.003*"know" + 0.003*"little" + 0.002*"made" + 0.002*"time" + 0.002*"last"'),
 (7,
  '0.005*"life" + 0.005*"man" + 0.004*"old" + 0.004*"little" + 0.004*"much" + '
  '0.004*"time" + 0.003*"made" + 0.003*"know" + 0.003*"well" + 0.003*"men"'),
 (8,
  '0.005*"old" + 0.005*"good" + 0.005*"upon" + 0.004*"know" + 0.004*"like" + '
  '0.004*"see" + 0.004*"sir" + 0.004*"little" + 0.004*"men" + 0.004*"must"'),
 (9,
  '0.005*"sir" + 0.004*"like" + 0.004*"know" + 0.004*"old" + 0.004*"upon" + '
  '0.003*"man" + 0.003*"much" + 0.003*"life" + 0.003*"think" + 0.003*"tell"'),
 (10,
  '0.004*"know" + 0.004*"life" + 0.004*"like" + 0.003*"never" + 0.003*"old" + '
  '0.003*"upon" + 0.003*"come" + 0.003*"way" + 0.003*"good" + 0.003*"sir"'),
 (11,
  '0.007*"old" + 0.006*"like" + 0.005*"know" + 0.004*"man" + 0.004*"good" + '
  '0.004*"go" + 0.004*"well" + 0.004*"way" + 0.003*"got" + 0.003*"day"'),
 (12,
  '0.005*"like" + 0.004*"know" + 0.004*"man" + 0.004*"life" + 0.004*"well" + '
  '0.003*"time" + 0.003*"much" + 0.003*"old" + 0.003*"little" + 0.003*"good"'),
 (13,
  '0.005*"man" + 0.004*"like" + 0.004*"much" + 0.004*"nature" + 0.003*"life" + '
  '0.003*"world" + 0.003*"soul" + 0.003*"know" + 0.003*"god" + 0.003*"men"'),
 (14,
  '0.005*"upon" + 0.004*"know" + 0.004*"like" + 0.004*"life" + 0.004*"king" + '
  '0.004*"sir" + 0.004*"every" + 0.004*"old" + 0.003*"man" + 0.003*"little"')]
```

```
In [23]: M lda_model = gensim.models.ldamodel.LdaModel(corpus=corpus,
                                                      id2word=id2word,
                                                      num_topics=15,
                                                      random_state=100,
                                                      update_every=1,
                                                      chunksize=100,
                                                      passes=10,
                                                      alpha="auto")
```

```
pyLDAvis.enable_notebook()
vis = pyLDAvis.gensim_models.prepare(lda_model, corpus, id2word, mds="mmds", R=30)
vis
```

Fig.11. Making LDA model with 15 topics

Generating the model:

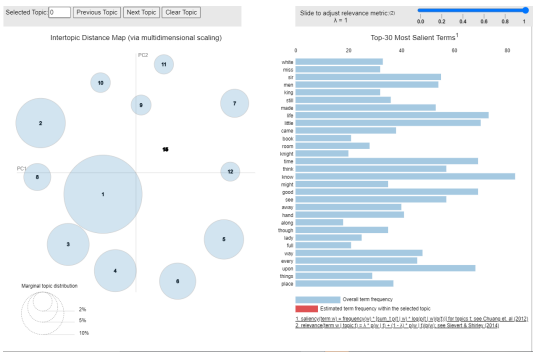


Fig.12. General distribution

Topic 1 word distribution:

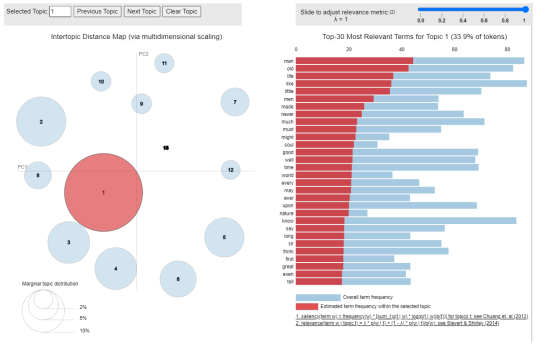


Fig.13. Topic 1 model

Topic 2 word distribution:

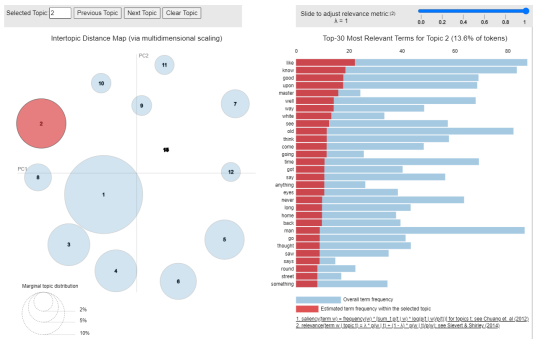


Fig.14. Topic 2 model

Topic 3 word distribution:

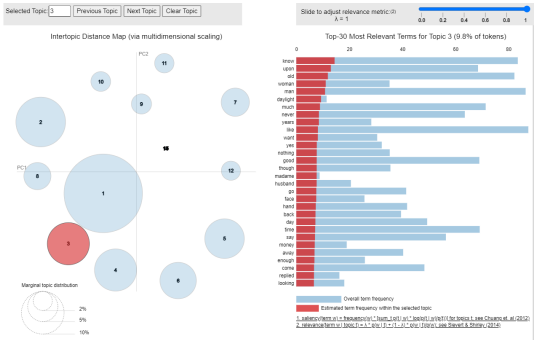


Fig.15. Topic 3 model

Topic 4 word distribution:

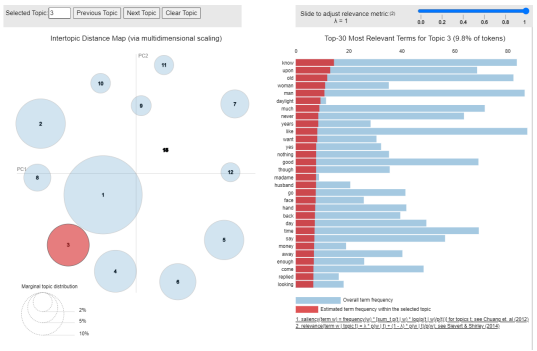


Fig.16. Topic 4 model

Topic 5 word distribution:

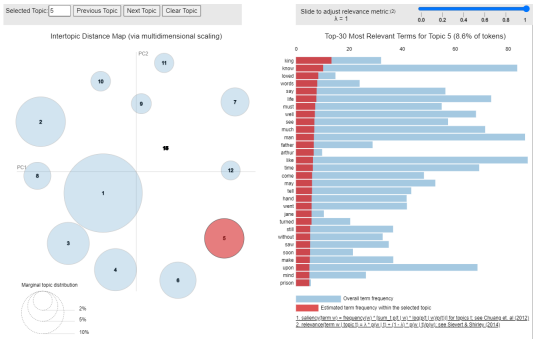


Fig.17. Topic 5 model

Topic 6 word distribution:

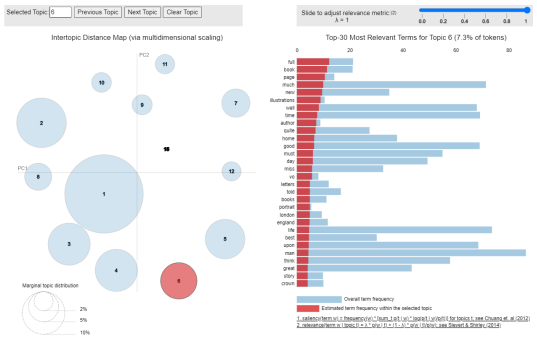


Fig.18. Topic 6 model

Topic 7 word distribution:

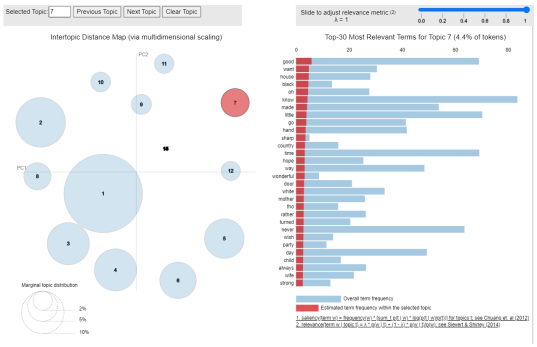


Fig.19. Topic 7 model

Topic 8 word distribution:

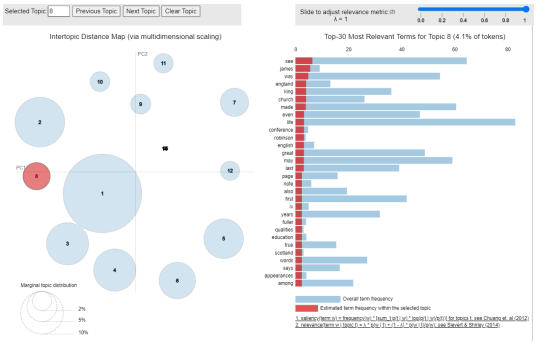


Fig.20. Topic 8 model

Topic 9 word distribution:

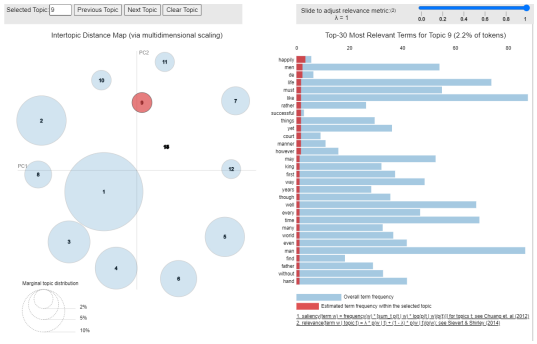


Fig.21. Topic 9 model

Topic 10 word distribution:

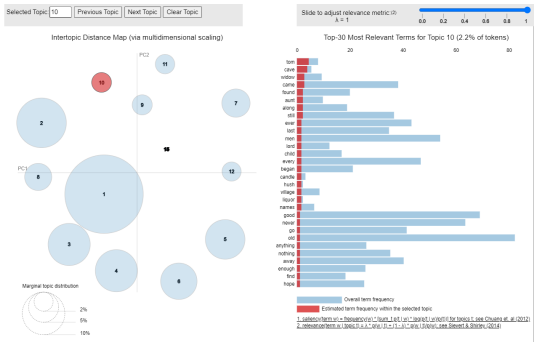


Fig.22. Topic 10 model

Topic 11 word distribution:

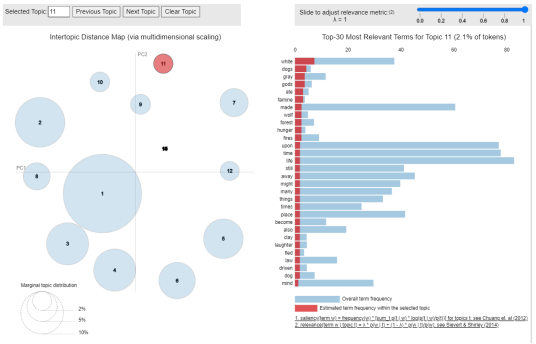


Fig.23. Topic 11 model

Topic 12 word distribution:

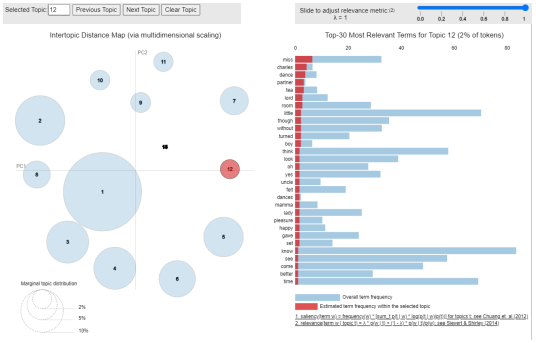


Fig.24. Topic 12 model

Topic 13 word distribution:

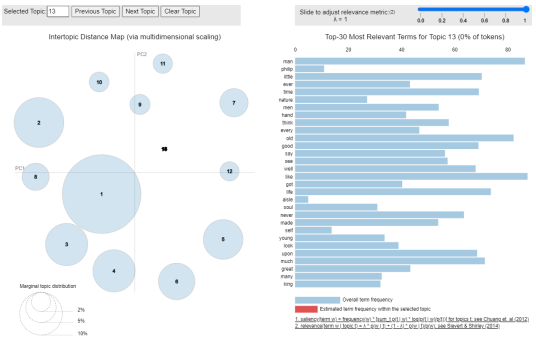


Fig.25. Topic 13 model

Topic 14 word distribution:

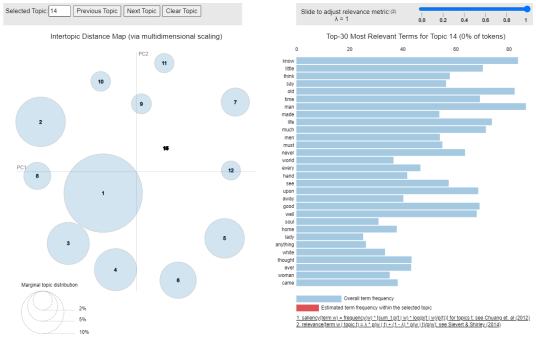


Fig.26. Topic 14 model

Topic 15 word distribution:

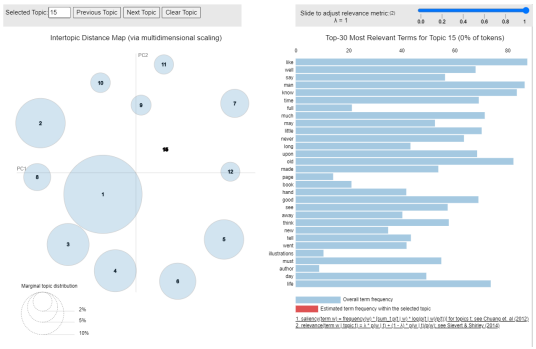


Fig.27. Topic 15 model

4 Findings

4.1 Data

The data used is Fifty Victorian Era Novelists Authorship Attribution with 93,600 instances, meaning there are 93,600 rows of text data containing works from fifty authors.

4.2 Word Cloud

Using word cloud library, 15 word clouds are visualized based on the model topics.

Word Cloud based on Topic 1:

Fig.28. Topic 1 word cloud

Word Cloud based on Topic 2:

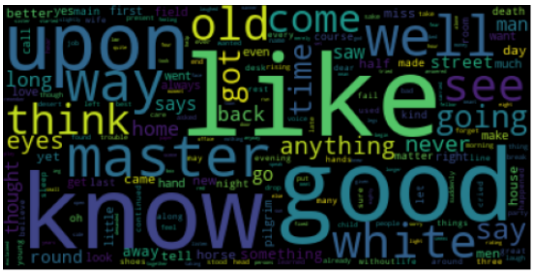


Fig.29. Topic 2 word cloud

Word Cloud based on Topic 3:

Fig.30. Topic 3 word cloud

Word Cloud based on Topic 4:

Fig.31. Topic 4 word cloud

Word Cloud based on Topic 5:

Fig.32. Topic 5 word cloud

Word Cloud based on Topic 6:

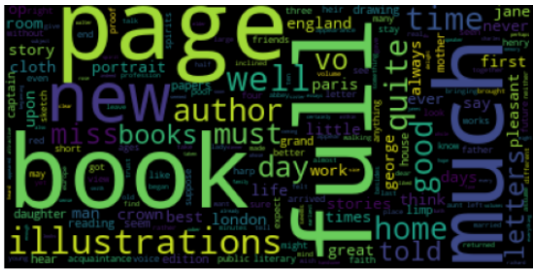


Fig.33. Topic 6 word cloud

Word Cloud based on Topic 7:

Fig.34. Topic 7 word cloud

Word Cloud based on Topic 8:

Fig.35. Topic 8 word cloud

Word Cloud based on Topic 9:

Fig.36. Topic 9 word cloud

Word Cloud based on Topic 10:

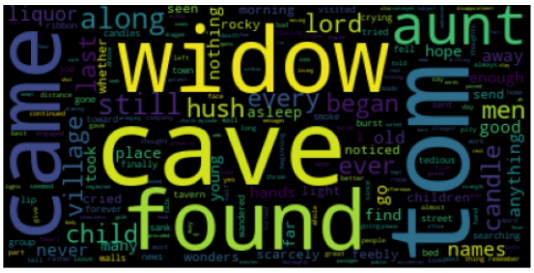


Fig.37. Topic 10 word cloud

Word Cloud based on Topic 11:

Fig.38. Topic 11 word cloud

Word Cloud based on Topic 12:

Fig.39. Topic 12 word cloud

Word Cloud based on Topic 13:

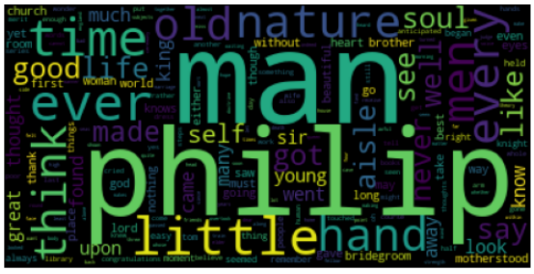


Fig.40. Topic 13 word cloud

Word Cloud based on Topic 14:

Fig.41. Topic 14 word cloud

Word Cloud based on Topic 15:

Fig.42. Topic 15 word cloud

4.3 Common Topics

From LDA modeling, a general distribution of the commonly used keywords is found:

white, miss, sir, men, king, still, made, life, little, came, book, room, knight, time, think, know, might, good, see, away, hand, along, though, lady, full, way, every, upon, things, place

In the top 30 commonly used words, the most commonly used word is 'know' and the least commonly used word is 'along'.

Fifteen topics are grouped based on the distribution:

• Topic 1 = man, old, life, like, little, men, made, never, much, must, might, soul, good, well, time, world, every, may, ever, upon, nature, know, say, long, sir, think, first, great, even, tell

The most commonly used word is 'man'.

• Topic 2 = like, know, good, upon, master, well, way, white, see, old, think, come, going, time, got, say, anything, eyes, never, long, home, back, man, go, thought, saw, says, round, street, something

The most commonly used word is 'like'.

• Topic 3 = know, upon, old, woman, man, daylight, much, never, years, like, want, yes, nothing, good, though, madame, husband, go, face, hand, back, day, time, say, money, away, enough, come, replied, looking

The most commonly used word is 'know'.

• Topic 4 = sir, knight, think, much, lady, know, anne, john, may, old, time, eyes, every, smith, letter, little, place, must, king, life, thought, friend, never, man, see, day, went, father, yes, things

The most commonly used word is 'sir'.

• Topic 5 = king, know, loved, words, say, life, must, well, see, much, man, father, arthur, like, time, come, may, tell, hand, went, jane, turned, still, without, saw, soon, make, upon, mind, prison

The most commonly used word is 'king'.

• Topic 6 = full, book, page, much, new, illustrations, well, time, author, quite, home, good, must, day, miss, vo, letters, told, books, portrait, london, england, life, best, upon, man, think, great, story, crown

The most commonly used word is 'full'.

• Topic 7 = good, want, house, black, oh, know, made, little, go, hand, sharp, country, time, hope, way, wonderful, door, white, mother, tho, rather, turned, never, wish, party, day, child, always, wife, strong

The most commonly used word is 'good'.

• Topic 8 = see, james, way, england, king, church, made, even, life, conference, robinson, english, great, may, last, page, note, also, first, ix, years, fuller, qualities, education, true, scotland, words, says, appearances, among

The most commonly used word is 'see'.

• Topic 9 = happily, men, de, life, must, like, rather, successful, things, yet, court, manner, however, may, king, first, way, years, though, well, every, time, many, world, even, man, find, father, without, hand

The most commonly used word is 'happily'.

• Topic 10 = tom, cave, widow, came, found, aunt, along, still, ever, last, men, lord, child, every, began, candle, hush, village, liquor, names, good, never, go, old, anything, nothing, away, enough, find, hope

The most commonly used word is 'tom'.

• Topic 11 = white, dogs, gray, gods, ate, famine, made, wolf, forest, hunger, fires, upon, time, life, still, away, might, many, things, times, place, become, also, clay, laughter, fled, law, driven, dog, mind

The most commonly used word is 'white'.

• Topic 12 = miss, charles, dance, partner, tea, lord, room, little, though, without, turned, boy, think, look, oh, yes, uncle, felt, dances, mamma, lady, pleasure, happy, gave, set, know, see, come, better, time

The most commonly used word is 'miss'.

• Topic 13 = man, philip, little, ever, time, nature, men, hand, think, every, old, good, say, see, well, like, got, life, aisle, soul, never, made, self, young, look, upon, much, great, many, king

The most commonly used word is 'man'.

• Topic 14 = know, little, think, say, old, time, man, made, life, much, men, must, never, world, every, hand, see, upon, away, good, well, soul, home, lady, anything, white, thought, ever, woman, came

The most commonly used word is 'know'.

• Topic 15 = like, well, say, man, know, time, full, much, may, little, never, long, upon, old, made, page, book, hand, good, see, away, think, new, tell, went, illustrations, must, author, day, life

The most commonly used word is 'like'.

4.4 Conclusion

In conclusion, the works of fifty 19th century English language authors consist of 93,600 rows of text data containing works from their published books. To create clear data for analysis, data cleaning is done by removing punctuation, lower casing, remov-

ing stop words, and removing some common unimportant words. Better data cleaning and processing create clearer analysis models. The works are modeled into 15 different topics in this project, with 'know' as the most commonly used word, and 'along' as the least commonly used word in the top 30 distribution. Some hypotheses can be made by using the models:

topic 1: Involving a man, and men. Has something to do with soul, world, time, and nature.

topic 2: Involving a master, and a man. Has something to do with white, and home. Street takes a big role as place.

topic 3: Involving a woman, a man, a madame, and a husband. Has something to do with money.

topic 4: Involving a man, a knight, a lady, someone named anne, someone named john, someone named smith, a king, a friend, and a father. Has something to do with letter, and life.

topic 5: Involving a king, a man, a father, someone named Arthur, and someone named Jane. Has something to do with loved, and life. Prison takes a big role as place.

topic 6: Involving an author, a woman, and a man. Has something to do with a books, illustrations, home, letters, portrait, life, and crown. London and England take big roles as place.

topic 7: Involving a mother, a child, and a wife. Has something to do with wish, part, and strong. House and country take bi roles as place.

topic 8: Involving someone named James, someone named Robinson, and a king. Has something to do with life, conference, page, note, and education. England, church, and Scotland take big roles as place.

topic 9: Involving men, a king, a man, and a father. Has something to do with court, manner, time, and world.

topic 10: Involving someone named tom, a widow, an aunt, men, a lord, and a child. Has something to do with candle, liquor, and hope. Cave and village take big roles as place

topic 11: Involving dogs, and a wolf. Has something to do with famine, hunger, life, and clay. Forest takes a big role as place.

topic 12: Involving a woman, someone named Charles, a boy, an uncle, a mamma, and a lady. Has something to do with tea, room, and dances.

topic 13: Involving a man, someone named Philip, men, and a king. Has something to do with nature, life, aisle, and soul.

topic 14: Involving a man, a lady, and woman. Has something to do with life, soul, and home.

topic 15: Involving a man, and an author. Has something to do with life, and illustrations.

Word cloud library is used to visualize the common words based on their frequencies. For images of the visualization, check the Word Cloud subsection in the Findings section.

References

[1] Univerity of California Irvine Machine Learning Repository. Victorian Era Authorship Attribution Data Set. Irvine: Univerity of California Irvine, 2018. Available at: https://archive.ics.uci.edu/ml, October. 2022.

[2] Gungor A. Characteristic Functions on Graphs: Benchmarking Authorship Attribution Techniques Using Over A Thousand Books by Fifty Victorian Era Novelists, Purdue Master of Thesis, 2018.

[3] E. Stamatatos. A Survey of Modern Authorship Attribution Methods. In *Journal of the American Society for Information Science and Technology*, 2009.

[4] Kluyver T, Ragan-Kelley B, Fernando, erez, Granger B, Bussonnier M, Frederic J, et al. Jupyter Notebooks, a publishing format for reproducible computational workflows. In *Loizides F, Schmidt B, editors*, Positioning and Power in Academic Publishing: Players, Agents and Agendas, 2016, 87-90.

[5] Oesper, L., Merico, D., Isserlin, R., Bader, G. D. WordCloud: a Cytoscape plugin to create a visual semantic summary of networks. In *Source code for biology and medicine* , 6(1), p.7, 2011.

[6] Blei, D. M., Ng, A. Y. Jordan, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 2003.