Public Policies and Public Opinions Behind New York Times News Articles in 1971, 1989, 2010, and 2020

— Investigating public-politics relationship through natural language processing

1 Background

In 1995, a popular theory went viral among criminology scholars, policy officials, and the general public. John J. Dilulio, a young political scientist and criminologist at Princeton, warned in a conservative newspaper that a wave of young 'super-predators' was forthcoming. The self-coined term represented juveniles who had cruelty and violence running in their DNAs and who committed crimes without remorse and were born to spread bloods over the streets.He predicted that by early 20 century, 'an additional 30,000 young "murderers, rapists, and muggers' would be roaming America's streets". His argument, although was not supported by any known findings on the causes of juvenile crimes, soon received its popularity among media and was published by some of the most known outlets, such as the Associated Press and the Chicago Tribune.

Dilulio's theory was brought up when the case of Central Park Five, in which five young black and Hispanic teenagers were (wrongfully) indicted on crimes of murder and rape, was still fresh in most Americans' memory. People were fearing that they would be the next victims and the call for tougher crime laws was unprecedentedly high, despite that statistically, the violent crime rates were going down. By exaggerating juveniles crimes as a national problem, the theory unsurprisingly fostered the passing of several toughest crime prevention policies on both state and national levels. We can easily recognize some of the names - three-strikes law, life incarceration without parole, and the trend of regarding juveniles as adult offenders and sentencing them in adult courts where their verdicts would be harsher. One and a half decade later, Dilulio's idea has been proved to be wrong, however, the consequences are irreversible. For instance, because of his theory, the incarceration number was at an all-time high and a lot of people were sentenced to a time longer than they deserved. Moreover, because of the social construction of racism, the catastrophic influence was especially falling on racial minorities.

I learned about the term 'super-predator' four years ago and the story behind it one year ago, but I am constantly reminded that politics, media, and public are closely connected that one individual's words, after being captured and operated by media, can trigger political responses, which then causes chained effects on the lives of ordinary individuals. Put it simply, news reports can shed light on what people care about and what people think, and politicians respond to the thoughts of their constituencies.

New York Times is one of the most reputable and influential newspapers in the nation whose readers are mostly young, well-educated, and upper-class white people, although its readers come from all social spectrum. The broad reader base of NYT makes the journal a great place to look into what people were thinking in years important to the history as well as understand how people's attentions have changed over years. I was able to scrape the NYT news archive data from the NYT archive database. From there, I picked four years when groundbreaking criminal justice incidents, policies, or Supreme Court verdicts took place. This is the starting point for my analysis.

2 Method

The whole project can be divided into several sections, starting with data gathering and ending with data modeling with deep learning.

2.1 Data Gathering

The original NYT archive database includes all NYT news data from 1920 to 2020. Because of the large data size, I selected only four years to focus on. These four year respectively represent an influential criminal justice policy/case/event:

- 1971, the beginning of the war on drug campaign by president Nixon. War on drug, though it might come with the intention of stopping the propagation of drugs, developed to be a new age of Jim Crow and caused systematic discrimination of Black people.
- 1989, the case of *Connor vs Graham*. *Connor vs Graham* is an influential Supreme Court case on the Fourth Amendment. The Graham Principles that came with the SCOUTS rulings, ruled when and where a police officer can adopt force.
- 2010, the year when mass incarceration came to its highest point in the US. According to a graph presented by the Prison Policy Initiative, in 2010, the number of people imprisoned behind the bar reached a climax.

- 2020, the year when the killing of George Floyd by Minneapolis Police Officers occurred. As the most prominent police violence case in recent years, the death of Mr.Floyd has led to a series of social changes, including the Black Lives Matter movement that broke out in different cities on a scale that was not seen before.

For each year, the data contained two columns: year and text. The text or sentence column represents all news reports in that year, and each row represents a news article published by NYT. I moved to the next phase with initial datasets.

2.2 Data Cleaning

The scraped data is usually not directly ready to use, for which we will need clean our data. In a natural language processing, pre-processing usually include steps of converting words into lowercase, removing punctuation, removing numbers, removing stop words, and dealing with missing data by discarding NAs or imputing values. The concept of stop words, which are usually conjunctures and some common nouns, is especially salient in working with language/text data. These words usually are impoverished semantically for which adding them would not only contribute little to out analysis but also hide words that are meaningful. Beyond, I also include steps to remove all single-character or bi-character words, for example, "is", "wa", or "j". The reason is that words that are short length usually don't reveal much information and that we want to dig out the real "treasure" behind texts. Some NLP analysis also take stemming and/or lemmatization, which removed words based on their linguistic origins, into accounts, but I didn't do it here as I believe in most cases, word origins (eg "ly", "ish", "un") could also tell us information for which they shall be preserved.

Texts that have gone through these pro-processing steps are in the form of strings, and for our exploratory analysis, we will want lists of strings instead. Therefore, my next step was dividing the strings into tokens/single words. Because I care about both words that are grouped by single words(tokens or unigrams) and words group by two words (bigrams), this step include two small steps, tokenize the large string and bigrammize the large string. The concepts of token and bigrams/n-grams are hard to explain in simple words but easier to understand when putting them in contexts. Taking this sentence as an example: "President Nixon issued the order", tokenizing it will lead to a list of ["President", "Nixon", "issued", "the", "order"] and bigramming it will lead to this list instead ["President Nixon", "Nixon issued", "issued the", "the order"].

Lastly, for the simplicity of my future analysis, I also rearranged the list of tokens and list of bigrams in each data-set so that for each year, I am getting two large lists, one with all the tokens in that year and one with all the bigrams. I then concatenated these lists together together, so I am getting a big data frame of 4*5, where four rows represent four years and five columns represent year, list of words, counts of total words, list of bigrams, and counts of total bigrams. With the fully cleaned data, I am interested in conducting some exploratory analysis to find out the most frequent words and bigrams in each year and how has the popularity of certain key words varied across time.

2.3 Exploratory Data Analysis

One important goal of any data science projects is digging out information that underlies the data, and one way of doing so is through Exploratory Data Analysis or EDA. EDA, as compared to data modeling, focuses on exploring data through graphics and summary statistics more than drawing inferences. For numeric data, EDA can be achieved through finding and mapping out the mean, median, standard deviation, or cross tabulations. For textual data, we instead focus on seeing the most frequent tokens/grams.

My goal of this phase was two-fold. First, I want to find out the most salient words and grams as measured by the number of times a single word popping up in documents. Moreover, because my issue interest lies on criminal justice and k12 education, I defined two sets of keywords that are common in this two fields. I then mapped out how the frequency of occurrence for words in each set varied across 1971, 1989, 2010, and 2020. Because both goals have year as the standard unit, I used the concatenated lists/data frames I acquired in the last section as the source of data.

Depending on the goal, I utilized different graphics to visualize the information. For the first goal, word clouds and bar charts were the best choices. For the second goal, I used line charts to show changes in frequencies of words/bigrams. This step requires more data wrangling efforts. Additionally, python's collection library has tools to get the frequency counts, which turned out to be really helpful.

2.4 Modeling - Clustering and Word Embedding

In the last section, I modeled and clustered the data. For clustering, I used K-means as the tool. For modeling, I used deep learning to automate the whole process. I noticed from the line charts that both crime-related and education-related keywords popped up most frequently in NYT reports in 1971. For this reason, I decided to use a sample of 2000 observations from the NYT 1971 data frame. Data sampling is necessary, considering the time it will take to run the whole data, which has 324893 observations in total (doing all actually exhaust all memories). With randomly sampling, I was able to preserve the statistical power and that the sample is a random representation of the whole dataset.

2.4.1 Clustering with tf-idf and Word2Vec

Because the NYT data does not come with a target attribute/column that is categorical or pairwise, I adopted non-supervised machine learning techniques, specifically, clustering. Simply put, clustering is grouping data points into a few clusters based on some similarity measures. There are multiple clustering techniques, and here, I am using K-means clustering, which is the simplest but a very powerful one. The k-means algorithm started with choosing k points as centroids of clusters, after which all data points are assigned to clusters that are closest in distance. Centroids are then recalculated until a convergence is met, usually defined as the smallest sum of square error. Sum of square error is the sum of square of distance between one data point and a centroid for all data points and all centroids. Mathematically, it has the form of:

$$SEE = \sum_{j=1}^{k} \sum_{x} d(x,m)^2 \tag{1}$$

K-means is the most form when the data has the linear form and is not skewed. It has the advantages of time efficiency and avoiding overfitting, however, it also has the pitfall of strong outlier sensitivity. For the data, I decided to use K-means despite that there are some outliers as we can see from the scatter plots. This is because the simplicity of K-means algorithm, which makes it a great pick.

The clustered data has a high dimension, for which I then ran PCA (Principal Components Analysis) to lower the dimension of data. PCA is a simple approach that generates several hyper-planes, on which each hyperplane or each PC represents an analysis. The algorithm also captures the internal relationship between attributes through the use of eigenvectors and eigenvalues. According to this Medium post, one biggest strength of PCA as contrast to other techniques, for example, t-SNE, is that it treats all clusters at a whole (i.e. global clustering) and that it costs the least time.

Because computer recognizes numbers rather than texts, the last important concern in this phase is text representation. There are three common approaches to do this: bag of words (bow), tf-idf, and Word2Vec. I practiced with the latter two.

1. The frequency approach - Tf-df

   Tf-idf stands for the collection of Term frequency (tf) and Inverse document frequency (idf). Usually treated as an augmented/improved version of simple bag of words, tf-idf highlights the frequency of one word both in one document and across documents that the more times one word appears in one document and the fewer number of documents it appears on, the higher the tf-idf score. By taking weights into accounts, the approach adds some semantic meanings to traditional bag-of-words, which in some cases, can be really helpful. Mathematically, two indexes can be expressed as:

$$tf(t,d) = \frac{N_t}{N_d} \tag{2}$$

$$idf(t,d,D) = \log \frac{N}{N_{tind,dinD}} \tag{3}$$

   where t is the term, d is the document, D is the corpus (collection of documents), N_t is the number of times t appear in document d, N_d is the total number of words in document d, N is the total number of documents, and N_{dinD} is the number of documents where word t appears.

2. The word embedding approach - Word2Vec

   After the clustering, I realized that this approach might not be ideal as some clusters overlap and the shape of clusters are not as I expected. Moreover, considering that the tf-idf has the shortcoming of filtering out words' semantic meanings in the encoding process (though an improvement than bag-of-word), I decided to try out with Word2Vec, which was developed by Google in 2013.

   Word2Vec is probably the most prominent one in word embedding. This approach considers if one word is more likely to come before or after another word, therefore, include more features. Scores are calculated by looking at the cosine similarity between words, and the larger the cosine similarity score, the more similar two words are. Word2Vec model can be based on either CBOW or skip-gram algorithm. By default, CBOW is the default one in Python, which predicted the targeted word using its neighbors (words surrounding the targeted word). The theory behind Word2Vec is more complex, however, the intuition is simple.

2.5 Deep learning with Tensorflow

Lastly, I modeled the data with deep learning. Such approach usually requires a target attribute, which is not presented. Therefore, I created a column that represents if the news article (i.e. row) includes any of the crime or education related keywords, which again, are self-defined. Additionally, I set the training/testing ratio to be 0.3 in the cross-validation phase so that the model can model a larger dataset with a small one.

With tensorflow, I utilized a sequential model in which each layer comes on top of each other. In this design, each layer passing to the sequential constructor must only have one input and one output. The model was ran over 15 epochs, and in each epoch, I retrieved the model loss and accuracy statistics, which are used to measure the performance of model and construct visualizations.

3 Results and Analysis

3.1 EDA

The final goal of this project is to model the NYT archives data with some models, however, to make that happen, we must first understand the data. That makes EDA important as previously stated. For the reason that media report is often closely related to public opinion and public policy, I am especially interested in knowing what people were thinking in those four years. Reflecting on our data, that is to ask what the most frequently used words by news reporters were in those four years. This question can be answered via our word clouds and bar charts.

But before that, it is necessary to take a look at our data. The cleaned 1971 data can be seen in the Appendix section, table 1. As we can see, the data has four columns representing original text, cleaned text, tokens, and bigrams. The cleaned text column is important in our modeling, while the tokens and bigrams columns matter for our exploratory analysis.

Appendix section table 2 is the concatenated, cumulative table that was mentioned at the end of section 2.2. This table contains all words and tokens from articles published in that specific year. As we can see here, fewer articles were included in the year 2020 database, and this could be either because NYT published less news articles or that the database on their website is incomplete. Compared to the number of tokens in years 1971 and 1989, year 2010 also has fewer news articles but not that dramatically as year 2020. This may influence the validity of my analysis but that is beyond my control.

The counter function from the collections library offers us a shortpath to get the word frequency. For year 1971, the top 5 words and their respective counts are:
’says’, 16531
’new’, 14766
’today’, 8378
’city’, 7453
’state’, 7212
I noticed that a lot of words from the list actually don’t make sense. For example, we can read little from the word "news", which is common as labels and also in the name of the journal. Similarly, "says" as a verb can have different meanings depending on the context, however, leaving this word out without knowing what’s before and/or after the word is simply vain. Based on this finding, I added more words and bigrams to the list of stop words, which will be used as a filter in our making of word clouds.

3.1.1 The most popular single Words - Word clouds and bar charts

The two sets of word clouds (see Appendix section, figures 1 and 2) shed light on what words and bigrams occur the most frequently in these four years. According to the first set (i.e. figure 1), politics, especially presidential politics, have been the center of NYT news report regardless of the time. Similarly, finance-related words, such as "income", "bank", "finance" also show up on graphs frequently.

Beyond these central themes, there are also a lot of words that are time-specific. For example, in the word cloud of year 1971, we can see words like "war", "black", "police" that are very contextual - 1971 is the year when US’s involvement in the Vietnam War triggered a series of non-official protests and when the civil right movement hasn’t faded. These words disappeared when we looked at other years, but they provided us important insights into knowing what the media was reporting. Year 1989 has the most financial related words showing up in the top 30 frequent list, and this is interesting and I don't really know why. In year 2010, the impacts caused by 2008 Wall Street Crisis haven't wane so that we are also seeing words such as "bank", "billion" on the word cloud. For year 2020, it's no-surprising that cornonavirus and the presidential election were the single two most important events that captured the media (and the public)'s attentions.

One limit of these word clouds is that we can get limited information with single words and that a lot of words don't make sense. For example, "illus" is probably among the most used words in 1971, however, because the context is lost, it's hard to know what this means. With these limits, I also examined bigrams using a very similar approach.

Lastly, I visualized similar information using bar charts, which are displayed in Appendix, figure 3.

3.1.2 The most popular bigrams - Word clouds and bar charts

Bigrams, i.e. two words, reveal a lot more information than single tokens. Moreover, looking at bigrams also helped us clear questions that originated when looking at single words. For instance, in year 1971, phrases "atty gen" (the acronym for "attorney general"), "grand jury", "communist china" came to our sight. These words are missing when counting by single words. For articles in year 1989, financial words were still a big thing that we saw phrases like "company reports", "net inc" (net income), "share earns" showing up a lot of times, which is consistent with our discovery from single words. For year 2010, now we are seeing important phrases like "health care", "financial crisis", "wall street" on the word cloud. A lot of this, esp "health care", was left out in our inspections of single words. Lastly, we do not retrieve a lot more information from our word cloud of year 2020 bigrams that all the phrases are still around the events of covid and presidential election.

Again, the same information is visualized with bar charts, which can be seen in Appendix, figure 4.

3.1.3 line chart - change across time

The second goal of my EDA is to draw graphs that represent how the occurrence of certain crime/education-related keywords change across these four years. This was realized through the line charts.

I first defined these two sets of keywords. The crime set has the following words: "crime", "police", "violence", "gun", and "arrest", all of which are common in literal English (so no jargon terms). The education set includes these words; "education", "school", "teacher", "parent", and "grade". Associating figures 5 and 6, I found that words "police" and "school" occurred more frequently than other keywords in their respective set, and the frequency of both words exhibited a pattern of steadily decreasing from 1971. The numbers of appearance of other words in the crime set do not differ that much, while for words in the education set, "education" has been the word with secondly high occurrence (just below "school").

It's surprising that "police" on top of other crime-related words, but this conclusion makes logically sense as well as police violence, police brutality have been among issues that concerned Americans the most. Regardless of the time, if police officers are performing their duty in a right manner (i.e. reasonably) have been an important topic. For the other four words in the crime set, it seems that the NYT does not prefer them that much but still, there were an average of 4-500 hundred reports that contained these words in each year. For the education sets, the pattern makes sense if we think from the perspective that "grade", "parent", and "teacher" are the associated words from "school" and "education". The latter two are at a higher level than the former three that we may talk about grade when we talk about school and education and we may not. This explains why "education" and "school" occur far more frequently than the other three.

3.2 Clustering

Clustering is the unsupervised machine learning technique with the purpose of grouping data into a few clusters based on their similarity scores/distance. As I said in the method section, the clustering part of my project can be further divided into two phases - k-means clustering with tf-idf as the way to enumerate data and k-means clustering with Word2Vec to embed words. Both clustering were applied on a sample of the 1971 dataset.

For the clustering with tf-idf, I found out that words in these four clusters actually do not have a central theme. For each cluster, their list of words shares multiple meanings that one usual case is inside each cluster, there are words about politics and words about business. This is also reflected on the scatter plot (see Appendix, figure 7), where the boundary between cluster is vague and there are a lot of outliers. This might suggest that the tf-idf is not a good way to encode our data and something is missing from our modeling.

After changing the word embedding method to Word2Vec with CBOW as the type, we see that the clustering was significantly improved (see Appendix, figure 8). There still have the issue of ambiguous themes, but in a general sense, clusters are more uni-themed than what we got with tf-idf. I noticed that the themes/topics of clusters do not exactly match the themes of the four clusters in the tf-idf graph, and this is as I expected given that two algorithms differ in what they do and the math behind. The first five rows of the K-means processed table is presented in Appendix section, table 3.

Because Word2Vec calculated the cosine similarity of any two words, I am able to find out which words are the most similar to target words. In my analysis, I designed 8 words of interests as my target words and attempted to find the 10 words most similar to each of them. The first four is given below. As we are able to tell, a lot of these similar words actually do not make sense, for example, the algorithm tells me "voters" is related to "possible" and "since", however, such similarity may because of how we write English sentences rather than reasons about public opinion.

Top 10 closed words for word **voters** are: 'possible', 'cities', 'home', 'made', 'soviet', 'also', 'since', 'people', 'times', 'announced'

Top 10 closed words for word **prison** are: 'pres', 'says', 'two', 'club', 'record', 'million', 'time', 'comm', 'would', 'new'

Top 10 closed words for word **criminal** are: 'death', 'san', 'well', 'record', 'coll', 'state', 'exec', 'way', 'money', 'move'

Top 10 closed words for word **student** are: 'area', 'per', 'plans', 'new', 'mrs', 'set', 'nixon', 'israel', 'members', 'left'

3.3 Deep Learning

Lastly, I used deep learning, specially the sequential model in the tensorflow library, to model our data. The goal of my analysis is to find out if one news article is about crime/education from other information. In tensorflow, anywords with index 0 are padded and hereby, ignored in analysis. I also designed the training/testing set ratio to be 0.3, that is, 0.3 of all observations are used as the training set to model the rest 70 percent observations, which constituted the testing set.

There are two important metrics here, accuracy and loss. Accuracy is the percentage of data correctly classified, whereas loss is measured by binary crossentropy, which is a measure of uncertainty in a distribution q(x) and has the form of $H_p(q) = -\sum_{c=1}^{c} q(y_c)\log_2 p(y_c)$.

I found out that with my model, the total loss is 0.0707 and the total accuracy is 0.99. This number is very high, showing that my model is a great fit of the data. Variations of model loss and accuracy over 15 epochs are also visualized (see Appendix, figures 9 and 10).

4 Limits and Future Considerations

My analysis is a successful attempt to explore the data and find out some information but is insufficient or somehow unsuccessful for a computational linguistic project. A lot of drawbacks of my analysis came from my unfamiliarity with natural language processing as this is the first time I worked with NLP. This unfamiliarity comes in a few facets.

First, I did not inspect the data by reading each sentences. It wasn't until I finished the project that I realizes that for each sentence, the last few words indicate the column name or where the article can be found. For instance, a lot of lines end with "Letter to Editors". Ideally, these phrases shall be removed in the preprocessing phase, so that they won't be counted in the frequency analysis.

In a similar fashion, I should also remove all words that have three characters. I initially thought that three-character words are also meaningful, however, I later realized that this is not the case. Three-character words, such as "new", "day", and alike are on top of the frequency tables but they don't tell much information by themselves.

More importantly, a lot more words should be included in the list of stop words and removed from analysis. As we are seeing from the bar charts and word clouds, a lot of words don't make sense. Most of these words are nouns, but some adjectives and verbs are also in this list. As an example, words such as "ger", "final", "son", "day", "term", "list" add little to our semantic and linguistic investigation of NYT news articles.

Moreover, because the 2020 data has far fewer observations (i.e. its dimension is much smaller) than the data of other years, using the dataset may not be ideal. In future analysis, it may be better to use another dataset, for example, the 2000 data that has a similar dimension to keep the statistical validity should we do any cross-dataset analysis.

Last but not least, K-means may not be the best algorithm since outliers are omnipresent in my data. Instead, I should use a clustering algorithm that is less outlier-sensitive. Also, t-SNE may be a better dimensionality reduction technique than PCA since t-SNE regards each cluster as the unit, more tolerant to outliers, and usually fits better. It might worthwhile to try out using SVM with t-SNE to cluster the data and lower the dimension should I decide to redo the analysis.

5 Appreciation

The final project could not be finished without wonderful Metallica songs, which inspired me when I was at my lowest point. This quarter had been so overwhelming with my extremely hard computer science class and an unsmooth internship searching cycle. I was desperate, lost, and broken but heavy metal songs brought me hope, energy, and lights. So many times I wanted to scream out the magic of "I want out" but I know the magic won't happen. Whatever, I want to give my sincere appreciation to metal. Metal will never die!

6 Appendix

Table 1: Head of Cleaned NYT 1971 News Data

sentence	sentence processed	sentence tokenized	sentence bigrammed
A Moursund Jr lr on T Winner's Dec 22 (32:4) l...	moursund winner dec disputes winner suggestion...	[moursund, winner, dec, disputes, winner, sugg...	[moursund winner, winner dec, dec disputes, di...
Amer Gas Assn repts gas utility and pipeline i...	amer gas assn repts gas utility pipeline indus...	[amer, gas, assn, repts, gas, utility, pipelin...	[amer gas, gas assn, assn repts, repts gas, ga...
Dr Spock lr disputes Prof Bickel Dec 21 Op-Ed ...	spock disputes prof bickel dec article attacki...	[spock, disputes, prof, bickel, dec, article, ...	[spock disputes, disputes prof, prof bickel, b...
Northfield Savings Bank begins operation under...	northfield savings bank begins operation new n...	[northfield, savings, bank, begins, operation,...	[northfield savings, savings bank, bank begins...
NYS Health Dept repts 638,077 children, 26.4% ...	nys health dept repts children population aged...	[nys, health, dept, repts, children, populatio...	[nys health, health dept, dept repts, repts ch...

Table 2: Cumulative Tokens and Bigrams Table

Years	Tokens	Num of Tokens	Bigrams	Num of Bigrams
1971	[moursund, winner, dec, disputes, winner, sugg...	2569523	[moursund winner, winner dec, dec disputes, di...	2476133
1989	[lead, editor, bah, humbug, lead, chile, milit...	2862803	[lead editor, editor bah, bah humbug, lead chi...	2758272
2010	[age, digital, imagery, one, man, still, earns...	1983852	[age digital, digital imagery, imagery one, on...	1871404
2020	[president, terrible, toddler, anyone, else, w...	273005	[president terrible, terrible toddler, toddler...	258480

Table 3: Head of Word2Vec-Embedded Clustering Table

Word	x1	X2	Group
rev	1.210986	-0.003230	2
conn	-0.069128	0.001126	0
united	0.864325	0.002869	2
long	0.857688	0.006123	2
away	-0.319121	0.003351	1

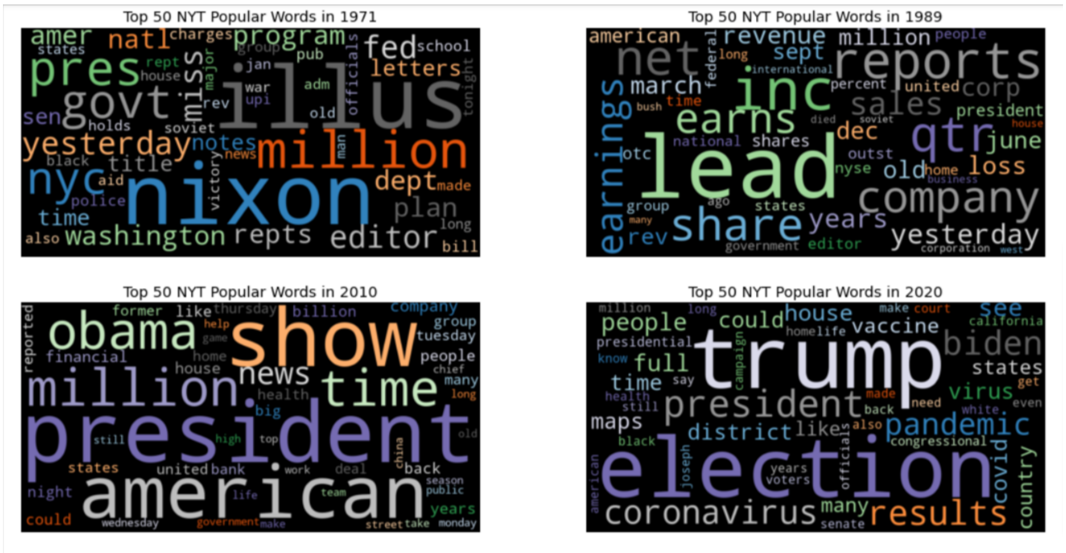

Figure 1: Most Popular Single Words By Year

Figure 2: Most Popular Bigrams By Year

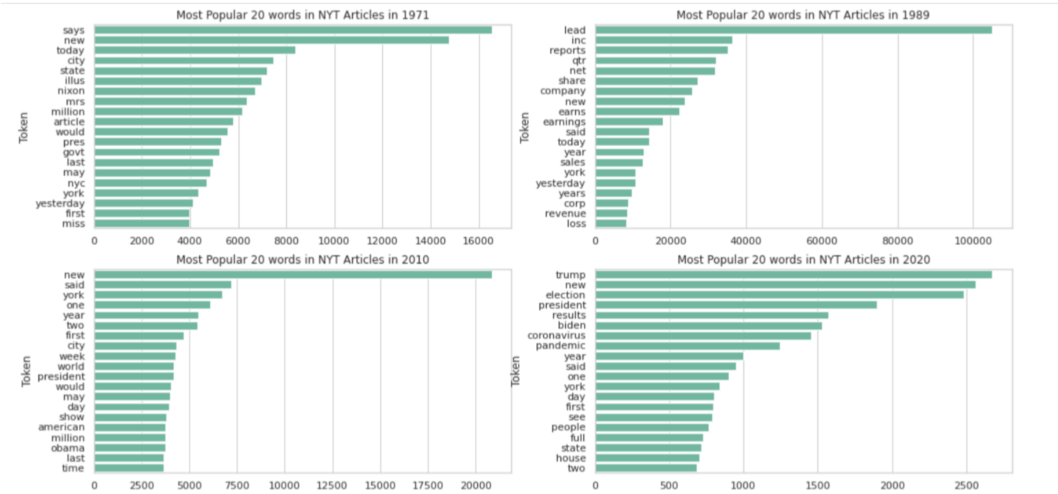

Figure 3: Top-used Single Words in NYT By Year

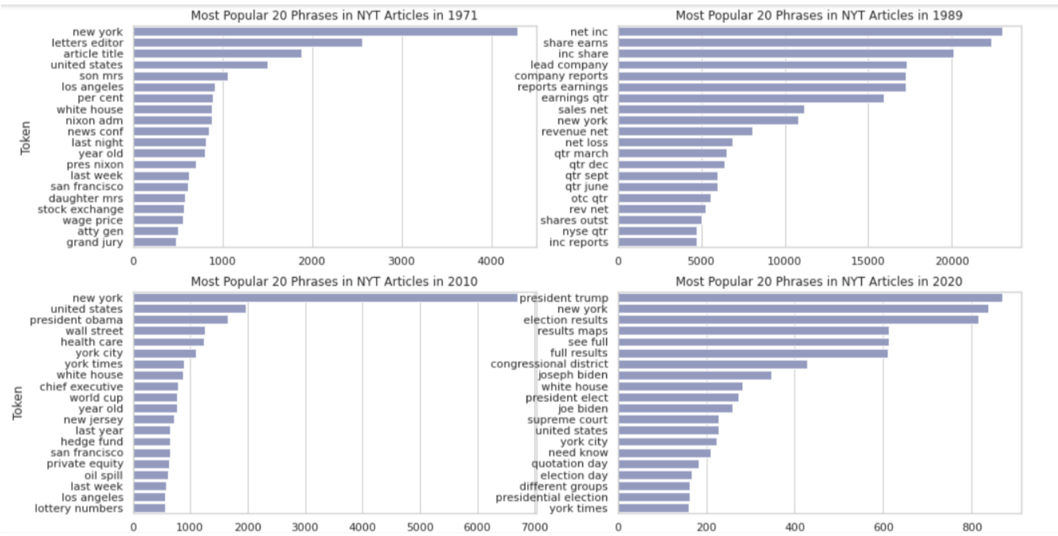

Figure 4: Top-used Bigrams in NYT By Year

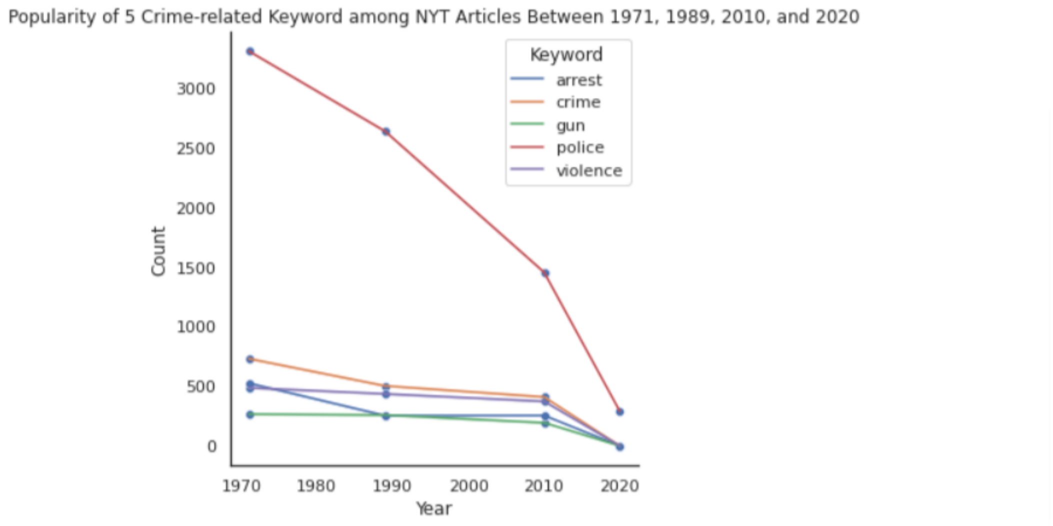

Figure 5: Frequency of Crime Keywords in NYT By Year

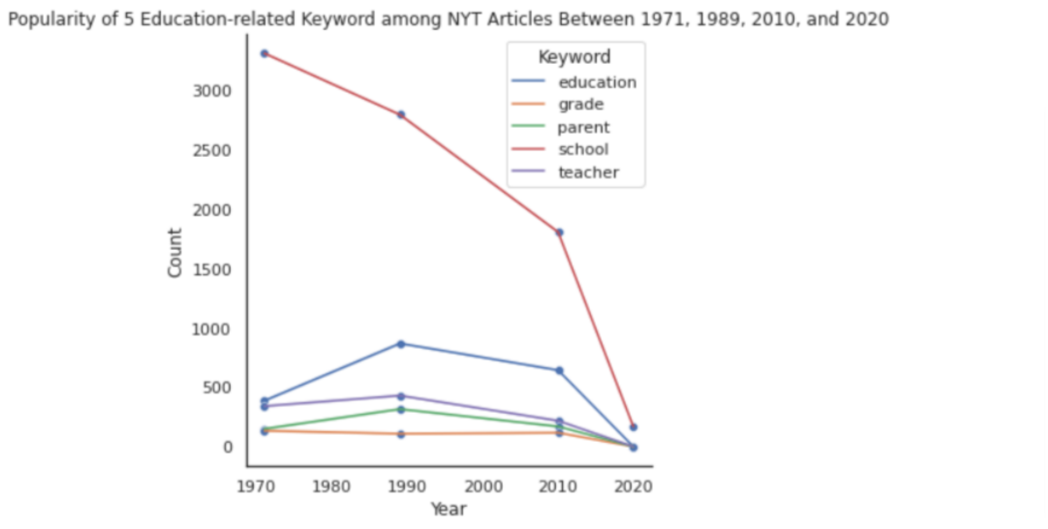

Figure 6: Frequency of Education Keywords in NYT By Year

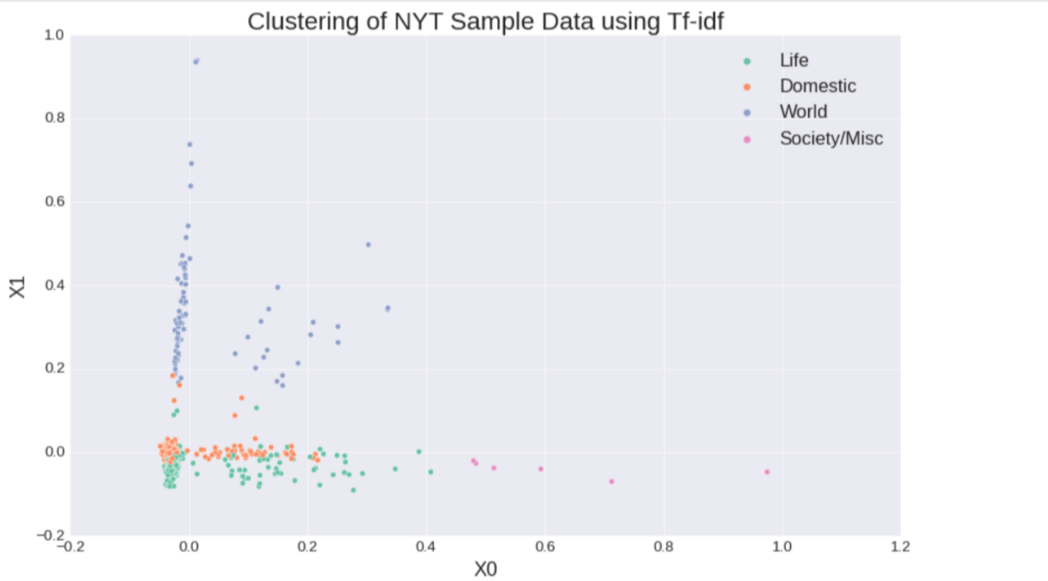

Figure 7: K-means Clustering using Tf-Idf Index

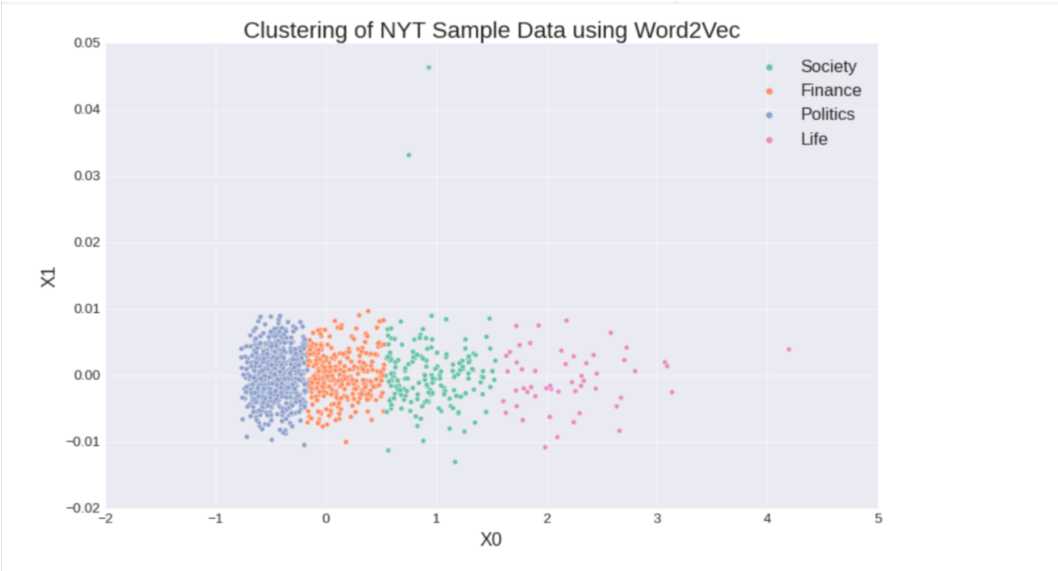

Figure 8: K-means Clustering using Word2Vec

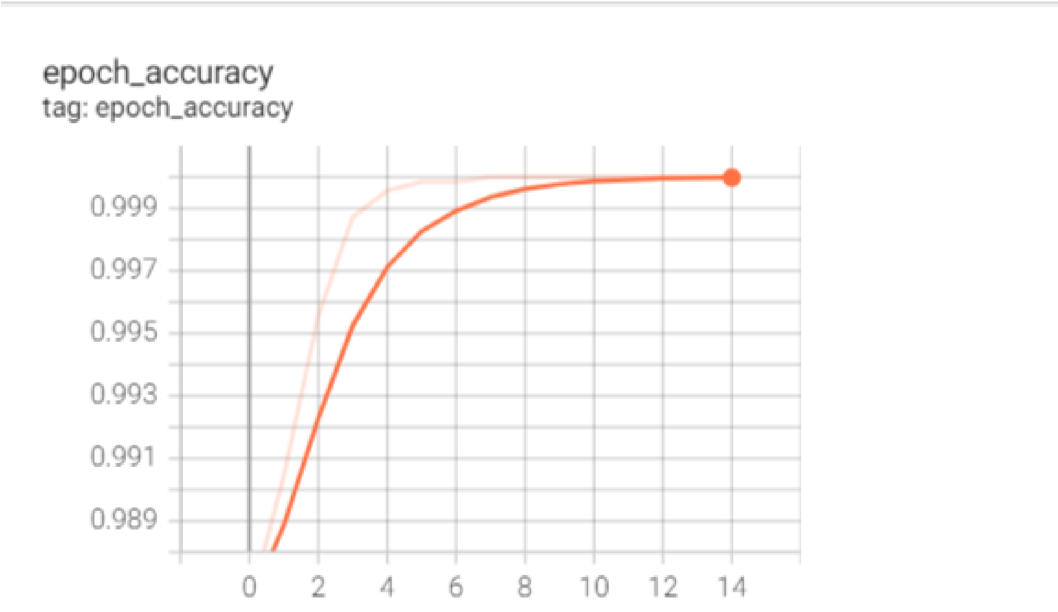

Figure 9: Model Accuracy over Epochs

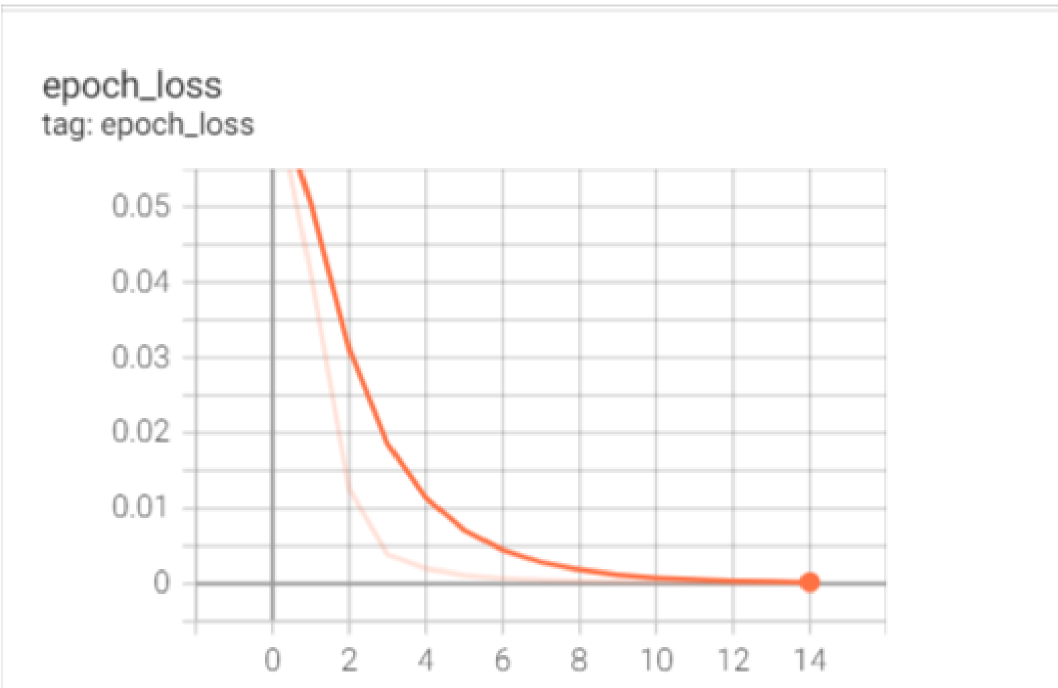

Figure 10: Model Loss over Epoch