Comparative Post-Hoc Analysis of Open-Ended Survey Responses

Giovanni Maya

2025-05-29

Contents

Introduction

With the rise of social media in socio-political contexts, open-ended responses present rich opportunities for understanding public opinion beyond the scope of multiple-choice questions. However, as much as in other disciplines, open-ended responses have been notoriously difficult to quantitatively analyze at scale, often requiring human-in-the-loop mechanisms for validation. Topic modeling provides an avenue to tackle this problem, and among the earliest adoptions, Latent Dirichlet Allocation (LDA) was developed as a foundational method. However, this method lacked a major component: the ability to include pertinent covariates such as political affliation, ideology, or demographics, which are known to be heavily influential in socio-political contexts.

In their work, *Structural Topic Models for Open-ended Responses*, Roberts et. al (2014) address this concern with the Structural Topic Model, a probabilistic model that integrates document-level covariates into both an analysis of topic prevalence and topic content. The Structural Topic Model introduced many advances in metholodological innovation for causal inference and textual analysis, providing a backdrop for applicability in social science contexts.

In this project, we aim to replicate the core results of Roberts et. al including a reproduction of the validation metrics, STM model estimations, and generalized applicability to the 3 datasets outlined in the research. The main goal of this project then, is to apply a comparative analysis using modern-day technology, transformer architectures. We build on the original approach outlined in Roberts et. al by using BERTopic, a transformer-based topic model that uses contextual embeddings, dimensionality reduction, and clustering to generate topics. Unlike STM, we note that BERTopic does not model covariates directly. To address this issue, we propose a similar post-analytic framework to assess how respondent characteristics relate to topic assignment. This is accomplished by modeling the topic assignment probabilities as a function of covariates and their interactions with a logistic regression model. This enables us to test whether BERTopic is sufficiently able to discern the patterns of partisanship, demographic stratification, and treatment responsiveness. In doing so, we assess coherence, interpretability, and model sensitivity to known experimental effects, highlighting the pertinent trade-offs in modern generative and embedding-based topic modeling.

The Original Paper

Literature Context

The research provided by Roberts et al (2013) is situated in the intersection between computational text analysis and social science at large, particularly an analysis of open-ended survey response data and public opinion[1]. As mentioned in the introduction, unlike traditional bag-of-words approaches such as LDA, STM provides a mechanism by which to incorporate respondent metadata or analyze treatment effects across studies.

The Structural Topic Model builds upon earlier metholodologies aimed at analyzing open-ended responses. The authors in particular cite two notable motivations in their work: Simon and Xenos

[1] Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, and David G. Rand. 2014. "Structural Topic Models for Open-Ended Survey Responses." American Journal of Political Science 58(4): 1064–1082. https://doi.org/10.1111/ajps.12103

(2004) and Hopkins (2010). It is noteworthy to our work that we offer an understanding of the literature context, so we briefly outline them in minimal detail.

Simons and Xenos (2004) proposed a method utilizing factor analysis on word-frequency data to analyze political texts[2]. They outline 3 steps in their methodology: data preparation, exploratory factor analysis, and hypothesis testing. They, too, draw inspiration from latent semantic analysis (LSA), aiming to provide an alternative to traditional intersubjective content analysis. This approach, however, did not incorporate document-level covariates or model topic content variation.

Similarly, Hopkins (2010) developed a method for automated content analysis that estimates the proportion of documents falling to pre-specified categories[3]. Although this technique proved useful for large corpora where manual coding is impractical, it relied on pre-defined categories, and again, does not integrate metadata into the modeling process.

While Simon and Xenos (2004) and Hopkins (2010) laid important groundwork in the analysis of open-ended responses, STM built on these motivations via its method of covariate integration. The previous work was imperative to deeper questions STM sought to answer during this time.

Data

The paper utilizes a number of datasets to demonstrate the applicability and generalizability of their method. To validate the STM framework to recover known treatment effects, Roberts et al. conduct simulations using synthetic data generated from a simplified LDA generative process with exogenously induced treatment effects. To create the data, the simulation proceeds as follows: 1) word distributions are drawn with $\beta_k \sim \text{Dirichlet}(0.05)$, 2) topic prevalence vectors are modified by treatment: $\alpha_{t=0} = (0.3, 0.4, 0.3), \quad \alpha_{t=1} = (0.3 - \text{ATE}, 0.4, 0.3 + \text{ATE})$, and 3) each document is drawn as follows:

$$N_d \sim \text{Poisson}(\zeta = 40)$$

$$\theta_d \sim \text{Dirichlet}(\alpha_d \cdot G_0), \quad G_0 = \frac{1}{3}$$

$$\mathbf{w}_d \sim \text{Multinomial}(N_d, \theta_d \cdot \beta)$$

The out-of-sample dataset they consider at the end of the paper is the American National Election Studies (2008) where they gathered a conglomeration of respondent responses to being asked: “What is the most pressing issue the country is facing today?”. As with a study of this sort, information was gathered in regard to demographics and political affliation. It is important to note that unlike the two datasets to follow, the ANES data does not have a treatment - we will revisit this later. Table 1 shows a sample of this data.

To further validate STM performance, Roberts et al refer to Rand, Greene, and Nowak’s (2014) study that analyzed how intuitive versus reflective reasoning influences decision making in public goods game using a number of experimental conditions. The researchers analyze the respondents post-game reflections of their strategies and their relationship to game contributions. Table 2 demonstrates a sample of this data.

[2] Simon, Adam F., and Michael A. Xenos. 2004. “Dimensional Reduction of Word-Frequency Data as a Substitute for Intersubjective Content Analysis.” Political Analysis 12(1): 63–75. https://doi.org/10.1093/pan/mph004

[3] Hopkins, Daniel J., and Gary King. 2010. “A Method of Automated Nonparametric Content Analysis for Social Science.” American Journal of Political Science 54(1): 229–247. https://doi.org/10.1111/j.1540-5907.2009.00428.x

Age	PID Summary	Issue 1 (mippol1)	Issue 2 (mippol2)
35	4	the economy	immigration
39	5	how our government handles our economic crisis	gay marriage
50	3	the spending budget	education
72	6	economy no	terrorism
66	4	to do what your profess...	its a problem that we have political parties
56	4	morals	morals

Table 1: Sample of ANES Respondent Responses

Age	Primed	Contribution	Normalized	Risk	Strategy Description
22	1	0	0.0	4	to maximize my personal earnings
20	1	0	0.0	5	i could not trust that the other group members...
18	1	400	1.0	4	benefits the group the most
21	1	0	0.0	4	i would gain more by not adding money...
18	1	400	1.0	6	i am banking on the fact that people will be less selfish...

Table 2: Rand, Greee, and Nowak Sample: Contributions and Strategy Explanations

The last and final dataset Roberts et. al employed was Gadarian et. al (2012) experimental data. This dataset consists of 300+ responses from a survey experiment on immigration where respondents were randomly assigned to either think with reason or to think with fear in regard to attitudes toward immigration. A sample of this data is shown in Table 3.

Case ID	Treatment	Fear (ra1)	Republican ID	Open-Ended Response
287	1. worried	1	1.000	problems caused by the influx of illegal immigrants...
145	1. worried	1	1.000	i'm afraid of who might be getting into this country...
159	0. think	0	0.333	they should enter the same way my grandparents did...
421	0. think	1	0.500	legally entering the usa meeting the requirements is the law...
224	1. worried	2	0.667	terror, bombings, killing us, robbing america

Table 3: Gadarian Sample: Metadata and Immigration Responses

Methods

To facilitate broad adoption and ensure reproducibility, the authors developed and released the R package `stm` which includes built-in functionality for model estimation, visualiaztion, and hypothesis testing with various datasets and covariates. The package is emphasized to be accessible to applied researchers, carrying out variational expectation-maximization and automatic model selection tools (e.g., held-out likelihood and residuals for choosing the number of topics).

A central methodological contribution of STM is the ability to model covariate effects through a regression-like interface. They model topic prevalence as $\theta_d \sim \text{LogisticNormal}(\mu = X_d\gamma, \Sigma)$ where θ_d are topic proportions for document d, X_d is a matrix of document-level covariates (e.g., treatment, age, education), and γ are estimated coefficients showing how those covariates shift the prevalence of each topic. They implement smooth functions for continuous variables, interactions, and categorical methods.

Summary of Contributions and Findings

The findings are pertinent to evaluate the significance of this method and so, we discuss in relation to each dataset.

The simulation study developed aimed at determining whether STM was capable of recovering true treatment effects. This was systematically varied with number of documents (100 - 1000) and varying treatment effect sizes. They showed that with no treatment effect, the STM correctly estimated no difference. With moderate treatment effects, STM was able to recover the effect well over an increased sample size, demonstrating robustness to small treatment effects.

In the ANES open-ended responses, the STM correctly drew out topics pertaining to health care, economy, war, taxes, and abortion. A notable finding that aligns very well with previous research: Republicans with higher education are less likely to mention the war, whereas Democrats demonstrate the opposite, was also notably present in this analysis. This application was a testament to the role of negative interaction between education and party affliation on topic prevalence. The Rande, Green, and Nowak (2013) displays congruence. STM was able to recover the subtlety of the treatment effects, correctly distinguishing participants' decisions and vocabulary with which treatment condition they were a part of. The simple, yet powerful labeling of the experimental assignment allowed the model to be more precise as well. The Gadarian and Albertson Immigration (2014) experiment found that the "terrorism", "welfare" topic was significantly more prevalent in the treatment, confirming the authors' hypothesis that priming threat increases national security framing.

Overall, a synthesis across all datasets demonstrates that STM was able to: detect experimental treatment effects on topic usage, uncover complex interactions, and provide rich qualitative insight for political text data, proving to be incredibly significant in this domain.

Replication

Approach

To replicate the results and methods from Roberts et. al (2014), we systematically reconstructed the full empirical pipeline across each of the datasets in the original paper. We completed this task with access to a Harvard Dataverse folder of R code files and pre-processed versions of the data suitable for analysis. We note that the R code files were deprecated versions of the package. We referenced descriptions provided in the paper and appendix if further clarification was needed.

Simulation Test and Example

For the validation component, we implemented the generative model described in "Validating the Model: Simulations Tests and Examples" of the paper using a synthetic data generation pipeline in R. The topic proportions were manipulated via a treatment-induced hyperparameter ATE. The specification of the parameters are as outlined in the original paper's Appendix. Our Appendix displays the code in its entirety. The code generation process for documents is demonstrated below.

```r
set.seed(123)
K <- 3 # Number of topics
V <- 500 # Vocabulary size
Ns <- seq(100, 1000, 100) # Sample sizes
zeta <- 40 # Expected words per document (Poisson)
ATE_true <- 0.2 # True ATE size
GO <- 1 / 3 # Concentration parameter for Dirichlet

# Generate topic-word distributions (beta) from Dirichlet(.05)
generate_beta <- function(K, V) {
    beta <- matrix(0, nrow = K, ncol = V)
    for (k in 1:K) {
        beta[k, ] <- as.numeric(rdirichlet(1, rep(0.05, V)))
    }
    beta
}

# Simulate documents, with treatment assigned to half
simulate_docs <- function(N_docs, zeta, alpha_control, alpha_treated, beta) {
    docs <- list()
    treatment <- c(rep(0, N_docs / 2), rep(1, N_docs / 2))
    treatment <- sample(treatment) # Randomize order

    for (i in 1:N_docs) {
        alpha <- if (treatment[i] == 1) alpha_treated else alpha_control
        theta <- as.numeric(rdirichlet(1, alpha))
        Nd <- rpois(1, zeta)
        words <- integer(Nd)
        for (j in 1:Nd) {
            z <- sample(1:K, 1, prob = theta)
            w <- sample(1:V, 1, prob = beta[z, ])
            words[j] <- w
        }
        docs[[i]] <- words
    }
    list(docs = docs, treatment = treatment)
}
```

To confirm that the data generation process was sufficiently capturing the results of the researchers, we replicated Figure 2 from the paper as our Figure 1 below. From our Figure 1, we can note a couple of divergences from the original Figure 2. On the left, we see that the STM model performs well at not recovering the effect when it simply does not exist. As the number of documents increases, the estimates correctly cluster around 0, the true ATE value. On the right panel, we examine the treatment group. Unlike the original paper, we see consistently more variation as the number of documents increases, sporadically appearing near the designated ATE value, and on a couple of instances failuring to detect the treatment. That said, the confidence intervals do shrink, indicating increasing statistical power. Overall, the visualization does demonstrate the ability of the STM to recover both null and positive treatment effects.

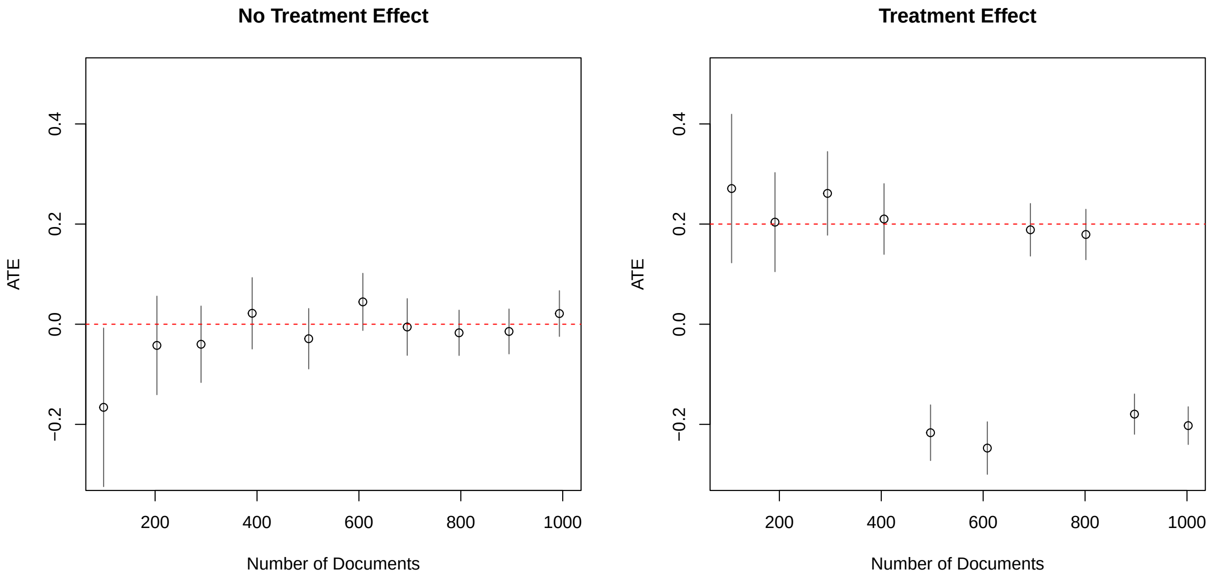


Figure 1: Simulated STM Treatment Effect Recovery

ANES (2008)

For the ANES 2008 study, we followed a similar empirical pipeline, albeit with more resources at our disposal. We were provided three files: the `ldac` containing the documents, a metadata csv containing demographics, and a file containing the vocabulary. We applied the same filtering used in the STM paper: respondents must have non-missing values for `pid_summary`, `age`, and `highest.grade.completed`. We proceeded to estimate a 60-topic STM to match the original paper with a prevalence formula of: `s(pid_summary) + s(age) + s(highest.grade.completed) + s(highest.grade.completed * pid_summary)` such that it captures smooth terms for party affliation, age, and education. An interaction term between education and party identification was included as well to mirror the analysis of moderation effects. The results of this analysis produced a model with 60 topics, all of which ranged with various top words. A replication of Figure 17 from the original paper is shown for our run in Figure 2.

Moreover, since the main strength of the framework is to model the relationship of covariates, we replicate the comparison between the ANES hand-coders and our implementation as done in Figure 18 of the original paper. To estimate the effect of education and party identification, a separate effects model was developed as a subset of the larger model with the prevalence formula: `~ highest.grade.completed * pid_group`. This allows us to investigate strictly the relationship between the topic prevalence and the interaction term. We visualize these results with the

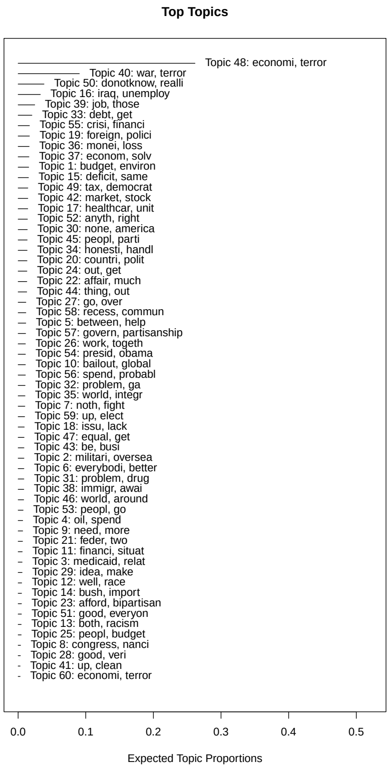


Figure 2: Words with Highest Topic Proportion Per Topic

`plot.estimateEffect` function from `stm` with overlays pertinent to both the Democrats and the Republicans. Figure 3 displays the results of this analysis. On the left panel, we examine the ANES human coders from the data. The human coders clearly denote trends pertaining to Democrats and Republicans. On one hand, the Democrats with higher education display a higher tendency to talk about the war topic. On the other, Republicans show the opposite trend: they talk less about war as they acquire higher education. Our STM model faithfully reproduces results in line with this analysis. Although we see negative decreases across both, we see that, consistent with the hand-coding, the Republicans display a starker decrease as education increases. Overall, the general trend remains consistent among both model representations.

Rand, Greene, and Nowak (2012)

We move our attention to the Rand, Green, and Nowak (2012) study and our empirical results. We replicate in particular the method of topic proportion differences. We replicate Figure 11 from the original paper as our Figure 4. In the original paper, topics 1 and 4 reflected intuition and egoism and self-preservation respectively. For our analysis, we found that topic 1 did not fit that category. We retrieved topic 3 as the best fit with terms such as “logic”, “thought”, “felt” and topic 4 remained as is. On the right side of Figure 4, we see that the estimate for topic 3 is negative indicating that it is more prevalent in the control group rather than the treatment group. This may suggest that treatment may decrease the use of moral language. Topic 4’s estimate is positive and significantly more common in the treated group. Our results do not entirely align with the paper, positing incongruence in model interpretability.

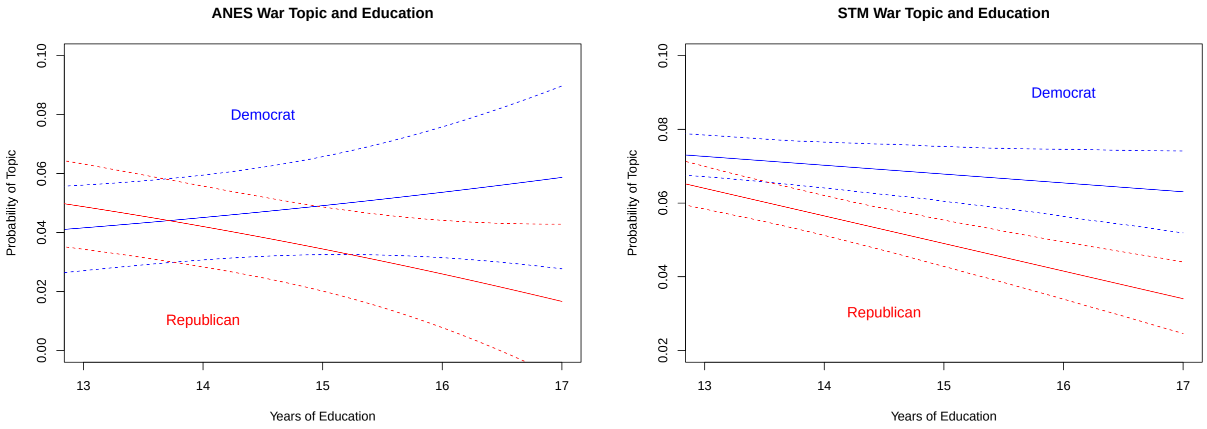


Figure 3: War Topic Prevalence by Education and PID

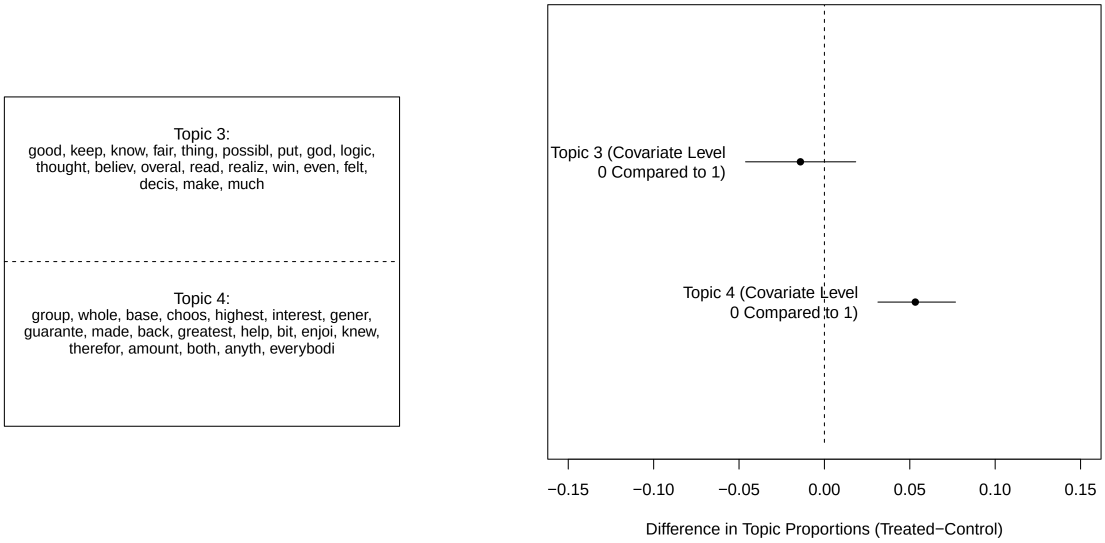


Figure 4: Topics from Intuition vs. Reflection Priming

As with ANES study, we conduct a replication of covariate relations. Figure 5 displays a continuous covariate plot of how topic proportions vary across document. Aligning with the results from the paper, we see that the expectation for individuals acting on intuition tend to contribute more, whereas individuals who rely on an instrumental frame exhibit a declining negative trend, and thus, contribute less.

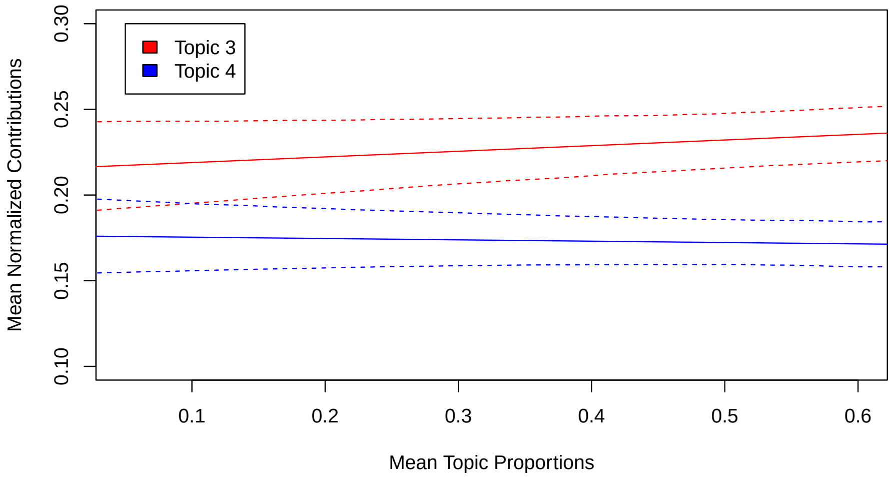


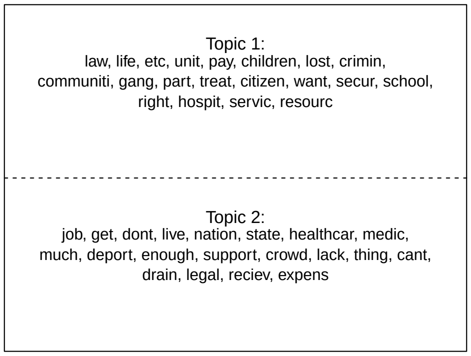
Figure 5: Intuition Topics and Contributions

Gadarian and Albertson Immigration (2014)

Lastly, we visit the Gadarian and Albertson Immigration study. The `stm` package comes nicely loaded with both a `gadarian` dataset and `gadarianFit` which denotes their own fitting of the model. Similar to previous methods, we relied on built-in methods and functions to replicate the analysis. To begin, we were able to successfully reproduce a 3-topic STM model as specified by the paper. We proceeded to gauge how well our topics aligned with the ones in the paper and we note a couple of divergences. In Figure 6, we examine the prominent words in both topic 1 and 2, as shown in the paper. However, unlike their results, our model produced 2 topics that appeared relatively indiscernable - both highly correlated with negative inclinations towards immigration with words such as criminal, gang, and lack.

To get a better representation of responses within each topic, we utilize the `findThoughts` function to extract quotes pertinent to each topic. For Topic 1, we retrieve: "loss of american culture. loss of rights, money and services to americans. lack of security. double standards. lack of unified english. end of american way of life", and for topic 2, we retrieve: "when i think of immigration, i think of the people who legally come to this country and apply for the necessary citizenships so that they can get a job and have the benefits of being an american and all the freedoms that come with that". With this simple extension, we see that topic 2 is more humanitarian and logical, whereas topic 1 is negative-emotion driven, implicitly outlining our treatment groups successfully and aligning our results with the paper.

Alike the ANES analysis, we are particularly interested in how covariates impact this relationship. In Figure 7, we visualize the predicted proportion of Topic 1 via an interaction between party



Topic 1:
law, life, etc, unit, pay, children, lost, crimin, communiti, gang, part, treat, citizen, want, secur, school, right, hospit, servic, resourc

Topic 2:
job, get, dont, live, nation, state, healthcar, medic, much, deport, enough, support, crowd, lack, thing, cant, drain, legal, reciev, expens

Figure 6: Vocabulary in Topics 1 and 2

identification and treatment. Cited as the negatively connotated topic, the treatment group demonstrates higher predicted proportions of Topic 1 prevalence. The control demonstrates feelings that are in line with intuition, displaying lower predicted proportions. The major divergence between our visualization and their visualization is the slope of prediction. Although we could not discern the two groups by big margins, the marginality of difference is still ever so present - aligning with our expectation.

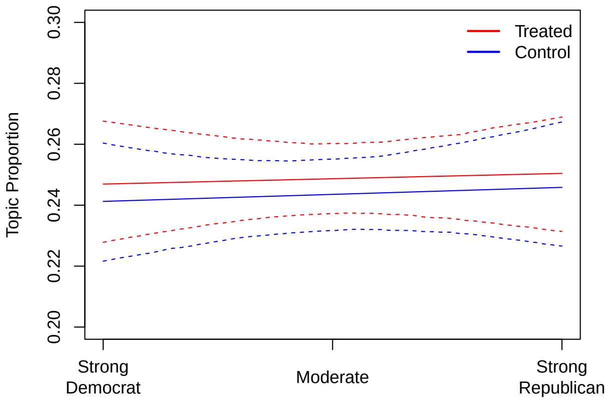


Figure 7: Party Identification, Treatment, and the Predicted Proportion in Topic 1

Extension

To extend the original analysis in Roberts et. al (2014), we implemented a complementary unsupervised topic modeling framework - BERTopic - across two notable datasets: ANES 2008 data and Gadarian and Albertson Immigration (2014). BERTopic, unlike STM, leverages pretrained transformer-based sentence embeddings, dimensionality reduction, and clustering algorithms to identify semantically meaningful topics from text without assuming a generative model structure. This post hoc framework allows for flexible topic discovery and comparison across experimental conditions.

Motivation

The motivation behind this decision was the application of a modern architecture to this problem. BERTopic uses contextual word embeddings to embed entire responses and discover clusters based on semantic similarity. This allows us to better examine alternative topic representations and how they alter results, how semantic coherence and topic labeling can differ (or stay the same) from high-dimensional embeddings, and lastly, if social or political applications would benefit from this analysis.

Methodology

For this extension, we developed a pipeline for extracting and analyzing latent topics in open-ended survey data using BERTopic. We conduct a post hoc covariate analysis in this setting to analyze topic prevalence similarly to the STM model.

In the form of mathematical formulaic, let:

- $D = \{d_1, \ldots, d_N\}$: a corpus of open-ended text responses
- X_i: metadata for respondent i (e.g., treatment, ideology, demographics)

The pipeline proceeds as follows:

1. Sentence embedding: each document d_i is mapped to a semantic vector:

$$e_i = \text{Encoder}(d_i) \in \mathbb{R}^p$$

We use a pretrained transformer-based encoder such as "all-MiniLM-L6-v2".

2. Dimensionality reduction: embeddings are projected into a low-dimensional latent space to preserve semantic similarity while enabling efficient clustering:

$$z_i = \text{UMAP}(e_i), \quad z_i \in \mathbb{R}^k$$

with $k \ll p$, typically 2–5 dimensions.

3. Topic clustering: documents are then clustered using an unsupervised clustering algorithm (e.g KMeans & HDBSCAN) to assign topic labels:

$$\text{topic}_i = \text{Clustering-Algorithm}(z_i), \quad \text{topic}_i \in \{1, \ldots, K\}$$

where K is the number of topics chosen.

4. Topic Representation: each topic T_j is denoted by a ranked list of informative terms extracted via TF-IDF weighting across documents for that topic:

$$T_j = \{w_{j1}, w_{j2}, \ldots, w_{jn}\}$$

where w_{jk} are the top n words ranked by importance to topic j.

5. Covariate Modeling: for a selected topic t^*, define a binary indicator:

$$y_i = \mathbb{I}(\text{topic}_i = t^*)$$

We model the relationship between topic assignment and respondent covariates using a generalized linear model (in our case, a logistic model):

$$\log\left(\frac{\Pr(y_i = 1)}{1 - \Pr(y_i = 1)}\right) = \beta_0 + \beta^\top X_i + \gamma^\top (X_i \odot Z_i)$$

where X_i contains main effects and $X_i \odot Z_i$ includes optional interactions (e.g., treatment × ideology).

From the fitted model, proceed to generate predicted topic assignment probabilities across a grid of covariate values:

$$\hat{p}_i = \Pr(y_i = 1 \mid X_i)$$

Results

Gadarian and Albertson Immigration (2014)

We apply the methodology outlined in the previous section to the Gadarian data. We employ UMAP with 5 components, and a kmeans model of 3 topics to fit with the original analysis. A table of top words per *representative* topic is shown in Table 4. The decision to exclude Topic 1 is that it did not align to the "welfare", "fear" characteristics described in the paper. Topic 0 better represents this.

Topic 0	**Topic 2**
jobs	immigration
people	think
illegal	immigrants
americans	people
welfare	country
care	need
taxes	legally
security	illegal
social	worry
immigrants	come

Table 4: Top Words in Topic 0 and 2

We proceed by emulating the Figure 7 from an earlier section using this model (Figure 8 from paper). Our Figure 8 displayed below shows the same Loess-smoothed line of the proportion of each response in Topic 0 on party identification. We see that our Figure 8 matches nearly perfectly with the figure in Roberts et. al, discriminating the fact that strong republicans in the treated group display a higher probability of being assigned to the topic than the control group. Alike the paper, this suggests that the treatment increases the prevalence of the target.

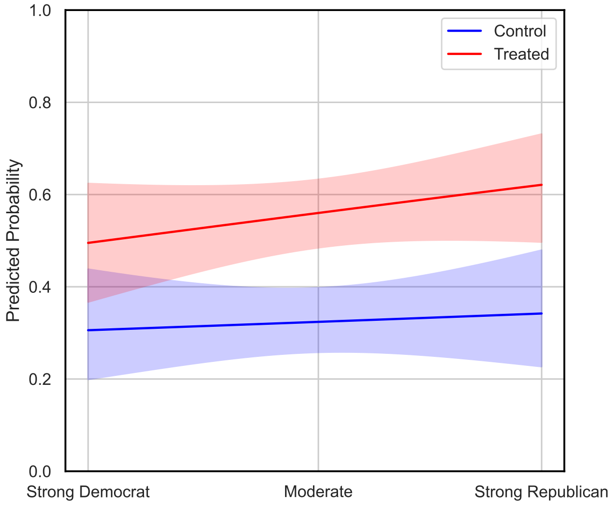


Figure 8: Party Identification, Treatment, and the Predicted Proportion in Topic 1

ANES 2008

We extend this analysis to the ANES 2008 data with different hyperparameters. To align with the paper, we attempted to fit 60 clusters - this did not work as the model only uncovered a maximum of 9. We used the same specification for UMAP as in the previous section. Two topics are shown in the Table 5 below. We examine that Topic 4 pertains entirely to President Barack Obama and of course, the election that is occurring during this time. Topic 5 aligns best with the topic of interest in the paper, topic 53, which pertains to fears about war and Iraq. Using this model, we conduct a covariate analysis in the consequent section.

Topic 4	**Topic 5**
president	war
obama	economy
black	wars
change	jobs
getting	irak
going	economic
think	ending
people	milatary
person	recovery
elected	enconomy

Table 5: Top Words in Topic 4 and 5 from ANES BERTopic Model

Figure 9 displays the results of the covariate analysis. From the visualization, we can denote a number of trends consistent with the paper and our own-iteration of STM. As education increases, we see that Democrats are more likely to speak on war. By contrast, we see that Republicans demonstrate considerably less impetus to do so regardless of education level. These results are congruent in nature with the original paper indicating that we discern nearly identical results from either analysis.

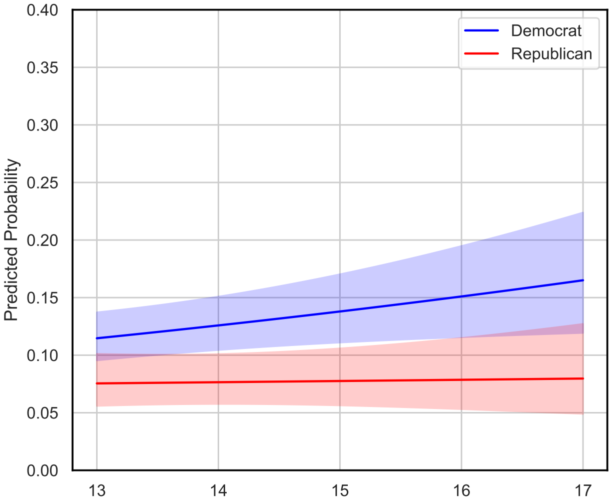


Figure 9: BERTopic War Topic and Education

Conclusion

In this project, we successfully replicated the Structural Topic Model analysis conducted by Roberts et al. (2014) across multiple datasets. Our efforts confirmed STM's ability to recover both null and true treatment effects on simulated data as well as meaningful topic-prevalence relationships in open-ended responses. We uncovered results that are in line with practical, human-encoding research and analyses.

Building on this foundation, our extension used BERTopic to examine if an embedding-based model can replicate the social-scientific insights. Despite BERTopic's lack of native covariate modeling, our post-hoc framework using logistic regression on topic assignments to uncover treatment and demographic effects produced highly comparable results to STM. We saw parallel partisan-education interactions in ANES and treatment effects in Gadarian's immigration study suggesting that BERTopic can be leveraged in modern-day to detect substantive group differences.

In all, this comparative analysis underscores the value of modern neural network models for qualitative inference while affirming the rigor and sensitivity of STM. However, this is at a considerable trade-off by means of complexity. Although BERTopic is readily available as a Python package, the underlying mechanisms are more complicated, and even in our framework, we relied on a number of unsupervised learning methods. Ultimately, STM remains superior for integrated covariate modeling and uncertainty quantification. The streamlined process and ease of translation make it an optimal, yet straightforward approach to including covariates into open-ended response data.

We see this work as a blueprint for researchers aiming to combine generative and neural approaches in post-survey analysis. Our code and outputs are reproducible, though further generalizability to other datasets and political science applications is needed to examine this further.

References

1. Roberts, Margaret E., Brandon M. Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, and David G. Rand. 2014. "Structural Topic Models for Open-Ended Survey Responses." *American Journal of Political Science* 58(4): 1064–1082. https://doi.org/10.1111/ajps.12103

2. Simon, Adam F., and Michael A. Xenos. 2004. "Dimensional Reduction of Word-Frequency Data as a Substitute for Intersubjective Content Analysis." *Political Analysis* 12(1): 63–75. https://doi.org/10.1093/pan/mph004

3. Hopkins, Daniel J., and Gary King. 2010. "A Method of Automated Nonparametric Content Analysis for Social Science." *American Journal of Political Science* 54(1): 229–247. https://doi.org/10.1111/j.1540-5907.2009.00428.x

Appendix

A1. Environment Setup

```r
library(stm)
library(MCMCpack)
library(ggplot2)
library(topicmodels)
library(slam)
library(Matrix)
library(xtable)
```

A2. Data Pipelines

A2.1 Simulated Validation Data

```r
# --- Parameters ---
set.seed(123)
K <- 3
V <- 500
Ns <- seq(100, 1000, 100)
zeta <- 40
ATE_true <- 0.2
G0 <- 1 / 3

# --- Helper Functions ---
generate_beta <- function(K, V) {
    beta <- matrix(0, nrow = K, ncol = V)
    for (k in 1:K) {
        beta[k, ] <- as.numeric(rdirichlet(1, rep(0.05, V)))
    }
    beta
}
```

```r
simulate_docs <- function(N_docs, zeta, alpha_control, alpha_treated, beta) {
    docs <- list()
    treatment <- c(rep(0, N_docs / 2), rep(1, N_docs / 2))
    treatment <- sample(treatment)

    for (i in 1:N_docs) {
        alpha <- if (treatment[i] == 1) alpha_treated else alpha_control
        theta <- as.numeric(rdirichlet(1, alpha))
        Nd <- rpois(1, zeta)
        words <- integer(Nd)
        for (j in 1:Nd) {
            z <- sample(1:K, 1, prob = theta)
            w <- sample(1:V, 1, prob = beta[z, ])
            words[j] <- w
        }
        docs[[i]] <- words
    }
    list(docs = docs, treatment = treatment)
}

convert_to_stm_format <- function(doc, vocab) {
    tab <- table(doc)
    word_indices <- match(names(tab), vocab) - 1
    rbind(word_indices, as.integer(tab))
}

extract_effect_row <- function(topic_effect, treat_row) {
    if (!treat_row %in% rownames(topic_effect)) {
        return(NULL)
    }
    est <- topic_effect[treat_row, "Estimate"]
    se <- topic_effect[treat_row, "Std. Error"]
    lower <- est - 1.96 * se
    upper <- est + 1.96 * se
    return(c(Estimate = est, Lower = lower, Upper = upper))
}

# --- Simulation & Estimation Loop ---
results <- data.frame(
    N = integer(), ATE = numeric(),
    lower = numeric(), upper = numeric(), condition = character()
)

for (effect_type in c("No Treatment Effect", "Treatment Effect")) {
    for (N in Ns) {
        cat(sprintf("\n--- N = %d, Condition = %s ---\n", N, effect_type))
        beta <- generate_beta(K, V)
        if (effect_type == "No Treatment Effect") {
            alpha_control <- c(0.3, 0.4, 0.3)
            alpha_treated <- c(0.3, 0.4, 0.3)
        } else {
            alpha_control <- c(0.3, 0.4, 0.3)
            alpha_treated <- c(0.3 - ATE_true, 0.4, 0.3 + ATE_true)
        }

        sim <- simulate_docs(N, zeta, alpha_control, alpha_treated, beta)
        meta_df <- data.frame(treatment = factor(sim$treatment))
        vocab <- as.character(1:V)
        docs_stm <- lapply(sim$docs, convert_to_stm_format, vocab = vocab)

        out <- tryCatch(
            prepDocuments(docs_stm,
                vocab = vocab, meta = meta_df,
                verbose = FALSE
            ),
            error = function(e) {
                cat("prepDocuments failed\n")
                return(NULL)
            }
        )
        if (is.null(out)) next

        fit <- tryCatch(
            stm(out$documents, out$vocab,
                K = K,
                prevalence = ~treatment, data = out$meta,
                verbose = FALSE, init.type = "Spectral"
            ),
            error = function(e) {
                cat("STM model failed\n")
                return(NULL)
            }
        )
        if (is.null(fit)) next

        effect <- tryCatch(
            estimateEffect(1:K ~ treatment, fit,
                meta = out$meta, uncertainty = "Global"
            ),
            error = function(e) {
                cat("estimateEffect failed\n")
                return(NULL)
            }
```

```r
        )
        if (is.null(effect)) next

        eff_summary <- summary(effect)
        topic_ates <- sapply(eff_summary$tables, function(tab) {
            if ("treatment1" %in% rownames(tab)) {
                return(abs(tab["treatment1", "Estimate"]))
            } else {
                return(NA)
            }
        })

        best_topic <- which.max(topic_ates)
        if (!is.na(best_topic)) {
            topic_effect <- eff_summary$tables[[best_topic]]
            if ("treatment1" %in% rownames(topic_effect)) {
                vals <- extract_effect_row(topic_effect, "treatment1")
                if (!is.null(vals)) {
                    results <- rbind(results, data.frame(
                        N = N,
                        ATE = as.numeric(vals["Estimate"]),
                        lower = as.numeric(vals["Lower"]),
                        upper = as.numeric(vals["Upper"]),
                        condition = effect_type
                    ))
                    cat(sprintf("Success (Topic %d)\n", best_topic))
                } else {
                    cat("Skipped: could not compute CI from selected topic\n")
                }
            } else {
                cat("Skipped: treatment1 row missing from best topic\n")
            }
        } else {
            cat("Skipped: could not select best topic\n")
        }
    }
}

# --- Plotting ---
results_no <- subset(results, condition == "No Treatment Effect")
results_yes <- subset(results, condition == "Treatment Effect")

par(mfrow = c(1, 2), mar = c(5, 4, 4, 2) + 0.1)

plot_ate_panel <- function(data, main_title, true_line) {
    x_jitter <- jitter(data$N, amount = 10)
    plot(data$N, data$ATE,
        ylim = c(-0.3, 0.5),
        pch = 1,
        xlab = "Number of Documents", ylab = "ATE",
        main = main_title,
        type = "n"
    )
    segments(x0 = x_jitter, y0 = data$lower, y1 = data$upper, col = "gray40")
    points(x_jitter, data$ATE, pch = 1, col = "black")
    abline(h = true_line, col = "red", lty = 2)
}

plot_ate_panel(results_no, "No Treatment Effect", 0)
plot_ate_panel(results_yes, "Treatment Effect", ATE_true)
```

A2.2 ANES (2008)

```r
# --- Setup ---
documents <- readLdac("data/final_anes.csv")
meta <- read.csv("data/final_anes_metadata.csv")
vocab <- read.csv("data/final_anes_vocab.csv")$term
documents <- documents[meta$pid_summary > 0 &
    meta$highest.grade.completed > 0 & meta$age > 0]
meta <- meta[meta$pid_summary > 0 &
    meta$highest.grade.completed > 0 & meta$age > 0, ]
data <- prepDocuments(documents = documents, vocab = vocab, meta = meta)
documents <- data$documents
vocab <- data$vocab
meta <- data$meta


# --- Fit Model ---
stm_stm_mod <- stm(documents, vocab,
    K = 60,
    prevalence = ~ s(pid_summary) + s(age) + s(`highest.grade.completed`)
        + s(`highest.grade.completed` * pid_summary),
    data = meta,
    init.type = "Spectral",
    seed = 1234
)

# --- Display Topics ---
labelTopics(stm_stm_mod, n = 15)
plot(stm_stm_mod, type = "summary", n = 2)

meta$pid_group <- ifelse(meta$pid_summary < 3, 1, 2)

effect_stm_model <- estimateEffect(
    formula = ~ highest.grade.completed * pid_group,
    stmobj = stm_stm_mod,
    metadata = meta,
    documents = documents,
    uncertainty = "Global"
)

# --- Plotting Interactions - Education & Partisanship ---
plot.estimateEffect(
    effect_stm_model,
    covariate = "highest.grade.completed",
    stm_moderator = "pid_group",
    stm_moderator.value = 1, # For Democrats
    topics = 40,
    method = "continuous",
    stm_model = stm_stm_mod,
    linecol = "blue",
    xlab = "Years of Education",
    ylab = "Probability of Topic",
    ylim = c(0.02, 0.1),
    labeltype = "prob",
    xlim = c(13, 17),
    main = "STM War Topic and Education",
    printlegend = FALSE,
    ci.level = 0.9
)

plot.estimateEffect(
    effect_stm_model,
    covariate = "highest.grade.completed",
    stm_moderator = "pid_summary",
    stm_moderator.value = 2, # For Republicans
    topics = 40,
    method = "continuous",
    stm_model = stm_stm_mod,
    linecol = "red",
    labeltype = "prob",
    add = TRUE,
    ci.level = 0.9
)

text(16, 0.09, "Democrat", col = "blue", cex = 1.2)
text(14.5, 0.03, "Republican", col = "red", cex = 1.2)

anes_code <- 3
meta$warall <- as.integer(rowSums(meta[
    ,
    paste0("mippol1_code", 1:8)
] == anes_code, na.rm = TRUE) > 0)

sub_dem <- meta$pid_summary < 3
sub_rep <- meta$pid_summary >= 3

# --- Plotting Loess Interaction ---
dem_df <- meta[sub_dem, ]
fit_dem <- loess(warall ~ highest.grade.completed, data = dem_df, span = 2)
x_dem <- seq(min(dem_df$highest.grade.completed),
    max(dem_df$highest.grade.completed),
    length.out = 100
)
pred_dem <- predict(fit_dem,
    newdata =
        data.frame(highest.grade.completed = x_dem), se = TRUE
)

rep_df <- meta[sub_rep, ]
fit_rep <- loess(warall ~ highest.grade.completed, data = rep_df, span = 2)
x_rep <- seq(min(rep_df$highest.grade.completed),
    max(rep_df$highest.grade.completed),
    length.out = 100
)
pred_rep <- predict(fit_rep,
    newdata =
        data.frame(highest.grade.completed = x_rep), se = TRUE
)

plot(x_dem, pred_dem$fit,
    type = "l", col = "blue",
    xlim = c(13, 17),
    ylim = c(0, .1),
    xlab = "Years of Education",
    ylab = "Probability of Topic",
    main = "ANES War Topic and Education"
)

lines(x_dem, pred_dem$fit + 1.64 * pred_dem$se, col = "blue", lty = 2)
lines(x_dem, pred_dem$fit - 1.64 * pred_dem$se, col = "blue", lty = 2)

lines(x_rep, pred_rep$fit, col = "red")
lines(x_rep, pred_rep$fit + 1.64 * pred_rep$se, col = "red", lty = 2)
lines(x_rep, pred_rep$fit - 1.64 * pred_rep$se, col = "red", lty = 2)

text(14, 0.01, "Republican", col = "red", cex = 1.2)
text(14.5, 0.08, "Democrat", col = "blue", cex = 1.2)
```

```r
# --- Human vs. STM Comparison ---
count_topic_match <- function(topic_ids, threshold = 0.15) {
    rowSums(stm_stm_mod$theta[, topic_ids] > threshold) > 0
}


econ_topics <- c(56, 19, 20, 36, 39)
war_topics <- c(16, 40, 48)
dk_topics <- c(50, 30)
job_topics <- c(36, 39, 19, 48)

econ_stm <- sum(count_topic_match(econ_topics))
war_stm <- sum(count_topic_match(war_topics))
dk_stm <- sum(count_topic_match(dk_topics))
job_stm <- sum(count_topic_match(job_topics))


econ_hand <- sum(meta$mippol1_code1 %in% 50:55)
war_hand <- sum(meta$mippol1_code1 %in% c(3, 4))
dk_hand <- sum(meta$mippol1_code1 == 95)
job_hand <- sum(meta$mippol1_code1 == 27)

comparison_table <- data.frame(
    `STM Topic` = c(
        "Economy", "War or Iraq War",
        "Don’t Know", "Unemployment and Job"
    ),
    `STM Count` = c(econ_stm, war_stm, dk_stm, job_stm),
    `ANES Topic` = c(
        "The Economy", "War, or Iraq War",
        "Don’t Know", "Employment"
    ),
    `Hand-Coding Count` = c(econ_hand, war_hand, dk_hand, job_hand)
)

# --- Printing ---
library(xtable)
print(xtable(comparison_table,
    caption = "Comparison of STM to Hand Coding", align = "llrrr"
), type = "html")
```

A2.3 Rand, Greene, and Nowak (2012)

```r
# --- Load and Prepare Data ---
set.seed(123)
```

```
dataldac <- readLdac("data/final_tdm.csv")
vocabcoarse <- read.csv("data/tdm1011_vocab.csv")$term
meta <- read.csv("data/tdm1011_metadata.csv")
meta$treatment <- as.numeric(meta$condition_number == 10)

data <- prepDocuments(dataldac, vocabcoarse, meta = meta)
documents <- data$documents
vocab <- data$vocab
meta <- data$meta

# --- Fit STM Model ---
model <- stm(documents, vocab, 5, prevalence = ~ meta$treatment)

# --- Estimate Treatment Effect on Topic Prevalence ---
effect <- estimateEffect(1:5 ~ treatment,
    stmobj = model, metadata = meta,
    uncertainty = "Global"
)

# --- Get Top Words for Topics ---
top_words <- labelTopics(model, n = 20)
topic3_words <- strwrap(paste(top_words$frex[3, ],
    collapse = ", "
), width = 60)
topic4_words <- strwrap(paste(top_words$frex[4, ],
    collapse = ", "
), width = 60)
topic5_words <- strwrap(paste(top_words$frex[5, ],
    collapse = ", "
), width = 60)

# --- Plot: Treatment Effect for Topics 3 & 4 ---
layout(matrix(1:2, nrow = 1), widths = c(0.9, 1.1))
par(mar = c(0, 0, 0, 0))
plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1))
rect(0.1, 0.2, 0.9, 0.8)
segments(0.1, 0.5, 0.9, 0.5, lty = "dashed")
text(0.5, 0.73, "Topic 3:", font = 1, adj = 0.5)
y1 <- 0.70
for (line in topic3_words) {
    text(0.5, y1, line, cex = 0.9, adj = 0.5)
    y1 <- y1 - 0.03
}
text(0.5, 0.43, "Topic 4:", font = 1, adj = 0.5)
y2 <- 0.40
for (line in topic4_words) {
```

```r
    text(0.5, y2, line, cex = 0.9, adj = 0.5)
    y2 <- y2 - 0.03
}
par(mar = c(5, 4, 2, 1))
plot.estimateEffect(
    effect,
    topics = c(3, 4),
    covariate = "treatment",
    method = "difference",
    cov.value1 = 0,
    cov.value2 = 1,
    xlab = "Difference in Topic Proportions (Treated-Control)",
    labeltype = "numbers",
    xlim = c(-.15, .15)
)

# --- Plot: Treatment Effect for Topics 3 & 5 ---
layout(matrix(1:2, nrow = 1), widths = c(0.9, 1.1))
par(mar = c(0, 0, 0, 0))
plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1))
rect(0.1, 0.2, 0.9, 0.8)
segments(0.1, 0.5, 0.9, 0.5, lty = "dashed")
text(0.5, 0.73, "Topic 3:", font = 1, adj = 0.5)
y1 <- 0.70
for (line in topic3_words) {
    text(0.5, y1, line, cex = 0.9, adj = 0.5)
    y1 <- y1 - 0.03
}
text(0.5, 0.43, "Topic 5:", font = 1, adj = 0.5)
y2 <- 0.40
for (line in topic5_words) {
    text(0.5, y2, line, cex = 0.9, adj = 0.5)
    y2 <- y2 - 0.03
}
par(mar = c(5, 4, 2, 1))
plot.estimateEffect(
    effect,
    topics = c(3, 5),
    covariate = "treatment",
    method = "difference",
    cov.value1 = 0,
    cov.value2 = 1,
    xlab = "Difference in Topic Proportions (Treated-Control)",
    labeltype = "numbers",
    xlim = c(-.1, .1)
)
```

```r
# --- Estimate Effect of Normalized Contribution on Topic Proportions ---
meta$normalized.contribution <-
    as.numeric(as.character(meta$normalized.contribution))
effect_norm <- estimateEffect(
    formula = c(3, 4) ~ normalized.contribution,
    stmobj = model,
    metadata = meta,
    documents = documents,
    uncertainty = "Global"
)
plot.estimateEffect(
    effect_norm,
    covariate = "normalized.contribution",
    method = "continuous",
    topics = c(3, 4),
    xlab = "Mean Topic Proportions",
    ylab = "Mean Normalized Contributions",
    labeltype = "numbers",
    linecol = c("red", "blue"),
    xlim = c(0.05, 0.6),
    ylim = c(0.1, 0.3)
)

# --- Content Covariate Perspective Plot (Gender) ---
meta$gender <- factor(ifelse(meta$female == 1, "female", "male"))
model_with_content <- stm(
    documents = documents,
    vocab = vocab,
    K = 5,
    prevalence = ~ meta$treatment,
    content = ~ meta$gender,
    init.type = "Spectral"
)
plot(model_with_content,
    type = "perspectives",
    topics = 1,
    covariate = "gender",
    cov.value1 = "female",
    cov.value2 = "male",
    text.cex = 1.1,
    plabels = c("Female", "Male")
)
```

A2.4 Gadarian and Albertson Immigration (2014)

```r
# --- 1. Extract Top Words (Figure 4) ---
top_words <- labelTopics(gadarianFit, n = 20)
topic1_words <- strwrap(paste(top_words$frex[1, ],
    collapse = ", "
), width = 60)
topic2_words <- strwrap(paste(top_words$frex[2, ],
    collapse = ", "
), width = 60)
top_words

par(mar = c(0, 0, 0, 0))
plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1))
rect(0.1, 0.2, 0.9, 0.8)
segments(0.1, 0.5, 0.9, 0.5, lty = "dashed")

text(0.5, 0.73, "Topic 1:", font = 1, adj = 0.5)
y1 <- 0.70
for (line in topic1_words) {
    text(0.5, y1, line, cex = 0.9, adj = 0.5)
    y1 <- y1 - 0.035
}

text(0.5, 0.43, "Topic 2:", font = 1, adj = 0.5)
y2 <- 0.40
for (line in topic2_words) {
    text(0.5, y2, line, cex = 0.9, adj = 0.5)
    y2 <- y2 - 0.035
}

# --- 2. Representative Quotes (Figures 5 & 6) ---
thought <- findThoughts(gadarianFit,
    texts = gadarian$open.ended.response, n = 10
)

# Topic 1 Quotes
par(mar = c(1, 1, 1, 1))
plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1))
rect(0.05, 0.05, 0.95, 0.95)
segments(0.05, 0.5, 0.95, 0.5, lty = "dashed")

text1 <- strwrap(thought$docs[[1]][6], width = 60)
y_start1 <- 0.82
for (line in text1) {
    text(0.5, y_start1, line, cex = 0.9)
    y_start1 <- y_start1 - 0.04
```

```r
}

text2 <- strwrap(thought$docs[[1]][4], width = 60)
y_start2 <- 0.35
for (line in text2) {
    text(0.5, y_start2, line, cex = 0.9)
    y_start2 <- y_start2 - 0.04
}

# Topic 2 Quotes
par(mar = c(1, 1, 1, 1))
plot.new()
plot.window(xlim = c(0, 1), ylim = c(0, 1))
rect(0.05, 0.05, 0.95, 0.95)
segments(0.05, 0.5, 0.95, 0.5, lty = "dashed")

text3 <- strwrap(thought$docs[[2]][5], width = 40)
y_start3 <- 0.75
for (line in text3) {
    text(0.5, y_start3, line, cex = 0.9)
    y_start3 <- y_start3 - 0.04
}

text4 <- strwrap(thought$docs[[2]][3], width = 40)
y_start4 <- 0.40
for (line in text4) {
    text(0.5, y_start4, line, cex = 0.9)
    y_start4 <- y_start4 - 0.04
}

# --- 3. Estimate Treatment Effect (Figure 7) ---
treatment_effect <- estimateEffect(
    1:2 ~ treatment,
    gadarianFit,
    metadata = gadarian,
    uncertainty = "Global"
)
topic_labels <- c("Topic 1", "Topic 2")
plot(
    treatment_effect,
    covariate = "treatment",
    method = "difference",
    cov.value1 = 0,
    cov.value2 = 1,
    xlab = "Difference in Topic Proportions (Treated-Control)",
    labeltype = "custom",
    custom.labels = topic_labels,
    xlim = c(-0.1, 0.1)
)

# --- 4. Interaction Plot by Party ID (Figure 9) ---
interaction_effect <- estimateEffect(
    formula = c(1) ~ treatment * pid_rep,
    gadarianFit,
    metadata = gadarian,
    uncertainty = "Global"
)

plot.estimateEffect(
    interaction_effect,
    covariate = "pid_rep",
    moderator = "treatment",
    moderator.value = 1,
    method = "continuous",
    topics = 1,
    model = gadarianFit,
    linecol = "red",
    ylab = "Topic Proportion",
    xaxt = "n",
    printlegend = FALSE,
    labeltype = "prob",
    ylim = c(0.2, 0.3)
)

plot.estimateEffect(
    interaction_effect,
    covariate = "pid_rep",
    moderator = "treatment",
    moderator.value = 0,
    method = "continuous",
    topics = 1,
    model = gadarianFit,
    linecol = "blue",
    add = TRUE,
    printlegend = FALSE,
    labeltype = "prob"
)

axis(
    1,
    at = c(0, 0.5, 1),
    labels = c(
        "Strong\nDemocrat",
        "Moderate", "Strong\nRepublican"
    ),
    padj = 0.5
)

legend(
    "topright",
    legend = c("Treated", "Control"),
    col = c("red", "blue"),
    lty = 1,
    lwd = 2,
    bty = "n"
)
```

A4. Extension Model (BERTopic)

A4.1 Imports

```
import pandas as pd
from sklearn.cluster import KMeans
from bertopic import BERTopic
from sklearn.feature_extraction.text import CountVectorizer
from sentence_transformers import SentenceTransformer
from umap import UMAP
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
```

A4.2 Gadarian + BERTopic

```
# --- Load and Prepare Data ---
DATA_DIR = "/Users/giomhern/04 Projects/topic-models/data"
df = pd.read_csv(f"{DATA_DIR}/gadarian_bertopic_input.csv")
texts = df["open.ended.response"].astype(str).tolist()

# --- Fit BERTopic with KMeans ---
embedding_model = SentenceTransformer("all-MiniLM-L6-v2")
umap_model = UMAP(n_neighbors=15, n_components=5,
min_dist=0.0, metric="cosine", random_state=42)
vectorizer_model = CountVectorizer(stop_words="english")
kmeans_model = KMeans(n_clusters=3, random_state=42)

topic_model = BERTopic(
    embedding_model=embedding_model,
    umap_model=umap_model,
    vectorizer_model=vectorizer_model,
    hdbscan_model=kmeans_model,
    calculate_probabilities=False,
    verbose=True
)

topics, _ = topic_model.fit_transform(texts)

# --- Label Topics ---
df["topic"] = topics
topic_labels = {
    0: "Economic Costs",
    1: "Border Control",
    2: "Moral Reasoning"
}
df["topic_label"] = df["topic"].map(topic_labels)

# --- Inspect Top Words per Topic ---
for topic_idx in topic_model.get_topics().keys():
    print(f"\n--- Topic {topic_idx} ---")
    for word, weight in topic_model.get_topic(topic_idx)[:15]:
        print(f"{word:<15} {weight:.5f}")

# --- Logistic Regression with Interaction ---
df["is_topic_1"] = (df["topic"] == 0).astype(int)
df["pid_centered"] = df["pid_rep"] - df["pid_rep"].mean()
df["interaction"] = df["pid_centered"] * df["treatment"]

X = sm.add_constant(df[["treatment",
"pid_centered", "interaction"]])
y = df["is_topic_1"]
model = sm.Logit(y, X).fit(disp=0)

# --- Prediction Grid ---
pid_vals = np.linspace(df["pid_rep"].min(),
df["pid_rep"].max(), 100)
grid = []
for t in [0, 1]:
    for pid in pid_vals:
        centered = pid - df["pid_rep"].mean()
        grid.append({
            "const": 1,
            "treatment": t,
            "pid_centered": centered,
            "interaction": centered * t,
            "pid_rep": pid,
            "label": "Treated" if t == 1 else "Control"
        })
```

```
pred_df = pd.DataFrame(grid)
pred_X = pred_df[["const", "treatment",
"pid_centered", "interaction"]]

# --- Predict with Confidence Intervals ---
pred = model.get_prediction(pred_X).summary_frame(alpha=0.05)
pred_df["predicted"] = pred["predicted"]
pred_df["lower"] = pred["ci_lower"]
pred_df["upper"] = pred["ci_upper"]

# --- Plot Predicted Probabilities by PID ---
plt.figure(figsize=(6, 5))

for label, color in zip(["Control", "Treated"], ["blue", "red"]):
    subset = pred_df[pred_df["label"] == label]
    plt.plot(subset["pid_rep"], subset["predicted"],
    color=color, label=label)
    plt.fill_between(subset["pid_rep"], subset["lower"],
    subset["upper"], color=color, alpha=0.2)

plt.xticks(
    [df["pid_rep"].min(), df["pid_rep"].mean(),
    df["pid_rep"].max()],
    labels=["Strong Democrat", "Moderate",
    "Strong Republican"]
)
plt.xlabel("")
plt.title("")
plt.ylabel("Predicted Probability")
plt.ylim(0, 1)

for spine in plt.gca().spines.values():
    spine.set_edgecolor("black")

plt.legend()
plt.tight_layout()
plt.show()
```

A4.3 ANES + BERTopic

```
# --- Load and Prepare Data ---
DATA_DIR = "/Users/giomhern/04 Projects/topic-models/data"
df = pd.read_csv(f"{DATA_DIR}/final_anes_metadata.csv")
df["text"] = df[["mii_1", "mii_2"]].fillna("").agg(" ".join, axis=1).str.strip()
df = df[
    (df["pid_summary"] > 0) &
    (df["highest grade completed"] > 0) &
    (df["age"] > 0) &
    (df["text"] != "") &
    df["female"].notna()
].copy()

texts = df["text"].tolist()

# --- Fit BERTopic Model ---
embedding_model = SentenceTransformer("all-MiniLM-L6-v2")
umap_model = UMAP(n_neighbors=15, n_components=5,
min_dist=0.0, metric="cosine", random_state=42)
vectorizer_model = CountVectorizer(stop_words="english")
kmeans_model = KMeans(n_clusters=10, random_state=42)

topic_model = BERTopic(
    embedding_model=embedding_model,
    umap_model=umap_model,
    vectorizer_model=vectorizer_model,
    hdbscan_model=kmeans_model,
    calculate_probabilities=False,
    verbose=True
)

topics = topic_model.fit_transform(texts)[0]
df["topic"] = topics

# --- Regression Prep ---
target_topic = 5
df["is_topic"] = (df["topic"] == target_topic).astype(int)
df["is_republican"] = (df["pid_summary"] >= 4).astype(int)
df["edu_centered"] =
    df["highest grade completed"] - df["highest grade completed"].mean()
df["interaction"] = df["edu_centered"] * df["is_republican"]

X = sm.add_constant(df[["edu_centered",
"is_republican", "interaction"]])
y = df["is_topic"]
model = sm.Logit(y, X).fit(disp=0)

# --- Prediction Grid ---
edu_vals = np.linspace(13, 17, 100)
grid = []
for party in [0, 1]:
    for edu in edu_vals:
        edu_c = edu - df["highest grade completed"].mean()
        grid.append({
            "const": 1,
            "edu_centered": edu_c,
            "is_republican": party,
            "interaction": edu_c * party,
            "education": edu,
            "label": "Republican" if party else "Democrat"
        })

pred_df = pd.DataFrame(grid)
pred_X = pred_df[["const", "edu_centered",
"is_republican", "interaction"]]
pred = model.get_prediction(pred_X).summary_frame(alpha=0.05)
pred_df["predicted"] = pred["predicted"]
pred_df["lower"] = pred["ci_lower"]
pred_df["upper"] = pred["ci_upper"]

# --- Plot Predicted Probabilities ---
plt.figure(figsize=(6, 5))

for label, color in zip(["Democrat", "Republican"], ["blue", "red"]):
    subset = pred_df[pred_df["label"] == label]
    plt.plot(subset["education"], subset["predicted"],
    color=color, label=label)
    plt.fill_between(subset["education"],
    subset["lower"], subset["upper"],
    color=color, alpha=0.2)

plt.xticks([13, 14, 15, 16, 17])
plt.xlabel("")
plt.ylabel("Predicted Probability")
plt.title("")
plt.ylim(0, 0.4)

for spine in plt.gca().spines.values():
    spine.set_edgecolor("black")

plt.legend()
plt.tight_layout()
plt.show()
```