Topic Modeling "Gentrification"

Introduction

Studying the term "gentrification" reveals a frustrating curiosity. On one hand, there is an extraordinary range of scholarship pertaining to and contained within gentrification theory. The methods for studying and identifying gentrification are diverse, drawing inspiration from a suite of techniques across many disciplines. No longer contained to London and New York, case studies examine the causes and effects of gentrification on six of the seven continents. It is also a common topic in the popular press, presumably entering the lexicon of most contemporary urban dwellers. On the other hand, gentrification scholars have been unable or unwilling to integrate all of this knowledge. Despite the astonishing breadth of research, or perhaps because of it, gentrification scholars remain comfortably within their siloed debates. A few deft scholars have found success in integrating two or three strands of the literature. Of the broadest attempts to move across conversations, the result resembles little more than a literature review. As illuminating as the extant research is, further inquiry into the economic, social, or legal preconditions of gentrification is not needed. What is needed is an explanation of the concept as a whole, one that makes commensurable these diffuse investigations.

No one has attempted to reconcile the meaning of gentrification using statistics. This is despite the tradition of statistical text analysis dating at least as far back as 1964[1], when Frederick Mosteller and David Wallace used a variety of Bayesian-inspired statistical tools to infer the authorship of *The Federalist* papers, numbers 49–58, 62, and 63[2]. Today, increased computing power and advanced statistical tools are available to streamline a process once doable only by hand. And with text documents increasingly available in digital form, there are ample opportunities to use statistical analysis of text documents to answer a variety of questions across disciplines.

One such form of analysis is topic modeling, a suite of generative models for natural language processing. Topic modeling is an unsupervised machine-learning method to extract emergent themes from a large volume of text. The model views each document in the corpus as consisting of a mixture of topics. Based on correlated frequency distributions of co-occurring words, it populates the topics with recurrent keywords. This could theoretically be done by hand with a small corpus if one were to highlight keywords with different colors based on theme. The document could then be classified as belonging to the dominant color signifying a topic. These topics can be tagged by the user after-the-fact. This method allows the user to effectively map shifting discursive themes in a large corpus over time.

The goal of this paper is to explore a subset of methodological uses for topic modeling in qualitative research. Specifically, two uses are of interest. The first as a means to identify catalysts for thematic shifts in discourse to better understand meaning. The second is to identify case studies for future inquiry. These options will be explored in the context of gentrification as the conceptual object of inquiry. A prototype for the first use will be discussed in depth. Procedures for achieving the second, which has not yet been specified, will also be discussed.

[1] Coincidentally, this is also the year Ruth Glass introduced the term "gentrification".

[2] Their conclusion that James Madison authored the twelve disputed papers remains widely accepted.

Literature Review

Models

Methods for the systematic analysis for text corpora to understand latent meaning have made significant advancements. Perhaps the most well-known and enduring method is content analysis. Content analysis seeks to answer a series of core questions about text documents in a corpus:

> Who
> Says What
> In Which Channel
> To Whom
> With What Effect? (Lasswell, 1948, p. 37).

Bernard Berelson advanced on Lasswell's taxonomy with quantitative flair, contributing a new definition for the emerging method: "a research technique for the objective, systematic and quantitative description of the manifest content of a communicator" (1952, p. 18). Content analysis relies on a researcher coding text documents in the corpus. Coding is a means to simplify the corpus into manageable groups of data where groups are pre-defined to test a hypothesis. Thus, content analysis marked a new step forward for non-invasive social research. However, it remains limited in several ways. Content analysis can be unreliable and difficult to reproduce, relying on a researcher's interpretation of text to code effectively and without error. Coded data may have limited applicability to other questions and may require re-coding for continued use. Most importantly, it is tedious. Manual coding takes an immense amount of human capital which limits realistic corpus size.

Improving on one weakness of content analysis, term frequency is a method for analyzing large volumes of text descriptively. It is a simple yet powerful idea as Hans Peter Luhn suggested: "the probability that the more frequently a notion and combination of notions occur, the more importance the author attaches to them as reflecting the essence of his overall idea" (1957, p. 315). An improved variation of this method is term frequency-inverse document frequency (tf-idf). This downward-adjusts the weights of words that appear frequently across the corpus in favor of words that appear proportionately more in a limited set of documents. That is to say, it adjusts for frequently occurring stopwords in the corpus, like "the" or "is", that may not necessarily be as important as their frequency suggests. While tf-idf can effectively reduce a large corpus to a small list of words and their associated "importance", it lacks any ability to account for context or document structure.

The next iteration of models attempted to solve the issue of context, notably latent semantic indexing (LSI)[3]. As its name suggests, LSI is used to uncover the underlying (latent) semantic structure of words in a corpus. It does this by creating a document-term matrix where rows correspond to terms and columns to documents. This means the algorithm is language independent, viewing words as counts (or weights) rather than as discrete, value-laden entities. The matrix is then compressed to an approximation using singular value decomposition (SVD)

[3] LSI can be used interchangeably with latent semantic analysis (LSA). LSI is older (1990) and is more focused on text search and information retrieval. LSA (2005) is used for natural language processing and speech recognition. Both rely on the same underlying statistical technique, singular value decomposition (SVD).

which assumes that topics are orthogonal[4]. SVD allows the algorithm to handle larger corpora. Finally, relying on the distributional hypothesis—words that are used in similar contexts have similar meanings—allows LSI to capture some degree of polysemy and synonymy, as well as the latent meaning structure (Deerwater, et al., 1990). However, the use of such models for topic analysis is contested given a more direct approach provided by both Bayesian and maximum likelihood methods (Blei, Ng, & Jordan, 2003).

An improvement to LSI is probabilistic latent semantic indexing (pLSI). Where LSI uses linear algebra as its foundation, pLSI uses probability. This allows topics to be non-orthogonal. It "models each word in a document as a sample from a mixture model, where the mixture components are multinomial random variables that can be viewed as representations of 'topics'. Thus each word is generated from a single topic" (Blei, Ng, & Jordan, 2003, p. 994). Each document is made up of multiple topics in differing proportions and thus can be represented by a probability distribution. There are two major drawbacks with pLSI: "(1) the number of parameters in the model grows linearly with the size of the corpus, which leads to serious problems with overfitting, and (2) it is not clear how to assign probability to a document outside of the training set" (ibid.).

Both LSI and pLSI are based on an exchangeability assumption[5]—that word order is irrelevant. Implicit in their formulation is that document order is similarly irrelevant. Drawing on de Finetti's theorem—words are exchangeable, but only independent with regards to an underlying, unobservable family of independent and identically distributed latent variables—latent Dirichlet allocation (LDA) improves on pLSI. It assumes a Dirichlet distribution of documents as random mixtures of topics, and topics as random mixtures of words. Thus, LDA views each document in the corpus as a bag-of-words[6]. Each topic is distributed amongst those document-bags "such that words that are strongly associated with the document's dominant topics have a higher chance of being selected and placed in the document bag. Given the above distributions, the author repeatedly picks a topic, then a word and places them in the bag until a document is complete" (Mohr & Bogdanov, 2013, p. 547). Notably, all documents share the same core group of topics, but topic proportions vary across documents.

A variation on LDA is hierarchical Dirichlet process (HDP). It adds another layer of randomness to the generative model, that of the number of topics in a document. It is useful when the number of topics need be unbounded rather than specified in advance. Another variation is the correlated topic model (CTM). Where the Dirichlet makes a strong independence assumption, the CTM accounts for expected correlations between subsets of latent topics in the corpora. The drawback of CTM is the increased complexity. This is a challenge that faces many "improvements" on LDA. For this reason, LDA remains a popular topic modeling technique to simplify large corpora.

[4] This assumption makes LSI faster but less accurate. The orthogonal basis partially discards data to maximize the amount of represented information. A parallel may be lossy image compression.

[5] This can be a drawback in that recurring multi-word phrases are ignored. A simple fix is to incorporate a multi-gram dictionary. A more complex solution is the incorporation of "turbo topics" (Blei & Lafferty, 2009).

[6] This is another way to conceptualize the exchangeability assumption.

Applications

The structure-preserving simplification and representation of complex data is arguably topic modeling's biggest advantage over other techniques[7]. This presents a question: what is to be done with this now interpretable data? Perhaps the most interesting application—at least in the context of this paper—is the quantification of cultural and institutional meaning. Topic models can build structures of relationships in ways that can be replicated and tested. They can interrogate and understand meaning with the empirical rigor that defines social science (Mohr, 1998). This lends topic models to many unique and temporally-broad questions that draw on a range of textual data.

DiMaggio, Nag, and Blei (2013) apply topic modeling to articles published by five major newspapers between 1986 and 1997 to understand why government grants supporting the arts became controversial during that period. The authors discovered a dramatic shift in the tone of coverage—from celebratory to controversy—occurred in 1989. Controversial frames focused on objectionable grants, mistakes made by the National Endowment for the Arts (NEA), and on the NEA as a catalyst for a larger cultural war. The focus on mistakes made by the NEA corresponded with the Federal budget cycle and the election of George H. W. Bush. The focus on the NEA as a catalyst eventually replaced the focus on objectionable grants as the conversation around national cultural polarization grew.

Similarly, Bonilla and Grimmer (2013) use topic models to identify media frames but only as part of a broader inquiry. Their study examines the effect of the Department of Homeland Security's terror alert system on the public perception of terror. In the first part of their analysis, they show that changing terror alerts ("Yellow" → "Orange") results in a substantial but brief shift in media frames. In the second part, they compare these frameshifts to surveys that were in the field during the period of interest (2002–2005). They conclude that the terror alerts increase the perceived likelihood of a terrorist attack, and they increase economic pessimism. The terror alerts do not, however, shift public policy preferences.

Miller (2013), like Bonilla and Grimmer, also uses state discourse. However, instead of state discourse being used to inform media frames, it is the modeled data in the form of digitized records on crime in 18th–19th century China. One finding, of which there are many, is that the imperial court developed new language to describe later rebellions from earlier rebellions. Later rebellions were described as "major rebellions" and language to describe them referred to militias, planning of events, and the movement of large groups, a deviation from the earlier "rebellions". Miller suggests this is indicative of the government perceiving a "higher threat and changing social circumstances posed by the nineteenth century rebellions" (p. 642).

Turning towards academic corpora, Priva and Austerweil (2015) use topic models to analyze how the cognitive science discipline has evolved by examining titles and abstracts published by *Cognition* between 1980 and 2014. They discovered that the journal has increasingly published experiment-framed articles at the expense of theory-framed articles. This trend began around 1990, though experiment-framing did not overtake theory-framing until 2000. Likewise, Griffiths and Steyvers (2004) also examine research abstracts: in their case from *Proceedings of the National Academy of Sciences* between 1991 and 2001. They use topic modeling to isolate

[7] This could also be a drawback in that it could reduce or reify essential complexities (Denzin, 1991).

uptrending and downtrending topics in the journal. The top three statistically significant uptrending topics were climate change, gene therapy, and apoptosis. The top three statistically significant downtrending topics were sequencing and cloning, structural biology, and immunology. Both trends corresponded to the awarding of Nobel Prizes: apoptosis was the subject of the 2002 Nobel Prize in Physiology, while immunology was the subject in 1989 and sequencing and cloning was the subject in 1993.

Model Selection and Specification

Two separate datasets were used for this project. The first used all 241 "gentrification" tagged articles in my reference management software[8]. This was originally done for convenience and to test the algorithm, but results were meaningful and will be discussed. These files were in PDF format and were converted to individual text files from the terminal (Debian 9.11).

The scope of the second dataset was articles in the popular press[9] containing the word "gentrification". Using API access, I retrieved[10] all articles from Lexis Uni containing the keyword gentrification between 1980 and 2019 (84,616 total). They were downloaded in batches of 50 (Appendix A) and were first tabularized (Appendix B) before being converted to individual text files using Python. After initial tests, this dataset proved too big to process in memory. The text files were arranged chronologically and a systematic sample—every 35th article—reduced the set to a manageable 2,419 files.

Both datasets were loaded into R (separately) and cleaned using the text mining library (Appendix C). All uppercase letters were lowered. Punctuation, white space, numbers, and stopwords were removed. After cleaning, the academic dataset had 1.3 million tokens. The popular press dataset had 1.1 million tokens.

Models were then created for each dataset using the topic models library. For the reasons of simplicity and broad ability, latent Dirichlet allocation is the chosen model for this inquiry. The models are augmented by Gibbs sampling, a Bayesian technique for sampling from a multivariate probability distribution to increase performance in existing algorithms. It is especially useful when information about the joint distribution is limited. In LDA, the conditional probability of topic prevalence given word occurrence, expressed $P(\mathbf{z}|\mathbf{w})$, is intractable, making Gibbs and LDA complementary (Griffiths & Steyvers, 2004).

[8] Retrieved on 2019-11-11.

[9] The print media is viewed as representative of public discourse for several reasons. Foremost, print media is a comprehensive catalog of events. The amount of articles available to parse is large. Further, the media (mostly) accurately and adequately addresses contemporary topics, especially when they are debatable, contestable, or controversial. Gentrification is certainly a topic that has been covered and will continue to be covered by the media with a broad stroke. Following this, the media offers a variety of personal and institutional perspectives, often built around the very words of local actors. This sort of direct and large-scale community interaction is a key component in understanding meaning and perception, especially when the object of inquiry is situated in space. DiMaggio, Nag, and Blei (2013) offer five additional mechanisms through which the media interacts with individual and collective understanding: 1) priming of existing schematic representations, 2) development of new representations, 3) integration with a broader schemata, 4) indirect influence through selective re-telling, and 5) proxy value (pp. 573-574).

[10] Retrieved between 2019-10-04 and 2019-10-17.

This technique, known as the heat bath algorithm in statistical physics, initiates at a random point in the population. It then proceeds to sample randomly but sequentially to create a Markov chain. Because of this random starting point, samples at the beginning of the chain may not be representative and should be thrown out. This is the burn-in period. And because of sample autocorrelation, samples should be thinned to reduce dependency and multicollinearity. This is accomplished by defining thinning and iteration parameters. That is, **i** iterations are performed and every **n**th iteration is kept. A Monte Carlo procedure is added to produce multiple, random runs. All parameters specified in the final models are in Appendix C.

After experimenting with several models, preferred outputs were saved in CSV format and were graphed in LibreOffice Calc to visualize trends.

Model Results

Academic Model

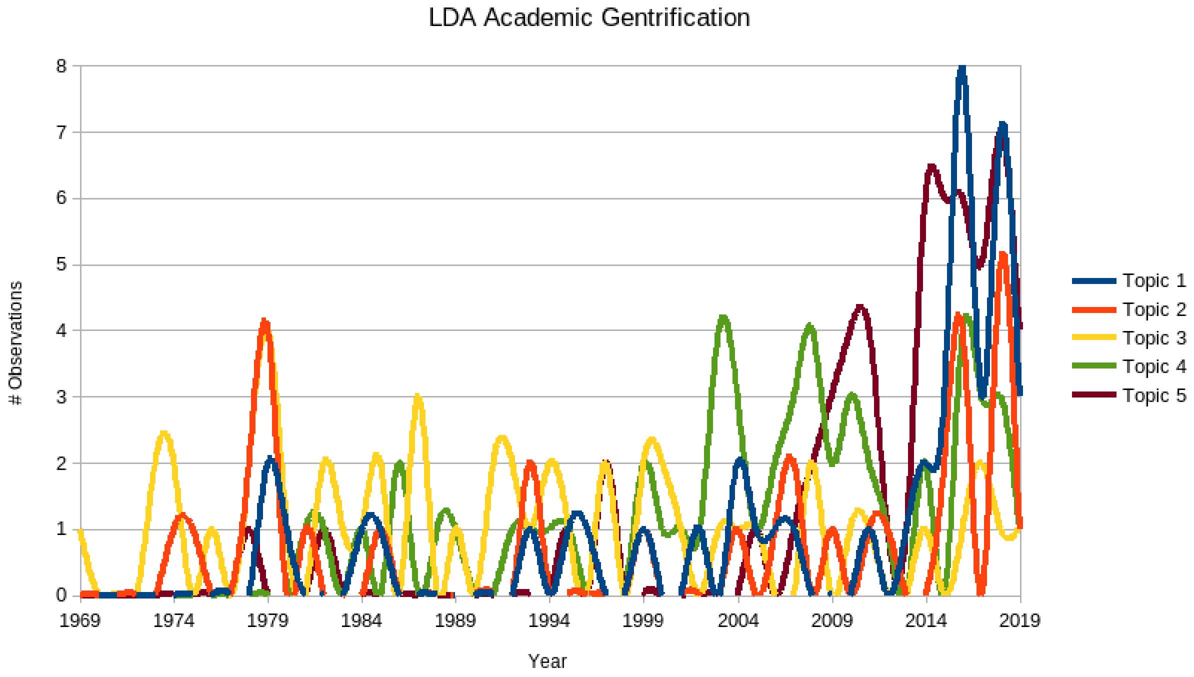
The first thing to notice about the academic model is the keywords associated with each topic (Table 1). For fear of overdetermining meaning, topics were not renamed. However, the topics are diverse and themes are evident. Topic 1 has income and change; Topic 2 has community, development, and displacement; Topic 3 has rent, social, and class; Topic 4 has class and culture; and Topic 5 has social, political, and space. Each of these themes corresponds with frames used to write about and interpret gentrification, as well as key events.

Table 1. Top 5 Topics in Academic Gentrification LDA

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Gentrif	Hous	Rent	Gentrif	Urban
Neighborhood	Citi	Urban	Citi	Social
Citi	Communiti	Social	Urban	Polit
Incom	Develop	Gentrif	Class	Space
Chang	Displac	Class	Cultur	Communti

For example, the spike in 1979 (Figure 1) results from a special issue on neighborhood revitalization published by the Journal of the American Planning Association. Many of the articles in this issue touch on the topic of gentrification and remain heavily cited today. Topic 3 appears to be the core of gentrification research and captures a wide breadth of what we know gentrification to be. Rent is likely to be capturing Neil Smith's rent gap theory (1979), which polarized the field through the late 1990s. There is a general increase in the number of total articles after 2000 which may or may not be precipitating the housing crisis. What is certain is that in the wake of the housing crisis, scholarship on gentrification moved from a theoretical niche to a more mainstream topic.

Figure 1. Academic Gentrification LDA



The strong uptick in Topic 1 may capture general housing market volatility. It may also capture the increasing attempts to use measurement models, largely based on census data, to identify gentrification. The increase in Topic 5 is likely due to the increase in empirical, case-driven, and ethnographic research. These shifts in trends also show how contested the gentrification field remains, especially in terms of how to define, frame, and measure the concept.

Popular Press Model

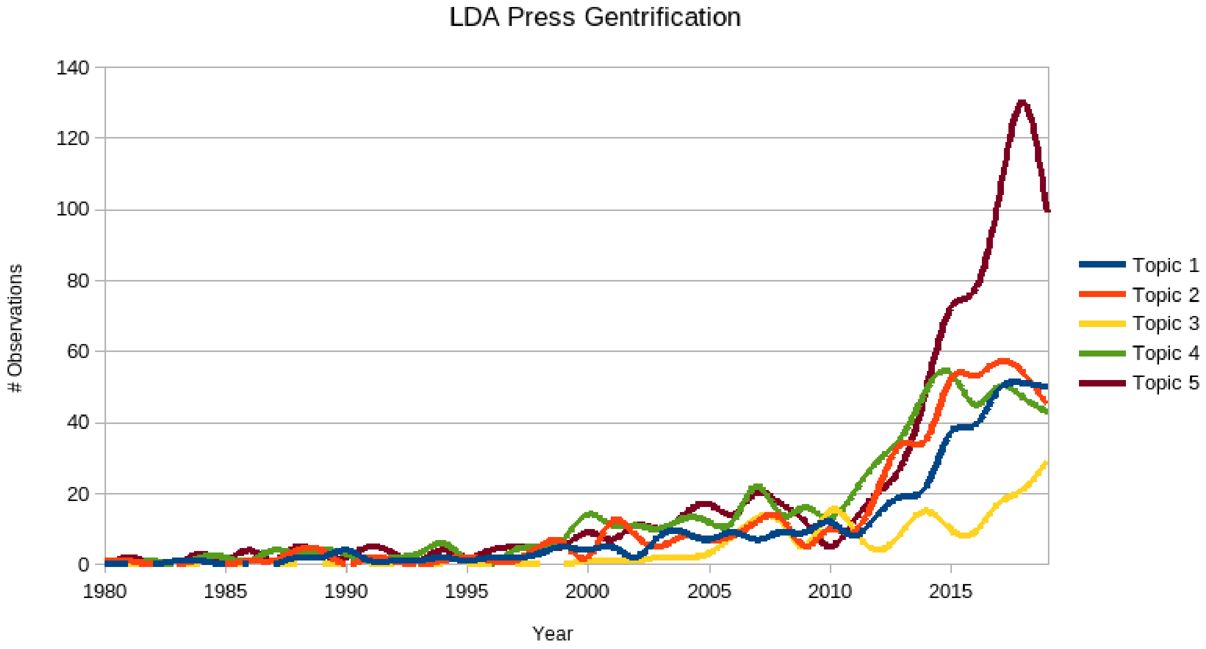
The most interesting aspect of the popular press model is how uninteresting it is. Keywords associated with each topic (Table 2) show some diversity. Topic 1 has art; Topic 2 has black, politics, and police; Topic 3[11] has film; Topic 4 has street and park; and Topic 5 has house, develop, and community. However, when looking at the visual trends (Figure 2), the first thirty years (1980–2010) show almost no discernible differentiation. There is a slow uptrend in the number of articles mentioning gentrification through 2007, with a crash following. One would expect that to be a result of the housing bubble and crisis.

[11] The other keywords appear to be days of the week. This must be corrected for in future models.

Table 2. Top 5 Topics in Popular Press Gentrification LDA

Topic 1	Topic 2	Topic 3	Topic 4	Topic 5
Art	People	Film	Street	Citi
New	Black	Sat	Area	Hous
Street	Said	Une	New	New
Work	Polit	Sun	Park	Develop
Film	Polic	Est	Place	Communiti

Figure 2. Popular Press Gentrification LDA



Then, of course, the housing recovery started and articles on the subject exploded. This mimics the academic trend at least with regards to change in amplitude. However, the spike mainly occurs in Topic 5. This is partially expected. Concepts, like gentrification, are particularly frustrating because they can become ambiguous and lose nuance over time. Here we can see this in full effect. The popular press writes about gentrification primarily through a single frame: gentrification as development.

Discussion

Thematic Shifts

The major goal of this inquiry was to map out shifts in gentrification discourse to better understand the meaning of the ambiguous and often frustrating concept. In that regard, topic modeling was successful, at least as a first step to future work. The models simplify the data in an interpretable way, and they depict the latent structure of both corpora.

Using these models, we can learn much about how gentrification is socially constructed and understood in two very different populations. The academic discourse certainly appears to be trend-driven. Those trends can be traced back to both macro-events (e.g., 2007 housing market collapse) and individual authors (e.g., Neil Smith). And the model articulates the frames through which scholars understand gentrification, as a process of income change, or displacement, or class and social change.

The popular press shows how rapidly the term was introduced into the common lexicon following the 2008 recession. This adoption may also suggest how the term has become so contentious. While it now may be a stand-in for unwanted development, it also contains a rich legacy of racial and class theory. Perhaps some can grasp this theory while others are simply unaware, resulting in the misapplication of the term to any development.

The most stunning result was the chasm in the construction of meaning between the two discourses. While one might expect the academic discourse to lag behind the faster-publishing popular press, this was not the case. It appears as though academic scholars and the popular press are using the same concept to have two very different conversations about cities.

Limitations

There are several limitations to this study that must be acknowledged. Most broadly, there are issues of generalizability and dependency on researcher interpretation. The former is addressed through the large dataset. Even so, the conversations around gentrification are relatively small and defined compared to other urban phenomena, which may be a hindrance. Broader search terms, including "gentrified" and "gentrifying" may help. The latter concern is more challenging to address. The identification of meaningful events and shifts in the discourse, especially in the qualitative narrative, will always be contestable. Careful awareness of my own biases and initial understanding of the literature is key.

Concerning the data itself, other limitations include the availability and quality of the data. As previously mentioned, the Lexis Uni database only has articles going back to 1980 and that is not inclusive of all papers in their database. The web scraper may find articles that are only tangentially linked to gentrification and may end up being superfluous. And of quality, there is a substantial difference between recent articles that Lexis Uni imports directly from media outlets versus the older articles that may be imported through scans or other means. Some articles may have been digitized through Optical Character Recognition (OCR). OCR may be increasingly accurate, but inaccuracies are magnified across the scale of the dataset and could cause the scraper to not identify articles related to gentrification (type 1 error). Finally, the heavy emphasis on the media must acknowledge the fact that the media faces declining readership and influence. Even where influence is still strong, concentration in ownership may confine the diversity of debate on the topic.

Identifying Cases

The second goal of this paper was to see if topic modeling could serve as a means to identify cases for future study. The given time constraints prevented exploration and validation. However, the idea is no less valid and should still be explained.

These topic models capture a surprising amount of temporal variation. However, gentrification is both temporal and spatial. One needs a way to capture the spatial variation to better understand how gentrification in a place like New York or London compares to gentrification in Detroit, Hong Kong, San Francisco, and almost every other city. Given the scope of both academic and popular press articles on gentrification, it should be viable to split the datasets into spatially-demarcated groups. Topic modeling on these subsets may yield three results: no change, temporal differences, or topic differences. The latter two are not mutually exclusive. Differences that emerge in the latter categories may suggest cities and periods for future case-study research.

Concluding Remarks

This paper used topic models to explore how "gentrification" is understood in academic articles and popular press articles. Topic modeling is a powerful tool for grappling with large datasets and a valuable complement to interpretative research. However, it must be used carefully and conscientiously, and researchers must take note that it is only a first step of inquiry.

This analysis raised valuable qualitative questions that should be explored in future research. Why did gentrification writing, in both discourses, grow so rapidly after the 2008 housing crash and subsequent recovery? Why isn't the same uptick evident in the bubble leading up to the crash? Is this rise in attention tied to larger cultural shifts (e.g., "wokeness"), or is it a result of the massive dispossession and urban restructuring that occurred as mortgages failed?

The analysis also raised technical questions. Would a correlated topic model be more effective given the expected collinearity between gentrification frames? Are other packages (e.g., gensim) more capable of handling the large corpora and with more control? And can manual subsampling adequately capture spatial variation in topic modeling?

Works Cited

Berelson, B. (1952). *Content Analysis in Communication Research*. Glencoe: Free Press.

Blei, D. M., & Lafferty, J. D. (2007). A Correlated Topic Model of Science. *The Annals of Applied Statistics, 1*(1), 17–35.

Blei, D. M., & Lafferty, J. D. (2009). Visualizing Topics with Multi-Word Expressions. Retrieved from https://arxiv.org/abs/0907.1013

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research, 3*, 993–1022.

Bonilla, T., & Grimmer, J. (2013). Elevated threat levels and decreased expectations: How democracy handles terrorist threats. *Poetics, 41*, 650–669.

Deerwater, S., Dumais, S., Landauer, T., Furnas, G., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science, 41*(6), 391–407.

Denzin, N. K. (1991). Empiricist cultural studies in America: a deconstructive reading. *Current Perspectives in Social Theory, 11*, 17–39.

DiMaggio, P., Nag, M., & Blei, D. (2013). Exploiting affinities between topic modeling and the sociological perspectives on culture: Application to newspaper coverage of U.S. government arts funding. *Poetics, 41*, 570–606.

Glass, R. (1964). Introduction. In R. Glass, E. J. Hobsbawm, H. Pollins, W. Ashworth, J. H. Westergaard, W. Holford, M. Jeffreys, J. Jackson, & S. Patterson (Eds.), *London: Aspects of Change* (pp. xiii–xlii). London: MacGibbon & Kee.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences, 101*(suppl. 1), 5228–35.

Jockers, M. L., & Mimno, D. (2013). Significant themes in 19th-century literature. *Poetics, 41*, 750–769.

Lasswell, H. (1948). The Structure and Function of Communication in Society. In *The Communication of Ideas (Ed. B. Lyman).* New York: Institute for Religious and Social Studies, 37–51.

Luhn, H. P. (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. *IBM Journal of Research and Development, 1*(4), 309–17.

Miller, I. M. (2013). Rebellion, crime and violence in Qing China, 1722–1911: A topic modeling approach. *Poetics, 41*, 626–649.

Mohr, J. W. (1998). Measuring Meaning Structures. *Annual Review of Sociology, 24*, 345–70.

Mohr, J. W., & Bogdanov, P. (2013). Introduction—Topic models: What they are and why they matter. *Poetics, 41*, 545–69.

Mosteller, F., & Wallace, D. L. (1964). *Inference and Disputed Authorship: The Federalist*. Reading: Addison-Wesley.

Priva, U. C., & Austerweil, J. L. (2015). Analyzing the history of *Cognition* using Topic Models. *Cognition, 135*, 4–9.

Smith, N. (1979). Towards a theory of gentrification: A back to the city movement by capital, not people. *Journal of the American Planning Association, 45*(4), 538–48.

Appendix A

```
# This will loop through all results in a query, 50 at a time, saving individual JSON files to the same directory as the script

import requests
from requests.auth import HTTPBasicAuth
from datetime import datetime  # used to name json files
from time import sleep
import json

query = "gentrification" #  Place entire query string inside these quotes. (e.g. "'Edwin Willers'")
filter = "" #  Place entire query string inside these quotes. (e.g. "SearchType eq LexisNexis.ServicesApi.SearchType'Boolean' and PublicationType eq 'TmV3c3BhcGVycw' and GroupDuplicates eq LexisNexis.ServicesApi.GroupDuplicates'ModerateSimilarity' and Language eq LexisNexis.ServicesApi.Language'English'")

client_id = # Replace with real Client ID
secret =   #Replace replace with real Secret

############# Begin Function Definitions #############

def get_token(client_id, secret):
    """Gets Authorizaton token to use in other requests."""
    auth_url = 'https://auth-api.lexisnexis.com/oauth/v2/token'
    payload = ('grant_type=client_credentials&scope=http%3a%2f%2f'
                'oauth.lexisnexis.com%2fall')
    headers = {'Content-Type': 'application/x-www-form-urlencoded'}
    r = requests.post(
            auth_url,
            auth=HTTPBasicAuth(client_id, secret),
            headers=headers,
            data=payload)
    json_data = r.json()
    return json_data['access_token']


def build_url(content='News', query=", skip=0, expand='Document', top=50, filter=None):
    """Builds the URL part of the request to Web Services API."""
    if filter != None:  # Filter is an optional parameter
        api_url = ('https://services-api.lexisnexis.com/v1/' + content +
                    '?$expand=' + expand + '&$search=' + query +
                    '&$skip=' + str(skip) + '&$top=' + str(top) +
                    '&$filter=' + filter)
    else:
        api_url = ('https://services-api.lexisnexis.com/v1/' + content +
                    '?$expand=' + expand + '&$search=' + query +
                    '&$skip=' + str(skip) + '&$top=' + str(top))
    return api_url


def build_header(token):
    """Builds the headers part of the request to Web Services API."""
    headers = {'Accept': 'application/json;odata.metadata=minimal',
                'Connection': 'Keep-Alive',
            'Host': 'services-api.lexisnexis.com'}
    headers['Authorization'] = 'Bearer ' + token
    return headers


def get_result_count(json_data):
    """Gets the number of results from @odata.count in the response"""
    return json_data['@odata.count']


def time_now():
    """Gets current time to the second."""
    now = datetime.now()
    return now.strftime('%Y-%m-%d-%H%M%S')

############# End Function Defnitions #############

############# Begin business logic #############

token = get_token(client_id, secret)  # 1 token will work for multiple requests
request_headers = build_header(token)
skip_value = 0  # Sets starting skip
top = 50  # Adjusts the number of results to return

while True:
    request_url = build_url(content='News', query=query, skip=skip_value, expand='Document', top=top,
filter=None)  # Filter is set to filter=None here. Change to filter=filter to use the filter specified above
    r = requests.get(request_url, headers=request_headers)

    with open(str(time_now()) + '.json', 'w') as f_out:  # Creates a file with the current time as the file name.
        f_out.write(r.text)

    skip_value = (skip_value + top)
    json_data = r.json()
    if skip_value > get_result_count(json_data):  # Check to see if all the results have been looped through
        break

    sleep(20)  # Limit 5 requests per minute (every 60 seconds) was originally 12
```

Appendix B

```
import sys
import logging
import json
import os
import glob
import time
import bs4 as bs
import pandas as pd

logging.basicConfig(
    filename="logfile.log",
    level=logging.INFO,
    format="%(asctime)s.%(msecs)03d %(levelname)s %(module)s - %(funcName)s: %(message)s",
    datefmt="%Y-%m-%d %H:%M:%S",
)
# logging.getLogger().addHandler(logging.StreamHandler())

def error_handling():
    """Function for checking error and error line"""
    return f"Error: {sys.exc_info()[0]}. {sys.exc_info()[1]}, line: {sys.exc_info()[2].tb_lineno}"

def get_all_json():
    """Gets the list of the JSON file names inside the current path"""

    # get current path
    path = os.path.dirname(os.path.abspath(__file__))

    # get all the files inside the current path
    onlyfiles = [f for f in os.listdir(path) if os.path.isfile(os.path.join(path, f))]

    json_files = []

    # filter only json files
    for file_name in onlyfiles:
        if file_name.endswith(".json"):
            json_files.append(file_name)

    return json_files

def scrape_data_from_article(article):
    """Takes an artile as a dict and returns a dict containing the desired fields"""

    try:
        soup = bs.BeautifulSoup(article["Document"]["Content"], "lxml")

    except Exception:
        logging.error(error_handling())

        return {
            "date_published": "",
            "date_updated": "",
            "document_id": "",
            "title": "",
            "article_text": "",
        }

    try:
        date_published = soup.find("published").text
    except Exception as e:
        logging.error(error_handling())
        date_published = ""

    try:
        date_updated = soup.find("updated").text
    except Exception as e:
        logging.error(error_handling())
        date_updated = ""

    try:
        document_id = article["ResultId"]
        if "urn:contentItem:" in document_id:
            document_id = document_id[16:]
    except Exception as e:
        logging.error(error_handling())
        document_id = ""

    try:
        title = article["Title"]
    except Exception as e:
        logging.error(error_handling())
        title = ""

    try:
        article_text = soup.find("bodytext").text
    except Exception as e:
        logging.error(error_handling())
        article_text = ""

    return {
        "date_published": date_published,
        "date_updated": date_updated,
        "document_id": document_id,
        "title": title,
        "article_text": article_text,
    }

def main():
    # file_name = sys.argv[1]
    # print(file_name)

    json_files = get_all_json()
    total_files = len(json_files)
    # print(json_files)

    output_columns = [
        "date_published",
        "date_updated",
        "document_id",
        "title",
        "article_text",
    ]

    df = pd.DataFrame(columns=output_columns)

    for index, file_name in enumerate(json_files):
        print(f"Processing {index+1} of {total_files} files")

        with open(file_name) as json_file:
            data = json.load(json_file)

        for article in data["value"]:
            row = scrape_data_from_article(article)
            df = df.append(row, ignore_index=True)

    time_now = str(time.time())
    time_now = time_now[: time_now.find(".")]

    # dot_on_file_name = file_name.find(".")
    # file_name_init = file_name[:dot_on_file_name]

    output_file_init = "_".join(["output", time_now])
    output_file_name = ".".join([output_file_init, "csv"])

    if not os.path.exists("output_folder"):
        os.makedirs("output_folder")

    output_location = "/".join(["output_folder", output_file_name])
    df.to_csv(output_location, index=False)

    print(f"Output file saved as {output_location}")

if __name__ == "__main__":
    main()
```

Appendix C

```
library(tm) #load text mining library

#load files into corpus
setwd()
filenames ← list.files(getwd(), pattern=”*.txt”)
files ← lapply(filenames, readLines) #read files into a vector
docs ← Corpus(VectorSource(files)) #create corpus from vector

#process data, remove unwanted characters
docs ← tm_map(docs, content_transformer(tolower))
docs ← tm_map(docs, removePunctuation)
docs ← tm_map(docs, removeNumbers)
docs ← tm_map(docs, removeWords, stopwords(“english”))
docs ← tm_map(docs, stripWhitespace)
docs ← tm_map(docs, stemDocument)

dtm ← DocumentTermMatrix(docs)
rownames(dtm) ← filenames
freq ← colSums(as.matrix(dtm)) #collapse matrix
length(freq)
ord ← order(freq, decreasing=TRUE)
write.csv(freq[ord], “word_freq.csv”)

library(topicmodels) #load topic models library

#set Gibbs parameters
burnin ← 4000 #discarded iterations
iter ← 2000 #see thin
thin ← 500 #every 500th iteration is returned for 2000 iterations
nstart ← 5 #number of repeated runs

k ← 5 #number of topics

ldaOut ← LDA(dtm, k, method=”Gibbs”, control=list(nstart=nstart, burnin=burnin, iter=iter, thin=thin))

#write results
ldaOut.topics ← as.matrix(topics(ldaOut))
write.csv(ldaOut.topics, file=paste(“LDAGibbs”, k, “DocsToTopics.csv”))

ldaOut.terms ← as.matrix(terms, ldaOut, 5) #pick top 5 words in each topic
write.csv(ldaOut.terms, file=paste(“LDAGibbs”, k, “TopicsToTerms.csv”))

topicProbabilities ← as.data.frame(ldaOut@gamma)
write.csv(topicProbabilities, file=paste(“LDAGibbs”, k, “TopicProbs.csv”))
```