Text Analysis for Detective Stories

Yun-Hsiang (Ray) Chan

I. Introduction

In the EDA section, we have already conducted a complete investigation on the components of detective stories from readers' annotations. However, there isn't much interesting insight show up in the EDA section.

Instead of further proving the obvious conclusions from EDA part, we decide to analyze these stories from the text side, try to dig into text features of detective stories and see whether there's any interesting fact.

One of the research questions our collaborators want to know is the evolution of detective stories. Our statistical analysis will focus on this part, try to see whether there's any special features from text between detective stories from different decades or years.

If there's nothing that can distinguish stories between decades, then we might want to see what is the key components differentiating the text features from different stories.

To do the text analysis, we will introduce the idea of **tf-idf score** for each word in text as the text feature of each document. Then we will implement the **K-means clustering algorithm** to fit these features and assign clusters for different stories. Lastly, we will plot the clusters out as a preliminary insight, and do hypothesis testing to figure out the potential feature that determine the clustering result.

II. Assign clusters for detective stories and Preliminary Investigation

In this section, our goal is to assign clusters to detective stories based on their content.

There are 3 main steps:

1. Use tf-idf scores for each word (excluding some rare and common words) as the representation vector for each detective story.

2. Implement K-means clustering algorithm for assigning the clusters.

3. Plot out the clustering result in a low-dimensional space, as a preliminary of next step.

These 3 steps will be dismantled into 6 subsections.

We will introduce the idea, the implementation details and the conclusion from all steps above.

2.1 Introduction to the idea of tf-idf

The tf-idf score, short for **term frequency-inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection of corpus. It's often used as a weighting factor in searches of information retrieval, text mining, and user modeling.

The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general.

The formula of tf-idf score of term t in document d is:

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

$f_{t,d}$ is the number of times that term t occurs in document occurs in document d

Intuition:

- If a term occurs less frequently among all documents while occurring many times in this specific document, it indicates that this term is valuable to this document.
- For example, if the word "doctor" appears 100 times in this story and it only appears in 2 different stories, the tf-idf score is 50. The other word, "knife", appears 10 times in this story and it appears in 10 different stories, the tf-idf score is 10.
- This indicates that "doctor" is a comparatively crucial word than "knife" to this story.

(Reference: https://en.wikipedia.org/wiki/Tf%E2%80%93idf)

2.2 tf-idf scores as representation vector

After introducing the idea of tf-idf, it's time to introduce how can we use them as a representation vector.

Popular method in information retrieval

Since tf-idf scores for each word can represent how important the word to this story, by forming them as a vector, it could be a representation learned from the text of the story.

Implementation Details

To do that, for each document, we need to calculate the tf-idf score for every word in the document, and use these scores to form a vector.

To eliminate the negative influence of some rare words or common words, we set the threshold for adding the word to the representation vector.

- If a word appears in more than 50% of the documents or less than 80% of the documents, then the tf-idf score for this word in every document will be included in the representation vector.
- Otherwise, the tf-idf score for this word won't be included.

The reason to set threshold is to reduce the noise of common word like 'he' or 'she', and the noise of rare terms that might appear only in a specific type of stories (e.g. Holmes). The 50% and 80% are picked based on empirical experience.

We hope that by reducing these noises, the vector can capture the pattern of commonly occurred verb or noun so that we can detect the topic or writing habits of a particular era.

At the end, we have a vector with 425 dimensions for each detective story, with each dimension represent the tf-idf of the term. Each term must appear in more than 50% of documents, or less than 80% of documents.

2.3 K-means clustering and Dimensionality reduction by MDS.

As we have the representation vectors, we can assign clusters for each story based on the vectors.

The reason we do this step is to figure out why some stories are assigned to a specific cluster in the next part.

Is it because of the publication year? Or it is due to authors, topics, or something we never notice before.

Implementation Details

The entire process was done in python. Comments are provided in Appendix 1.

We implement K-means clustering to assign the cluster and draw the plots by:

1. Prepare the tf-idf matrix, where each row represents a detective story, and each column represent the tf-idf score for a unique word.
2. Calculate the distance between each row vectors by cosine similarity and form a matrix.
3. Choose the number of clusters, and apply the K-means algorithm on both tf-idf matrix and distance matrix.
4. Reduce the dimension of distance similarity to 2 by Multidimensional scaling (MDS). This is a technique to visualize the level of similarity between vectors.
5. Draw the plots with publication year and author code as a preliminary insight.

Briefly Introduction to the methods

- K-means clustering is a method of vector quantization, that aims to partition n observations into k clusters, wehre each observation belongs to the cluster with the nearest mean.
- MDS is a means of visualizing the level of similarity of individual cases of a dataset. It's used to translate "information about the pairwise distances among a set of n objects" into a configuration of n points mapped into an abstract Cartesian space (2-dimensional space).

(Reference 1: https://en.wikipedia.org/wiki/K-means_clustering)

(Reference 2: https://en.wikipedia.org/wiki/Multidimensional_scaling)

Why doing clustering on both tf-idf matrix and similarity distance matrix?

- The similarity distance matrix was originally computed for visualization. However, we find something interesting when doing the clustering.
- The clustering result will be mostly the same when applying k-means clustering on tf-idf and distance matrices. But those tiny differences raise our interest.
- We believe both type of matrix can capture some special patterns that the other one can't capture. This is why we keep the result of clustering on both matrices.

Why clustering and why K-means?

- The reason to do clustering is to differentiate the stories by text feature vectors so that we can find out the reason behind cluster. Clustering is one the most popular unsupervised methods for pattern recognition. This is the reason for choosing it.
- In terms of clustering methods, we also try the hierarchical clustering method. The hierarchical clustering method will produce a similar clustering result for distance matrix, like the one we can obtain by K-means clustering, while it produces an inconsistent result for tf-idf matrix clustering. Therefore, to compare the clusters produced by two matrices at the same time, we believe K-means is more suitable in this case.

2.4 EDA

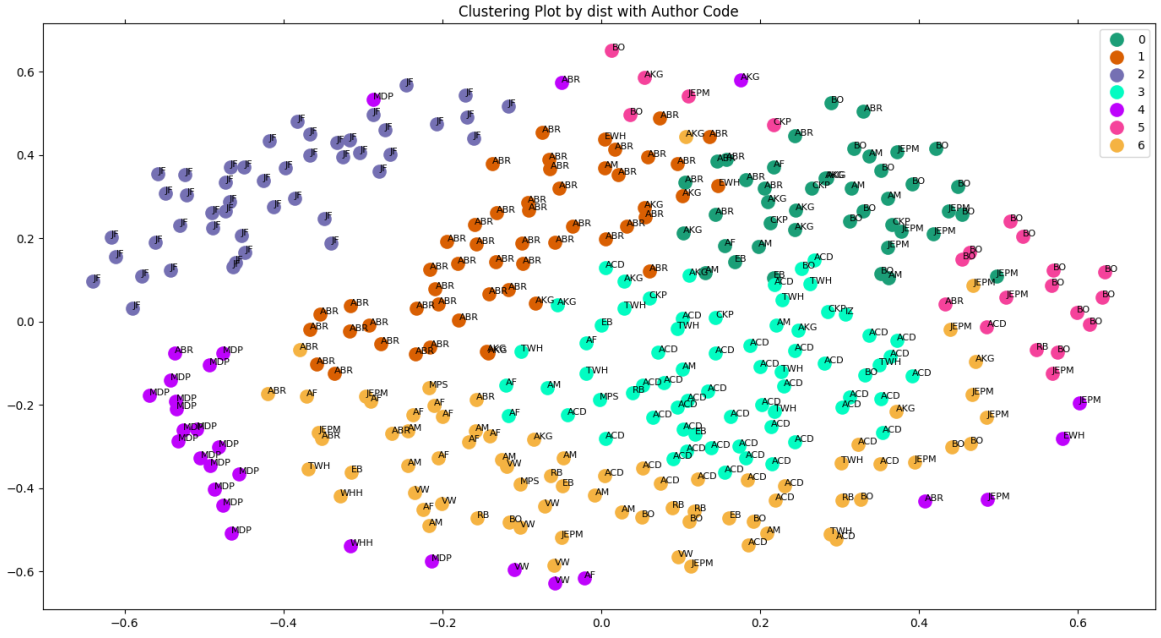
The two plots are clustering result on tf-idf matrix or distance matrix with author code.

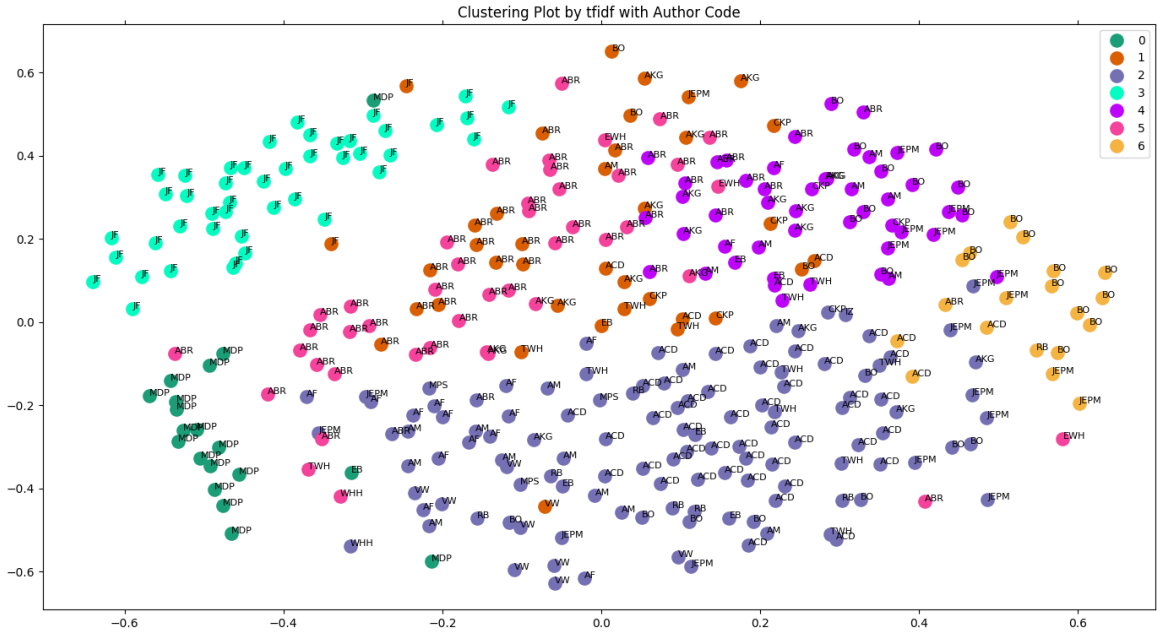
In these plots, each point represents one story, and each color represents the cluster.

Note:

- Empirically, we select $num\_clusters = 7$ for both distance matrix and tf-idf matrix clustering.
- The x and y axis in the plot have no actual meaning. They are only calculated to measure the distance between different stories.
- The position of each story in the plot are the same because they are both calculated on the distance matrix, while the color will be different because of clustering on different matrices (tf-idf or distance).
- The number of cluster will be slightly different by orders. This is because the clustering are done separately.

We first look at the clustering plot by distance matrix and tf-idf matrix with author code information.





From these two plots, we can know that although we really hope that the clustering of stories can rely on different eras, **it's more obvious that stories from the same author are more likely to be assigned to one cluster.**

For example, the stories from JF (Jacques Furtelle) are likely to be assigned to one cluster (cluster 2 in plot 1 and cluster 3 in plot 2). It's reasonable that they are classified in one cluster when the writing habits of the author is unique. Moreover, the topics of these stories are similar (most of them are from the series of detective stories, thinking machine).

Other clusters based on authors are stories from ABR (Arthur B. Reeve) or MDP (Melville Davisson Post) by distance matrix clustering.

2.5 Interesting Facts

There are still some clusters are assigned with other reasons, and these reasons are not publication years.

Note: the plots with publication year information are provided in the Appendix 2. There are also comments for justifying my claim.

Fact A: Cluster 1 and 5 in tf-idf matrix clustering

One interesting fact is that, while stories from ABR (Arthur B. Reeve) are assigned to the same cluster by distance matrix clustering, they are divided into different clusters by tf-idf matrix clustering. And from empirical experiments, if we choose $n\_clusters = 6$ (now it's 7), then these stories won't be divided into different clusters by tf-idf matrix clustering.

This indicates that there are some crucial text features in tf-idf matrix that could be used to distinguish between cluster 1 and 5 in plot 2, while these features can't be captured by similarity distances.

Fact B: Cluster 3 and 6 in distance matrix clustering

The other interesting fact is that, while cluster 2 from tf-idf matrix clustering (plot 2) looks like a combination of all remaining stories (this is even more obvious when *n_clusters = 3* or *4*), the similarity distance clustering method (plot 1) can assign different clusters for them (cluster 3 and cluster 6).

This indicates that there are some crucial features in similarity distance matrix that could be used to distinguish between cluster 3 and 6 (plot 1), while these features can't be captured by tf-idf features.

2.6 Next Steps

Therefore, in the next few sections, we will investigate:

1. What is differentiating cluster 3 and cluster 6 in plot 1 (similarity distance matrix clustering)?
2. What is differentiating cluster 1 and cluster 5 in plot 2 (tf-idf matrix clustering)?

Implementation

We plan to conduct several hypothesis testings to see whether these clusters are assigned due to similar topics.

If there's no explicit reason from the annotated dataset, then the authenticate logic behind the classification should be left for the experts of detective stories, our collaborators.

Other choices and limitations.

- It's also viable to look at the most influential column values (i.e. the tf-idf scores for some words) contributes to each cluster after tf-idf matrix clustering, and try to find out the reason from the word.
- However, these influential tf-idf scores are often noisy. There might exist some patterns when scanning them altogether, but these patterns are hard to be observed individually.
- Therefore, as we already have annotation of each story, we instead take the annotations for investigation.
- You can also find the most influential words (top 10) to each cluster for tf-idf matrix clustering in Appendix 3.
- Similarity distance clustering do not have these words, since the dimension is changed when calculating the matrix from tf-idf matrix and therefore the dimensions don't have actual meanings.

III. Statistical Anlaysis

In this section, we will mainly focus on hypothesis testing. We want to testify whether the implicit logic behind the grouping. For two the groups of clusters, are they assigned because they have similar topics or written in different eras?

Fisher's Exact Test of Independence

For examining the topics of two groups (cluster 3 / 6 from similarity distance matrix clustering, and cluster 1 / 5 from tf-idf matrix clustering), we can will specifically focus on the crime type and motive in the stories. These two components are the most representative annotations in our dataset.

Let's first look at the two tables below.

Crime.Types	Cluster.3	Cluster.6	Cluster.1	Cluster.5
Murder	43	36	24	33
Suspected.murder	10	7	1	4
Theft	30	42	19	14
Fraud	33	32	17	19

Crime.Types	Cluster.3	Cluster.6	Cluster.1	Cluster.5
Blackmail	19	5	5	4
Bribery	0	3	0	0
Assault	29	21	14	17
Forgery	3	15	5	8
Kidnapping	13	4	4	5
Mischief	11	8	7	8
Breaking.and.entering	8	13	8	6
Trafficking	0	1	0	2
Illegal.gambling	0	1	1	0
NA.	1	2	1	1

Motive.Types	Cluster.3	Cluster.6	Cluster.1	Cluster.5
Greed	47	52	23	29
Revenge	23	14	8	7
Jealousy	11	3	7	9
Love	8	3	6	4
Ideology	2	12	4	7
Pride	17	10	4	6
Duress	0	0	0	0
Crime.for.crime.s.sake	3	2	3	0
Paranoia	3	2	2	0
Self.defence	7	3	1	0
Insanity	0	1	2	3
NA.	2	3	0	2

From these two tables, we can know that for either group, the two clusters are not coming from a particular crime type or motive type.

Therefore, we would like to use Fisher exact test of independence to check whether the distributions of crime and motive types from these two clusters in each group are independent.

The p-value for the test of each group are below.

Group	p.value.for.crime.distribution.test	p.value.for.motive.distribution.test
Cluster 3 and 6 from dist clustering	0.001	0.02
Cluster 1 and 5 from tf-idf clustering	0.870	0.50

From the p-value above, we can conclude that:

1. Overall, stores in cluster 3 and 6 from distance clustering are more likely to have different topics.

- The p-values for the test on crime and motive type distributions of cluster 3 and 6 are smaller than 0.05.
- It's a strong evidence against the null hypothesis that these two groups are from the same distribution (topic).

2. Overall, stories in cluster 1 and 5 from tf-idf matrix clustering are more likely to have similar topics.

- The p-values for the test on crime and motive type distributions of cluster 1 and 5 are larger than 0.05.

- We fail to reject the null hypothesis that these two groups are from the same distribution (topic).

Choice of testing methods

Why choosing fisher's exact test

- We want to see whether they two clusters are having similar topic.
- It's hard to define topic by the annotations we have. However, we can look at the overall distribution of crime and topic as an overview of topics in the cluster.

Why not choosing chi-square test?

- The reason for not doing chi-square test is simple. Chi-square test is more useful when the sample size is large, while here these clusters are specifically assigned from our the small-size sample.
- Therefore, fisher's exact test is more suitable under this case.

In-depth Interpretation of test result

- If the clusters are from similar distribution, it indicates that there might be some commonly appeared word (e.g. police) in these particular topics.
- If the clusters are from different distribution, it indicates that there might exist some other reasons (other linguistic features) that push the clustering algorithm to divide them.
- Either result could become the next research question for our collaborators.

Next Steps for the collaborators

The analysis is ended in this part, since further investigation is too difficult without the domain knowledege of detective stories.

Here are some questions that could be investigated further by our collabortors:

1. The two clusters have different topics, then what are the implicit logic behind the assignment of cluster 3 and 6?

What elements (either components of stories or linguistic features) lead to the special representation vectors that could be classified into different groups?

2. What are the explicit language features in those specific topics from cluster 1 and 5?

Topic is a broad definition here. Can we define them in a more specific wat? What are the key factors lead to the final assignment?

IV. Explanation of Doing an Unconventional Analysis & Limitations

The main component of this analysis relies on the idea of text analysis and the choices of method to conduct the analysis (tf-idf, clustering, dimensionality reduction for visualization).

These methods (e.g. clustering) include some statistical component, while it's not the traditional testing or modeling expected in a general statistical analysis. This is mainly because of the characteristic of text data that I want to analyze.

I choose this topic as a challenge of learning new things and conduct insightful analysis from the text of detective stories, rather than the annotated dataset.

The design and choice of hypothesis test in the analysis part is not completely reasonable. The test itself and some assumptions might not be suitable for the research question. This is mainly because of the unconventional dataset we have and the difficult question that we want to investigate.

Even if I feel like they are not absolutely reasonable, I still try to put some traditional statistical analysis component to form it as a report that is originally expected in STA490.

However, I still believe that to obtain a more meaningful text data analysis, it's better to communicate and cooperate with our collaborators for in-depth understanding of the stories (those clusters) rather than doing the testing in this analysis.

I hope my decisions wouldn't be criticized and hurt the grade of this project.

Appendix 1: Data Preparing Process

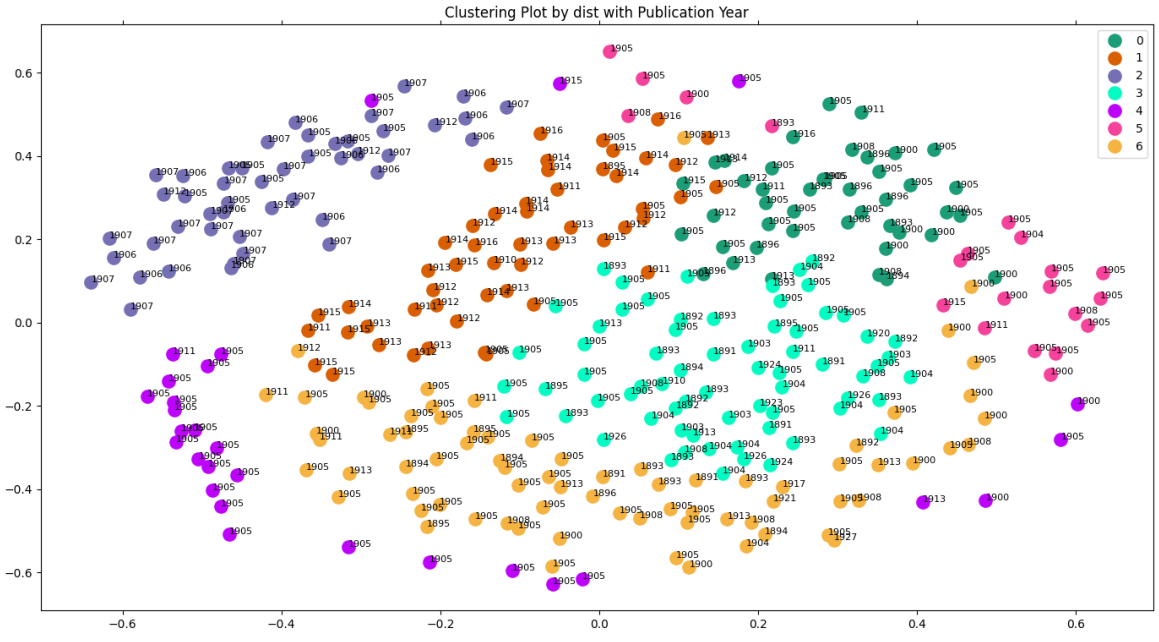
To deal with the text data, we utilize python instead of R. I upload all python code, R code, produced csv and plots in github repository (https://github.com/rachan1637/detective).

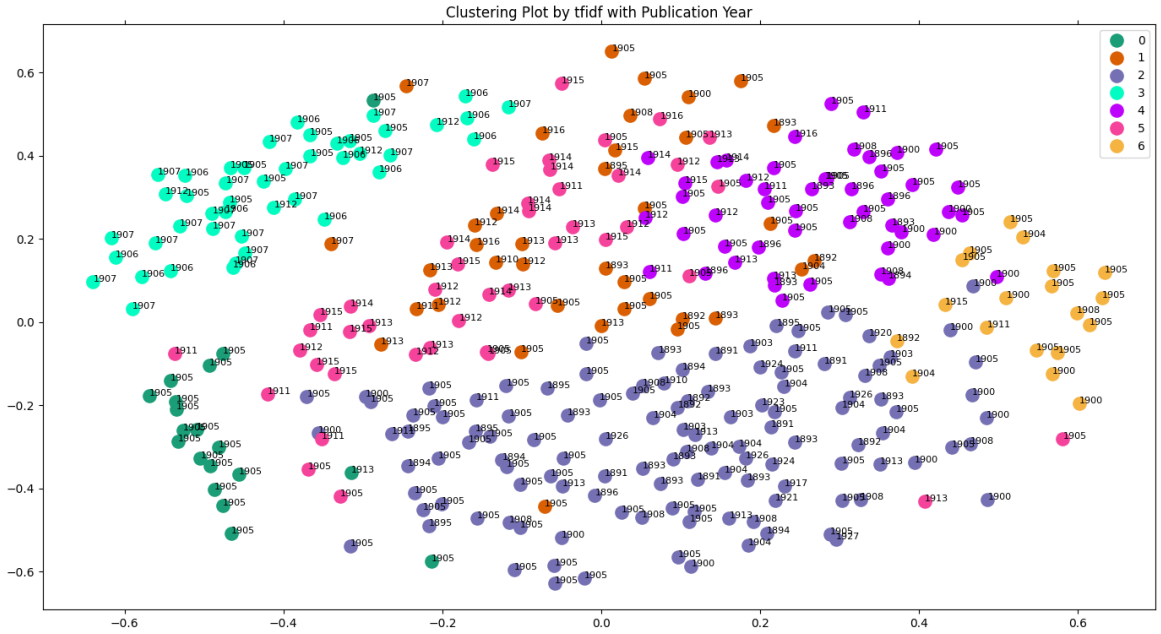
The preprocessing includes the text data normalization and partitions. The partition-by-reveal-border process was initially designed for doing sentiment analysis, which was mentioned by the collaborators. But since the target for this task is vague, and unsupervised sentiment analysis is harder than expected, we give up the plan eventually.

You can do 'python {path}/main.py' in the terminal / bash to run the process of doing preprocessing, tf-idf counting, clustering and plotting. The path inside the python program might be changed, depending on the directory root that you run the program.

I was too busy in these few weeks. As the grading rubric doesn't ask for a cleaned python code, I didn't add much annotations and clean my python code for your convenience. However, I am willing to do that if the TA, the instructor or the collaborators want to trace my code in future.

Appendix 2: Clustering Plots with Publication Year information





Why we exclude pubilcation year as a reason of clustering:

- We know that the collection of the dataset is biased, only the detective stories that could be found on the Internet are collected.
- The reason that the clusters seem to be assigned due to the year is that many stories written by the same author were published in adjacent year, and all stories we have in some particular years are from only one or two authors. This conclusion can be proved by the plot with author information above.
- Therefore. The publication year should not be considered as the primary reason of clustering.

Appendix 3: Top words contributes to tf-idf matrix clustering

Table 4: Top-10 influential words for each cluster

Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5	Cluster 6
replied	miss	sir	thinking	mrs	doctor	lady
cried	father	friend	miss	woman	new	miss
big	woman	police	detective	lady	mrs	woman
dead	girl	london	didn	wife	quickly	girl
father	mrs	window	woman	miss	miss	dear
girl	lady	round	mrs	girl	replied	london
woman	window	paper	replied	police	remarked	police
sir	herself	street	girl	herself	air	sir
fingers	replied	father	clock	replied	body	wife
chair	paper	business	money	money	blood	herself