Searching for Contrast in the Beige Books: Applying Transformers to Predict Future Regional Unemployment Rates

Stanford CS224N Custom Project

Douglas Callahan
SCPD
Stanford University
decal@stanford.edu

Abstract

Real-time economic statistics are generally unavailable: useful economic statistics often take months to compile and publish. Real-time *regional* economic statistics are even more of a rarity. The Beige Book presents a unique opportunity to fill this gap. Published 8 times a year by the U.S. Federal Reserve, the Beige Book provides real-time analyses of economic conditions in 12 different regions of the United States. This project uses transformer models to map those Beige Book reports to changes in the unemployment rate, using both out-of-the-box transformer models as well as fine-tuned transformers. Although these NLP methods ultimately add little in predictive accuracy to the baseline—a conventional time-series analysis—this paper concludes with several promising future research paths.

1 Key Information to include

- Mentor: Lucia Zheng
- External Collaborators (if you have any): None
- Sharing project: No

2 Introduction

As I type this manuscript in the middle of March 2022, regional unemployment rates for January 2022 have still not been released by the U.S. Bureau of Economic Statistics: the latest available rate information is for December 2021. This lag of several months—occurring between the economic reality of regional unemployment and its eventual tabulation and publication by institutional economists—is understandable but lamentable. It is understandable because the collection of regional unemployment information is a difficult and time-consuming econometric problem; and this lag is lamentable because it deprives economic actors of real-time information to guide their business and investment decisions.

The so-called "Beige Books" of the Federal Reserve banks offer an opportunity to substantially shrink this lag. The U.S. Federal Reserve includes 12 regional district banks; and eight times per year, each district bank issues a report on regional economic status. The compilation of these reports goes by the name of the "Beige Book." These Beige Books are based on interviews with diverse economic actors in the respective regions and on the latter's insights into and opinions of the regional economy.

Below is a representative excerpt from the St. Louis district's April 17, 2019 report:

> **Employment and Wages** - Employment has increased slightly since the previous report. Contacts throughout the District continued to note a tight labor market in a variety of industries, including construction, health care, manufacturing, and information technology. Companies have used a myriad of strategies to attract and retain workers, such as signing bonuses and paid time off. A contact in the trucking industry reported that insurance policies have prevented firms from hiring less-experienced drivers. Contacts in the agriculture industry near Memphis reported filling vacancies with temporary workers through the H-2A visa program.

Based as they are on interviews, the Beige Books generally provide qualitative, not quantitative, information about the regional economies, expressed in English natural language. In this regard, they lack the precision and quantitative measures associated with conventional economic statistics. The Beige Books thus offer an exchange: if economic actors are willing to accept non-quantitative economic indicators, they can access those indicators in real-time.

NLP and ML methods can potentially sweeten this exchange in three different ways: (1) distilling the text of the Beige Books into compact and actionable indicators, (2) estimating the uncertainty around these indicators, and (3) automating this distillation and encoding process.

To this end, this project processes the Beige Books, maps them to categorical indicators of changes in the regional unemployment rate ("decrease," "increase"), and then trains several models to "predict" current changes in regional unemployment rates for which authoritative statistics are not yet available.

3 Related Work

Other researchers have attempted to use the pronouncements of the Federal Reserve to predict the movement of interest rates (over which the Federal Reserve exercises a great deal of control). [1] [2] At least one student has attempted to predict movements in GDP using off-the-shelf sentiment-analysis models applied to the Beige Books. [3] And one graduate student has created finBERt—a BERT variant trained on a financial media corpus and fine-tuned for financial sentiment analysis. [4]

But none have so far attempted to use BERT and the Beige Books to predict movements in economic indices—especially indices as specific as the regional unemployment rate.

In my project milestone, I acknowledged the fact that I borrowed heavily from the data-scraping and -cleaning code available on Oscar Suen's github. Since the project milestone, all coding work has been my own.

4 Approach

To repeat: as part of exploring the feasibility of using the Beige Books to predict the movement upward or downward of regional economic indicators, this project attacks the specific problem of predicting the upward/downward movement of U.S. regional unemployment rates.

Among other available economic indicators, the unemployment rate recommends itself (1) because of its obvious importance to economic policy and business decisions and (2) because *qua* percentage it controls for population growth (unlike, for example, GDP).

4.1 Baseline Model

As a baseline against which to evaluate performance, this project chose a "vanilla" time-series analysis of past changes in the regional unemployment rate. For any given month from 1976 through 2021, this time series included the following data fields:

- the past 3 month-over-month changes in the regional unemployment rate,
- the previous month's unemployment rate,

- the year, and
- indicator variables representing the U.S. region.

These variables were then run through three different classifier "heads": Random Forests, Gradient Boosted Trees, and a feed-forward neural net. As discussed in Section 5, this fairly meager set of data fields proved to be a surprisingly powerful predictor.

4.2 Supplemental NLP Data

The vanilla time series was supplemented with additional NLP information of three different sorts:

- the final hidden state of a transformer model (distilBERT, BERT, or finBERT) produced after running the relevant district bank's Beige Book analysis through the transformer;
- the first 30 principal components of the aforementioned hidden state; and
- the output of a transformer model (distilBERT or BERT) that had been fine-tuned on the problem of predicting the upward/downward movement of regional unemployment rates based solely on the district bank's Beige Book analysis for that month.

The supplemented series were then again run through the same three classifier heads (Random Forests, Gradient Boosted Trees, and a feed-forward neural net), in order to see whether they outperformed the baseline.

5 Experiments

5.1 Data

5.1.1 Data Collection and Processing

The specific response variable for this classification problem is whether the month-over-month change in regional unemployment rate is negative or positive—i.e., did a given month's unemployment rate represent a decrease or increase with respect to the previous month's. The former were coded as 0, the latter as 1.

To calculate historical regional unemployment rates for each district, I collected state-level unemployment rates for every month since 1976 (available through the St. Louis FRED API). I then aggregated the state rates into regional rates through a weighted-average: every state unemployment rate in a given region was weighted by the state population at that time (time series of state populations are also available via the FRED API).

Historical regional unemployment rates and their month-over-month changes are represented in Figures 3 and 4 in the Appendix. (Please note that, in order to preserve a reasonable scale in the graph, I had to truncate the time series of month-over-month changes at March 2020. The pandemic-related jump in unemployment that occurred in March 2020 was the largest in the last 80 years by at least a factor of 5.)

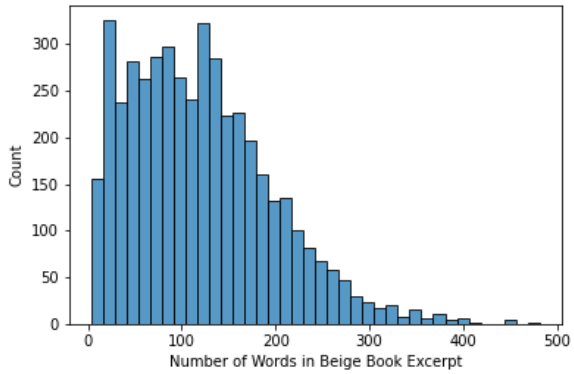
For the Beige Book text, I scraped the Minneapolis Federal Reserve bank website and downloaded every Beige Book statement made since 1976, organizing the statements by year, month, and Federal Reserve district a/k/a region. I then preprocessed these statements by, among other things, extracting only those sentences that contained some word lemma from this list: ['employ', 'labor', 'work', 'job', 'occupation', 'hire', 'hiring']. This process could produce a substantial collection of sentences: for a single date-region combination, the collection of sentences could approach 500 words total.

5.1.2 Data Challenges

In general, the dataset reaches back to 1976 and extends forward to December 2021. After dropping empty rows, this amounts to 4548 total datapoints.

Unfortunately, the dataset is imbalanced, and this represents one of its major challenges:

Figure 1: Number of Words in Beige Book Excerpts



there are nearly twice as many decreases as there are increases. Relatedly, these upward and downward movements are asymmetrical, as the chart below shows:

Table 1: Basic Statistics for Monthly Change in Unemployment Rate

	Count	Mean	Std Dev.	Minimum	Median	Maximum
Increases in Unemp. Rate	1581	0.17712	0.91750	0	0.06178	14.22487
Decreases in Unemp. Rate	2967	-0.08865	0.18284	-3.56008	-0.06114	0

The unemployment rate moves violently upward when catastrophic events occur: the 2009 financial crisis and the 2020 pandemic are two excellent examples when enormous numbers of people were thrown out of work in a matter of months or a single month, respectively. By contrast, when the unemployment rate moves downward, it tends to do so slowly, as employers gradually react to changing economic circumstances.

As will be seen in the analysis that follows, this imbalance and asymmetry will create significant modeling and fitting problems.

Table 6 in the Appendix displays the correlations among regional unemployment rate changes. There seem to be at least four groups that move together in their unemployment rates: the East Coast (Boston, New York, Philadelphia), the West Coast (San Francisco), the Midwest and Midatlantic (St. Louis, Cleveland, Richmond, Atlanta, Chicago, Minneapolis, Kansas City), and the Sunbelt (Dallas).

5.2 Evaluation method

I have predominantly defined success in this modeling exercise by the metric of accuracy—i.e., number of successfully classified observations over the total number of observations. In recognition of the imbalanced nature of the dataset, I have also included the F1 metric. A "dumb" classifier could achieve an accuracy of $\frac{2967}{4548} \approx 65.23\%$ simply by classifying every observation as a decrease. By combining both precision and recall, the F1 statistic provides better context on the classifier's performance in an imbalanced environment.

Beyond correctly predicting increases or decreases in the unemployment rates, it would also be very helpful if the model could offer some measure of the certainty attending its predictions. Cross-entropy—a classification metric sensitive to predicted probabilities—would therefore also be very helpful in measuring this model's performance relative to the baseline. Unfortunately, I lacked the time to identify a cross-entropy metric for the tree-based classifiers I used here. Accordingly, cross-entropy results are only available where I used neural nets.

Although a 20% hold-out test set is conventional, in my testing I universally used a 15% hold-out set, owing to the generally small size of the dataset.

5.3 Experimental details

When I supplemented the time series with the 768-dimensional vector representing the final hidden state of each transformer models, the results were underwhelming:

Table 2: Performance of Models Supplemented by Transformer Final Hidden State

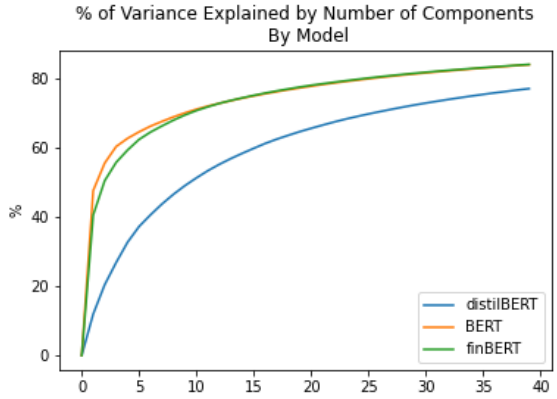
NLP Model	NLP Feature Set	Classifier Type	Accuracy	F1
None	N/A	Random Forests	0.82650	0.73109
distilBERT	Full hidden state	Random Forests	0.82606	0.71701
BERT	Full hidden state	Random Forests	0.82723	0.70385
finBERT	Full hidden state	Random Forests	0.82079	0.70053
None	N/A	Boosted Trees	0.84451	0.75702
distilBERT	Full hidden state	Boosted Trees	0.84480	0.76193
BERT	Full hidden state	Boosted Trees	0.84129	0.75153
finBERT	Full hidden state	Boosted Trees	0.84714	0.75918

As can be seen above, none of the supplemented models significantly outperformed the baseline. This naturally led to the suspicion that supplementing the data with a 768-dimensional vector was causing overfitting—a suspicion heightened by the relatively small size of the dataset. In order to reduce the dimensionality of the NLP content, I attempted two techniques: principal components analysis (PCA) and extracting the output (rather than the final hidden state) from the transformer models.

5.3.1 Dimension Reduction Through Principal Components Analysis

As a hyperparameter tuning exercise, I inspected how the sum of explained variance scaled with the number of PCA components for each transformer:

Figure 2: Determining the Number of Principal Components



About 80% of the variation in the 768-dimension hidden state is captured by the first 30 PCA components of BERT and finBERT. The distilBERT graph is strictly dominated by the other two, however, indicating that a 30-dimensional embedding of distilBERT's final hidden state occupies a higher-dimensional linear space and thus is less susceptible to PCA. This is exactly *the opposite* of what you would expect from a model with fewer parameters—perhaps a subject for further investigation.

5.3.2 Fine-tuning

As another means of reducing dimensionality, I fine-tuned distilBERT and BERT on the classification task using only the Beige Book text as input (the pretrained finBERT model was only capable of 3-class prediction and therefore not capable of being easily fine-tuned).

Perhaps owing to the small size of the dataset, significant overfitting began after 2 epochs of training (as evidenced by starkly diverging training and test losses). Below are the results of fine-tuning for 2 epochs:

Table 3: Performance of Different Models on Test Set After 2 Epochs of Training

Model	Loss Metric	Training Loss	Test Loss	Accuracy	F1
distilBERT	Cross-entropy	0.551500	0.651794	0.714495	0.696344
BERT	Cross-entropy	0.559200	0.617292	0.679356	0.659682
finBERT	N/A	N/A	N/A	N/A	N/A

Table 4: Confusion Matrices for Fine-Tuned Models

distilBERT			BERT			finBERT		
	Decr.	Incr.		Decr.	Incr.		Decr.	Incr.
Decr.	395	58	Decr.	382	71	Decr.	N/A	N/A
Incr.	137	93	Incr.	148	82	Incr.	N/A	N/A

The results were less than encouraging. A "dumb" classifier can achieve 65% accuracy, and the best fine-tuned model here was only able to achieve 71.4% accuracy (Table 3).

As Table 4 shows, the fine-tuned models tended to disproportionately classify points as "decreases". The distilBERT model, for example, classified $\frac{395+137}{683} \approx 78\%$ points as decreases, even though decreases only represent $\frac{395+58}{683} \approx 66\%$ of the dataset.

I tuned the training process by halting training after 2 epochs and lowering the learning rate to 2×10^{-6}. Some additional tweaking of the training hyperparameters very well could have resulted in better performance.

5.4 Results

5.4.1 Relative Classifier Performance

Of the three classifier "heads" used in these models, Gradient Boosted Trees proved to be the most consistently high-performing, followed by Random Forests and then by neural nets. The neural-net classifier never achieved an accuracy higher than 81% during testing and is for this reason omitted from the tables below.

Gradient Boosted Trees and Random Forests are generally known as two of the best-performing "out-of-the-box" classifier methods because they are particularly robust against overfitting and imbalanced datasets like this one.

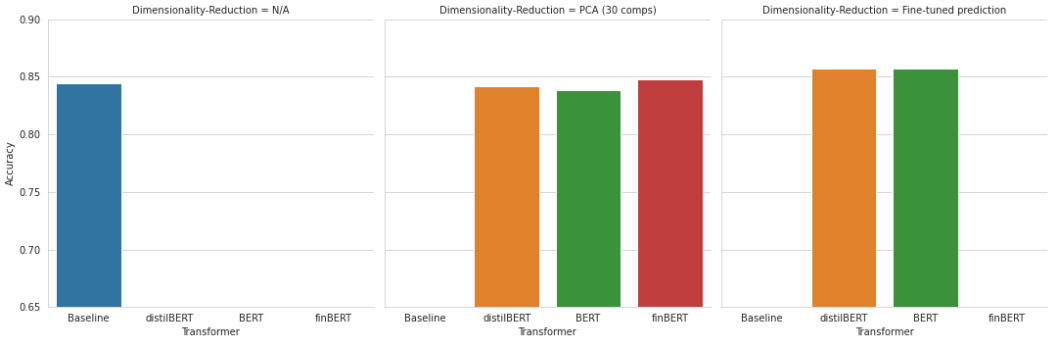
I assume that a neural net could be trained to outperform these tree-based methods, but it would require more subtle hyperparameter tuning than I was capable of. I used a two-layer neural-net with hidden layers of size 64 and 8, dropout layers, and batch normalization. Performance of the neural net severely degraded after roughly 2500 epochs. With this configuration, the best accuracy I could achieve with any set of features was 81%. Oftentimes, it was much worse.

5.4.2 Summary of Results

Table 5: Performance of Different Models on Test Set After 2 Epochs of Training

NLP Model	NLP Feature Set	Classifier Type	Accuracy	F1
None	N/A	Random Forests	0.82650	0.73109
distilBERT	PCA (30 comps)	Random Forests	0.82723	0.72292
distilBERT	Fine-tuned prediction	Random Forests	0.85827	0.78456
BERT	PCA (30 comps)	Random Forests	0.83075	0.73352
BERT	Fine-tuned prediction	Random Forests	0.85447	0.77474
finBERT	PCA (30 comps)	Random Forests	0.83089	0.73287
finBERT	Fine-tuned prediction	Random Forests	N/A	N/A
None	N/A	Boosted Trees	0.84451	0.75702
distilBERT	PCA (30 comps)	Boosted Trees	0.84158	0.75162
distilBERT	Fine-tuned prediction	Boosted Trees	0.85680	0.78613
BERT	PCA (30 comps)	Boosted Trees	0.83821	0.74930
BERT	Fine-tuned prediction	Boosted Trees	0.85725	0.78392
finBERT	PCA (30 comps)	Boosted Trees	0.84744	0.76641
finBERT	Fine-tuned prediction	Boosted Trees	N/A	N/A

Figure 3: Accuracy Rates of Gradient Boosted Tree Methods



5.4.3 Impressions

Table 5 indicates that, unfortunately, the supplementation of the models with transformers did not significantly improve performance. The best-performing supplemented model was a Random Forests Tree model supplemented with fine-tuned distilBERT predictions. Yet this model only exceeded the accuracy of the gradient-boosted baseline model by 1.4%. Such meager improvement is not surprising, given the generally poor performance of the fine-tuned models (as discussed in Section 5.3.3).

It should be noted that the PCA models performed the worst of all the models in the summary table. The PCA method of dimension-reduction can be counted as a failure.

6 Analysis

6.1 Identifiable Problems

- **Small Dataset** – The small size of the dataset was an easily foreseeable problem. In order to avoid overfitting, the neural-net methods used in this project—both the feed-forward networks used as classifier "heads" as well as the fine-tuned transformer models—all had to be stopped quite early in their training. This very well might explain their poor performance on the classification tasks.
- **Overfitting** – Although including the final hidden state of the transformers in the model seemed promising, the size of that hidden state (768 dimensions) was problematic and was

not remedied by PCA. In the worst case scenario, there is simply not enough meaningful information contained in the hidden state to merit inclusion in the model.

- **Imbalanced Data** – Each of the models tended to favor classifying months as "decreases" owing to the prevalence of "decreases" in the dataset.
- **Limits of a Binary Classifier** The binary response variable used in this project classifies every increase, no matter how small or large, as 1 (and every decrease, no matter its magnitude, as 0). This grouping masks great variation in the underlying data, especially as it pertains to increases in the unemployment rate, which exhibits large variance (Table 1).

6.2 Possible Solutions

- **Dividing the dataset by region** – I believe that I must revisit the decision to add the region indicator. Adding a region indicator possibly implies two things: (1) as a matter of time-series, the relation of past decreases/increases in unemployment rate to future decreases/increases varies from one region to the other; or (2) as a matter of NLP, the relation of the Beige Book statements to future decreases/increases in unemployment rate varies from one region to the other. I do not know that I can justify either of these implications. Furthermore, when I ran the vanilla time-series *without* the regional indicators, performance was just as high. This change would have the added benefit of increasing the effective size of the dataset by not subdividing the observations by region.
- **PCA** – Using PCA to reduce the size of the final hidden-state of the transformer models did not improve performance at all. I suspect that PCA does a generally poor job of capturing the variation among BERT hidden-states. At the same time, it is possible that I simply used too many PCA components. Were I to run the models again, I would choose 5 principal components, as the "elbow" in Figure 1 appears around 5 components.
- **Oversampling of "Increases"** In order to create a more balanced classifier, I would suggest oversampling "increase" observations.
- **Alternative Text Preprocessing** – I formulated the text preprocessing steps here on my own, from first principles. In particular, I included sentences containing certain word lemmas (e.g, "work", "employment") because they seemed to be the concepts most obviously relevant to the unemployment rate. But there are, obviously, many different ways that the preprocessing could have been performed. Different word lemmas could have been used, or fewer sentences could have been associated with each month's Beige Book. I suspect that some of the Beige Book entries that I included were too long for the transformer model to properly handle. Some of them approached the maximum of 512 tokens.
- **3-class Response Variable and Classifier** - To address the limitations of the binary response variable and classifier. The 3 classes: $-1 = (-\inf, -\varepsilon), 0 = [-\varepsilon, \varepsilon], 1 = (\varepsilon, \inf)$

7 Conclusion

This project started the journey for me of analyzing the Beige Books with the help of NLP. The lackluster performance of the supplemented models is compensated to some extent by the robust performance of the vanilla time-series model. 84.5% accuracy strikes me as enough to build an application around. I intend to pursue several of the solutions offered in Section 6.2.

References

[1] Sungil Kim Taeyoung Doh and Shu-Kuei Yang. How you say it matters: Text analysis of fomc statements using natural language processing, 2021.

[2] Bill Yuchen Lin, Chaoyang He, Zihang Zeng, Hulin Wang, Yufen Huang, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr. Fednlp: Benchmarking federated learning methods for natural language processing tasks, 2021.

[3] Oscar Suen. Github repository, 2020.

[4] Dogu Araci. FinBERT: Financial sentiment analysis with pre-trained language models. In *Thesis Submitted in Support of Masters of Data Science at the University of Amsterdam*, 2019.

A Appendix (optional)

Table 6: Correlation (as %) Among Unemployment Rates in the 12 Fed Districts:

	BO	NY	PH	CL	RI	AT	CH	SL	MI	KC	DA	SF
BO	100.0	94.0	88.1	67.7	80.2	74.2	68.7	66.0	66.5	66.1	46.5	87.4
NY	94.0	100.0	95.0	72.5	79.3	77.2	69.9	67.8	67.9	67.7	52.2	91.3
PH	88.1	95.0	100.0	88.6	87.6	87.3	85.1	82.7	83.1	78.2	60.4	92.5
CL	67.7	72.5	88.6	100.0	91.0	92.7	97.3	96.5	96.8	90.4	74.4	84.2
RI	80.2	79.3	87.6	91.0	100.0	92.9	93.8	94.3	94.0	91.1	67.5	93.2
AT	74.2	77.2	87.3	92.7	92.9	100.0	92.4	94.9	92.1	92.9	78.1	91.3
CH	68.7	69.9	85.1	97.3	93.8	92.4	100.0	98.9	98.9	92.3	71.6	83.6
SL	66.0	67.8	82.7	96.5	94.3	94.9	98.9	100.0	98.2	95.1	76.4	84.8
MI	66.5	67.9	83.1	96.8	94.0	92.1	98.9	98.2	100.0	93.3	72.4	83.6
KC	66.1	67.7	78.2	90.4	91.1	92.9	92.3	95.1	93.3	100.0	88.1	85.8
DA	46.5	52.2	60.4	74.4	67.5	78.1	71.6	76.4	72.4	88.1	100.0	68.8
SF	87.4	91.3	92.5	84.2	93.2	91.3	83.6	84.8	83.6	85.8	68.8	100.0

Table 7: Training distilBERT: Losses over Epochs

Model	Loss Metric	Epoch	Training Loss	Accuracy	F1
distilBERT	Cross-entropy	1	0.972435	0.553441	0.463237
distilBERT	Cross-entropy	2	0.947458	0.571010	0.482238
distilBERT	Cross-entropy	3	0.960134	0.547584	0.504236
distilBERT	Cross-entropy	4	0.993259	0.519766	0.513589
distilBERT	Cross-entropy	5	1.052546	0.528551	0.512969
distilBERT	Cross-entropy	6	1.09667	0.524158	0.528898

Figure 4: Unemployment Rates by Federal Reserve District

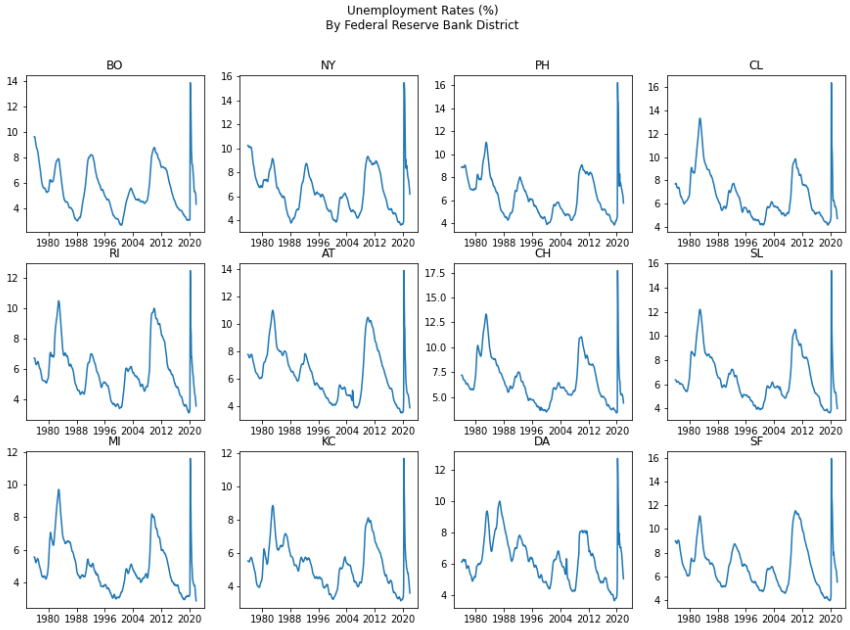


Figure 5: Month-over-Month Changes in the Unemployment Rate, by District