Using to AI to Understand the Power of Executive Orders

By: Jason Eappen and Turner Lent

Introduction:

Executive orders have been a crucial and evolving tool of presidential authority in the United States. The first known executive order, issued on June 8, 1789, by President George Washington, requested detailed reports from department heads in order to better manage the newly formed government. It stated "For this purpose I wish to receive in writing such a clear account of the Department at the head of which you have been, as may be sufficient (without overburdening or confusing a mind which has very many objects to claim its attention at the same instant) to impress me with a full, precise & distinct general idea of the United States, so far as they are comprehended in, or connected with that Department".[1] Although the term "executive order" would not be coined until the Lincoln administration, the early use of presidential directives established a precedent for presidential oversight and influence over the operations of the executive branch. Since Washington's presidency, every president has issued executive orders.[2]

Importantly, the U.S. Constitution does not directly mention the term executive orders, but grants broad executive powers through Article II. Specifically, The Vesting Clause in Section I states that "the executive power shall be vested in a President," while the Take Care Clause in Section III obligates the President to "take care that the laws be faithfully executed." Interpreting the meaning of these clauses has been subject to debate, but holistically, the Vesting Clause is the

[1] "Founders Online: From George Washington to John Jay, 8 June 1789." *Founders.archives.gov*, founders.archives.gov/documents/Washington/05-02-02-0335. Accessed 27 Sept. 2025.
[2] RUDALEVIGE, A. (2021). *By Executive Order: Bureaucratic Management and the Limits of Presidential Power*. Princeton University Press. https://doi.org/10.2307/j.ctv17nmzjz

source of the President's broad executive powers and authority,[3] and the Take Care Clause gives the President discretion in how to enforce the laws.[4] These constitutional provisions have underpinned Supreme Court rulings that affirm the legitimacy of executive orders as presidential tools.

Over American history, there is a notable expansion, as well as critical documentation, of the evolving usage of executive orders. During the late 1700s and early 1800s, executive orders were used primarily for administrative functions of the executive branch. For example, on January 22nd, 1791, an executive order was issued by George Washington ordering the formation of a commission to find a permanent plot of land for the government of the United States.[5] Washington continued to use executive orders to manage the day-to-day operations of the government. Subsequent presidents followed Washington's precedent of using executive orders to assert control over the bureaucracy. As the American government continued to expand, managing the bureaucracy became larger and increasingly difficult. Over time, the courts have permitted presidents to exert more expansive executive orders allowing them to grow strong, expanding the power of the executive branch. For example, during the Civil War, President Lincoln used executive orders to suspend *habeas corpus* in various parts of the United States.[6]

---

[3]Marshall, William. "Interpretation: Article II, Section 3 | the National Constitution Center." *National Constitution Center – Constitutioncenter.org*, constitutioncenter.org/the-constitution/articles/article-ii/clauses/348. Accessed 27 Sept. 2025.

[4]Marshall, William. "Interpretation: Article II, Section 3 | the National Constitution Center." *National Constitution Center – Constitutioncenter.org*, constitutioncenter.org/the-constitution/articles/article-ii/clauses/348. Accessed 27 Sept. 2025.

[5]https://www.presidency.ucsb.edu/documents/commission-appointing-commissioners-for-surveying-the-district-territory-for-the-permanent

[6] A Proclamation on the Suspension of Habeas Corpus, 1862 | Gilder Lehrman Institute of American History." *Www.gilderlehrman.org*, 2013, www.gilderlehrman.org/history-resources/spotlight-primary-source/proclamation-suspension-habeas-corpus-1862. Accessed 27 Sept. 2025.

More famously, he also issued Proclamation 95 – the Emancipation Proclamation[7] – freeing slaves throughout the Confederate States of America. This continued to encourage presidents to use broad, sweeping executive orders to accomplish policy objectives. As American politics became more polarized, presidents then relied on executive orders as a means to achieve a policy outcome. President Franklin Roosevelt was famous for his sweeping use of executive orders, issuing over 3,726 orders during the course of his presidency[8]. With executive orders becoming more resourceful to presidents, critics often argue that executive orders are expanding presidential power because of decreased congressional capacity and increased polarization.[9]

Modern presidents have been noted for their use of executive orders to achieve key campaign promises through administrative action. In addition to fulfilling campaign promises, modern presidents have also used executive orders to bypass the frustrating polarity of a divided government. As a result, between President George W. Bush and President Donald Trump's second term (as of September 2025), a total of 1,155 executive orders have been issued.[10] [11] President Obama is famously quoted as saying, "I've got a pen, and I've got a phone. And I can use that pen to sign executive orders and take executive actions and administrative actions that move the ball forward".[12] Whether or not the critics are correct, it is in conclusive that executive orders have become a powerful tool of the president that continues to shape modern politics.

---

[7]"The Emancipation Proclamation, 1863." *Www.archives.gov*, www.archives.gov/exhibits/american_originals_iv/sections/text_emancipation.html. Accessed 27 Sept. 2025.
[8]The American Presidency Project. "Executive Orders." *Ucsb.edu*, 2019, www.presidency.ucsb.edu/statistics/data/executive-orders. Accessed 27 Sept. 2025.
[9] Carmines, E., & Fowler, M. (2017). The Temptation of Executive Authority: How Increased Polarization and the Decline in Legislative Capacity Have Contributed to the Expansion of Presidential Power. *Indiana Journal of Global Legal Studies*, 24, 369 - 397. https://doi.org/10.2979/indjglolegstu.24.2.0369.
[10] The American Presidency Project. "Executive Orders." *Ucsb.edu*, 2019, www.presidency.ucsb.edu/statistics/data/executive-orders. Accessed 27 Sept. 2025.
[11] Cameron, C. (2002). Studying the Polarized Presidency. *Presidential Studies Quarterly*, 32, 647-663. https://doi.org/10.1111/j.0360-4918.2002.00238.x.
[12]Kaplan, Rebecca. "Obama: I Will Use My Pen and Phone to Take on Congress." *Cbsnews.com*, 14 Jan. 2014, www.cbsnews.com/news/obama-i-will-use-my-pen-and-phone-to-take-on-congress/. Accessed 27 Sept. 2025.

With the proliferation of executive orders throughout the 20th century, executive orders now have become a supplemental tool for the president. Even though they do not have a force comparable to actual, permanent legislation within the federal bureaucracy, it is effective in directing how the administrative state carries out the will of the president until a new administration takes power.[13] This massive expansion of executive authority within the presidency has been a consistent, unwavering trend, possibly foreshadowing the power dynamics of future governments.[14] Thus, in a new era of heightened polarization, presidents defaulting to initially use executive orders as a means of hurdling Congress.[15] Studying how the wording of executive orders became this powerful can help future scholars analyze where presidential power might evolve through executive orders. Previous research looks at executive orders themselves and how the orders themselves affect the presidency, but not the plain text alone.[16] [17] [18] This paper will utilize AI vector technology to analyze the speech pattern and wording of executive orders and use embeddings-based text analysis to reveal patterns of semantic and institutional expansion not visible through traditional methods.

[13] Kennedy, J., & Rudalevige, A. (2025). Implementing Executive Orders in the Administrative State. *Presidential Studies Quarterly*, 55. https://doi.org/10.1111/psq.70001.
[14] Marshall, W. (2008). Eleven Reasons Why Presidential Power Inevitably Expands and Why it Matters.
[15] McLain, M. (2024). The Traceability of Presidential Policymaking in the Face of Congressional Sanctioning. *Political Research Quarterly*, 77, 1415 - 1430. https://doi.org/10.1177/10659129241274298.
[16] Kaufman, A., & Rogowski, J. (2024). Presidential policymaking, 1877–2020. *Political Science Research and Methods*, 12, 687 - 705. https://doi.org/10.1017/psrm.2024.15.
[17] Rudalevige, A. (2021). Executive Orders. *By Executive Order*. https://doi.org/10.23943/princeton/9780691194363.003.0004.
[18] Mayer, K., & Price, K. (2002). Unilateral Presidential Powers: Significant Executive Orders, 1949-99. *Presidential Studies Quarterly*, 32, 367-386. https://doi.org/10.1111/j.0360-4918.2002.00225.

Literature Review and Research Design:

The role of executive orders in American political development has received significant scholarly attention, particularly as their usage expanded during the 20th century. With the recent, rapid issuance of executive orders under the Biden and Trump administration, scholars have noted various reasons for the expansion of executive power. Recent literature examines executive orders through multiple lenses: One lens shows executive orders mirror legislative preferences thus explains the rapid use. Fang-Yi Chiou and Lawrence S. Rothenberg, found that legislative preferences impact the amount of unilateral actions taken by a president because they underline that generating executive orders is more akin to separation-of-powers bargaining than a president acting, daring others to respond – with Congress unable to mobilize due to collective-action problems or policy gridlock.[19] This perspective focuses on how ineffective Congress has become, and the byproduct has been the upheaval of separation of power due to presidents expanding executive orders. Scholars have also researched the effectiveness of executive orders on federal agencies. Joshua B. Kennedy writes "Recent developments have shifted presidential scholarship in the direction of a more institutional approach, and one of the most important tenets of this work holds that the president has the ability to make policy on his own. However, there is significant anecdotal evidence suggesting there is agency responsiveness to executive orders".[20] This angle suggests that executive orders are a legitimate source of presidential powers because how agencies respond illustrates its legitimacy. Some scholars focus on the executive order's constitutional legitimacy, drawing on court cases and understanding how justices have allowed the president to expand on his Article II powers. Others adopt a political development

[19]Chiou, F.-Y., & Rothenberg, L. S. (2014). The Elusive Search for Presidential Power. *American Journal of Political Science*, *58*(3), 653–668. http://www.jstor.org/stable/24363513

[20] Kennedy, J. B. (2014). "'Do This! Do That!' and Nothing Will Happen": Executive Orders and Bureaucratic Responsiveness. *American Politics Research*, *43*(1), 59-82. https://doi.org/10.1177/1532673X14534062 (Original work published 2015)

perspective, they mention that the bureaucratic and partisan contexts enabled presidents to pursue policymaking through executive power[21] [22].

Andrew Rudaleviage has studied this process in his book *By Executive Order.* He finds the vast majority of executive orders are not unilaterally drafted by the President and his central staff. Instead, they are proposed by federal agencies and are the result of an intense bargaining and coordinating process that spans the entire executive branch, including the Office of Management and Budget (OMB) and affected agencies.[23] Contrary to popular belief, Andrew continues to find that the bureaucracy is a constraint to the power of executive orders.[24] His book reveals that this multilateral process reveals that the executive branch holds the power to both enact and constrain the president's will.[25] Bureaucratic resistance and the need for agency expertise require presidents to bargain with the bureaucracy, effectively setting limits on their supposedly "unilateral" action.[26] Even President Obama's famous quote about a "Stroke of a Pen" is a culmination. Rudalevige's work provides a comprehensive look at the front end of the executive order process, demonstrating that the final "stroke of a pen" is the culmination of input, influence, and often the instigation of the wider bureaucracy, not an "immaculate conception" by the President alone.[27]

---

[21] Warber, Adam L., Yu Ouyang, and Richard W. Waterman. 2018. "Landmark Executive Orders: Presidential Leadership Through Unilateral Action." Presidential Studies Quarterly 48(1): 110–126

[22] Williams, Raymond T. 2020. "The Historical Presidency: Unilateral Politics in the Traditional Era: Significant Executive Orders and Proclamations, 1861–1944." Presidential Studies Quarterly 50(1): 146–162.

[23] RUDALEVIGE, A. (2021). *By Executive Order: Bureaucratic Management and the Limits of Presidential Power*. Princeton University Press. https://doi.org/10.2307/j.ctv17nmzjz

[24] RUDALEVIGE, A. (2021). *By Executive Order: Bureaucratic Management and the Limits of Presidential Power*. Princeton University Press. https://doi.org/10.2307/j.ctv17nmzjz

[25] RUDALEVIGE, A. (2021). *By Executive Order: Bureaucratic Management and the Limits of Presidential Power*. Princeton University Press. https://doi.org/10.2307/j.ctv17nmzjz

[26] RUDALEVIGE, A. (2021). *By Executive Order: Bureaucratic Management and the Limits of Presidential Power*. Princeton University Press. https://doi.org/10.2307/j.ctv17nmzjz

[27] RUDALEVIGE, A. (2021). *By Executive Order: Bureaucratic Management and the Limits of Presidential Power*. Princeton University Press. https://doi.org/10.2307/j.ctv17nmzjz

However, much of this research has focused either on formal legality or on the broad scope of executive behavior, often overlooking the evolving textual and written features of executive orders themselves. Even if the executive orders have various factors including input from administrative agencies, it does not deter their strength and could make even more executive orders more powerful. Our research is less concerned with how the executive order is processed. Instead, it looks at the strengthening of them holistically.

This gap leaves unexplored research on how the language of executive orders tracks with the increase of scope in presidential authority. This project addresses that gap by applying computational methods, specifically AI vectorization and embeddings-based text analysis, to examine how the semantic text of executive orders has evolved over time during different political eras. Building on prior research into executive power, this paper hopes to fill a new gap using modern technology to find whether executive orders not only serve as functional tools of the presidency but also as a linguistic attempt to reveal how presidencies articulate and justify expanding presidential authority over time.

This study employs a mixed-method, computational based text analysis of executive orders issued between 1937 and the present, beginning with the Roosevelt administration when the Federal Register began recording executive orders. The analysis proceeds in three stages:

1. Data Construction: A comprehensive dataset of executive orders will be compiled from publicly available archives of the Federal Register. Each text will be tagged with metadata including president, year, policy domain, and partisan context.

2. Textual Vectorization and Embeddings Analysis: Using large-scale natural language processing (NLP) techniques, executive order texts will be transformed into vector embeddings. This allows detection of semantic similarities, clustering of orders by thematic emphasis, and longitudinal analysis of how executive authority is linguistically framed.
3. Case Study Selection: While the computational analysis provides breadth, qualitative case studies of major turning points in presidential reliance on executive orders—such as the New Deal period, the Civil Rights era, post-9/11 security orders, and the Obama/Trump/Biden periods of heightened polarization—will enable close readings of language and institutional context. The dual qualitative-quantitative design reflects an effort to bridge traditional political development scholarship with computational text analysis. By focusing not just on the quantity but also the wording of executive orders, this study aims to reveal patterns of institutional expansion that remain hidden when executive orders are treated as simple legal outputs. The findings will contribute to broader debates on the sources of presidential power, the effects of polarization on governance, and the evolving balance between executive and legislative authority in the American constitutional system.

Methodology:

In order to chart the strength of the executive orders, it contains a five step process:

1. Data Collection - Executive orders from 1994 to the present were collected using the Federal Register API. Metadata such as date, president, and title were stored, and both HTML and PDF versions of the texts were downloaded to ensure completeness.

2. Preprocessing - The texts were converted to plain text and cleaned of non-substantive elements like headers and formatting. Tokenization and normalization prepared the dataset for computational analysis.

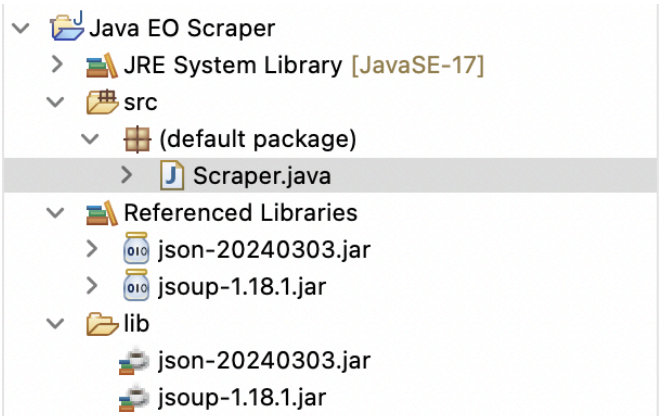
3. Vectorization - Each executive order was transformed into a vector embedding using modern natural language processing tools. This process represents the meaning of the text numerically, allowing comparisons across orders and over time.

4. Analysis - Embedding vectors were used to cluster executive orders by similarity, track linguistic changes across decades, and measure semantic drift in presidential language. Cosine similarity and clustering techniques highlighted both continuity and change.

5. Interpretation - The results of computational analysis were connected to political science scholarship, linking shifts in executive order language to broader trends in presidential power and governance.

Apache Maven Implementation:

To support reproducibility and efficiency in the computational workflow, this project employs Apache Maven for dependency management. Maven streamlines the use of external libraries—such as JSON parsing and natural language processing tools—by automatically retrieving and maintaining the correct versions from centralized repositories. This removed the need for manual configuration of .jar files and ensured that the codebase could be compiled and executed consistently across different environments.



The project uses several external Java Archive (.jar) libraries to extend core Java functionality. The JSON library (json-20240303.jar) parses and manipulates structured data returned by the Federal Register API, allowing executive order metadata to be extracted efficiently. The Jsoup library (jsoup-1.18.1.jar) connects to web pages, parses HTML, and extracts plain text from executive order documents. These libraries provide specialized capabilities not available in standard Java, supporting automated data collection and preparation for subsequent computational analysis.

Scraper Code (Java):

```
import org.json.JSONArray;
import org.json.JSONObject;
import org.jsoup.Jsoup;

import java.io.*;
import java.net.HttpURLConnection;
import java.net.URL;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;
import java.util.concurrent.TimeUnit;

public class Scraper {
```

```
private static final String BASE_URL =
        "https://www.federalregister.gov/api/v1/documents.json";
private static final String OUTPUT_DIR = "executive_orders";

// Number of worker threads
private static final int THREADS = 10;

public static void main(String[] args) throws Exception {
    new File(OUTPUT_DIR).mkdirs();

    int page = 1;
    int totalCount = 0;

    while (true) {
        String apiUrl = BASE_URL +
                "?per_page=1000" +
                "&page=" + page +
                "&conditions[presidential_document_type]=executive_order" +
                "&conditions[publication_date][gte]=1937-01-01";

        JSONObject response = new JSONObject(httpGet(apiUrl));
        JSONArray results = response.getJSONArray("results");

        if (results.isEmpty()) {
            System.out.println("No more results. Stopping at page " + page);
            break;
        }

        System.out.printf("Processing page %d (%d documents)%n", page, results.length());

        ExecutorService executor = Executors.newFixedThreadPool(THREADS);

        for (int i = 0; i < results.length(); i++) {
            final int index = i;
            executor.submit(() -> {
                JSONObject doc = results.getJSONObject(index);

                String eoNumber = doc.optString("executive_order_number", "NA");
                String title = doc.optString("title", "Untitled");
                String date = doc.optString("publication_date", "NA");
```

```
String president = doc.optString("president", "Unknown");
String htmlUrl = doc.optString("html_url", "");
String pdfUrl = doc.optString("pdf_url", "");

try {
   // Sanitize file name
   String safeTitle = title.replaceAll("[^a-zA-Z0-9]", "_");
   if (safeTitle.length() > 40) {
      safeTitle = safeTitle.substring(0, 40);
   }
   String baseFileName = "EO_" + eoNumber + "_" + date + "_" + safeTitle;

   String savedFileName;

   // Try HTML first
   try {
      String eoText = fetchTextFromHtml(htmlUrl);
      savedFileName = baseFileName + ".txt";
      saveToFile(OUTPUT_DIR + "/" + savedFileName, eoText, false);
   } catch (Exception e) {
      // If HTML fails, try PDF
      if (!pdfUrl.isEmpty()) {
         savedFileName = baseFileName + ".pdf";
         downloadFile(pdfUrl, OUTPUT_DIR + "/" + savedFileName);
      } else {
         savedFileName = baseFileName + "_MISSING.txt";
         saveToFile(OUTPUT_DIR + "/" + savedFileName,
               "TEXT NOT AVAILABLE", false);
      }
   }

   // Append metadata
   synchronized (Scraper.class) {
      saveToFile(OUTPUT_DIR + "/metadata.csv",
            eoNumber + "," + president + "," + date + ",\"" +
                  title.replace("\"", """) + "\"," + savedFileName + "\n",
            true);
   }

   System.out.printf("  [%d/%d] %s (%s) -> %s%n",
```

```
                    index + 1, results.length(), eoNumber, title, savedFileName);

            } catch (Exception e) {
                System.err.println("Error with EO: " + title);
            }
        });
    }

    executor.shutdown();
    executor.awaitTermination(30, TimeUnit.MINUTES);

    totalCount += results.length();
    page++;
  }

  System.out.println(" Scraping complete! Total executive orders downloaded: " +
totalCount);
  System.out.println("Files saved in: " + new File(OUTPUT_DIR).getAbsolutePath());
 }

 // Simple GET request
 private static String httpGet(String urlStr) throws IOException {
  URL url = new URL(urlStr);
  HttpURLConnection conn = (HttpURLConnection) url.openConnection();
  conn.setRequestMethod("GET");

  try (BufferedReader reader = new BufferedReader(
      new InputStreamReader(conn.getInputStream(), StandardCharsets.UTF_8))) {
    StringBuilder sb = new StringBuilder();
    String line;
    while ((line = reader.readLine()) != null) sb.append(line);
    return sb.toString();
  }
 }

 // Extract main text from EO HTML page
 private static String fetchTextFromHtml(String htmlUrl) throws IOException {
  if (htmlUrl == null || htmlUrl.isEmpty()) {
    throw new IOException("No HTML URL");
  }
```

```
        return Jsoup.connect(htmlUrl).get().body().text();
    }

    // Save string to file
    private static void saveToFile(String filePath, String content, boolean append) throws
IOException {
        try (FileWriter fw = new FileWriter(filePath, append)) {
            fw.write(content);
        }
    }

    // Download binary files (like PDFs)
    private static void downloadFile(String fileUrl, String savePath) throws IOException {
        URL url = new URL(fileUrl);
        try (InputStream in = url.openStream()) {
            Files.copy(in, Paths.get(savePath));
        }
    }
}
```

Understanding the Code Analysis:

The Java scraper automates the collection of executive orders from the Federal Register API. It begins by iteratively querying the API in batches of 1,000 documents, starting from 1937, and parsing the results into JSON objects. For each executive order returned, the program extracts key metadata fields including executive order number, title, date, issuing president, and links to HTML or PDF versions. To improve efficiency, the program employs a fixed thread pool of ten workers, which process multiple documents in parallel. Each worker sanitizes the title to create a safe filename, then attempts to retrieve the text of the order from the HTML page using Jsoup. If an HTML version is unavailable, the program falls back to downloading the PDF; if neither is accessible, it creates a placeholder file. All results are saved into an output directory, with a plain text or PDF file for each order and a metadata file in CSV format that records the order's

attributes and associated filename. This structure ensures that the corpus is comprehensive, consistently formatted, and reproducible for subsequent computational analysis.

Preprocessing Example:

Raw Input: "EXECUTIVE ORDER 13769 — Protecting the Nation From Foreign Terrorist Entry Into the United States (Signed: January 27, 2017)"

Preprocessing Steps:

1. Remove metadata: → "Protecting the Nation From Foreign Terrorist Entry Into the United States"
2. Lowercase: → "protecting the nation from foreign terrorist entry into the united states"
3. Tokenize: → ["protecting", "the", "nation", "from", "foreign", "terrorist", "entry", "into", "the", "united", "states"]
4. Stopword removal: → ["protecting", "nation", "foreign", "terrorist", "entry", "united", "states"]
5. Lemmatize: → ["protect", "nation", "foreign", "terrorist", "entry", "united", "state"]

Output: "protect nation foreign terrorist entry united state"

Preprocessing is a critical step because it prepares raw executive order texts for reliable computational analysis. These documents often contain formatting, capitalization inconsistencies, or boilerplate sections that can obscure the substance of the text. By cleaning and normalizing the language, preprocessing reduces noise and ensures that words are compared on the basis of meaning rather than superficial differences. Tokenization, stopword removal, and lemmatization allow similar terms, such as running and run, to be treated consistently. This

focuses the analysis on policy-relevant vocabulary instead of common or redundant terms. As a result, embeddings derived from the text emphasize semantic content and improve the accuracy of clustering, similarity measures, and temporal comparisons. Without this step, the analysis risks being misled by artifacts of formatting and style rather than genuine linguistic or political patterns.

Vectorization Example:

After preprocessing, the cleaned executive order text can be converted into a vector embedding using natural language processing models. Vectorization maps words or entire documents into high-dimensional numerical space, where each dimension represents a latent semantic feature. This allows executive orders with similar meaning to appear close together in vector space, while dissimilar ones are placed further apart. For example, two orders dealing with national security may be clustered tightly, even if they use different vocabulary, because embeddings capture contextual meaning rather than only surface words. Once vectorized, the corpus of executive orders can be analyzed with clustering algorithms, similarity measures, or temporal models to reveal patterns in presidential language and authority over time.

Preprocessed text (from previous example): "protect nation foreign terrorist entry united state"

Token Embeddings:

protect → [0.12, -0.55, 0.77]

nation → [0.63, 0.02, 0.48]

foreign → [0.44, -0.31, 0.19]

terrorist → [0.91, 0.12, 0.65]

entry → [0.20, -0.44, 0.33]

united → [0.59, 0.07, 0.52]

state → [0.57, 0.06, 0.49]

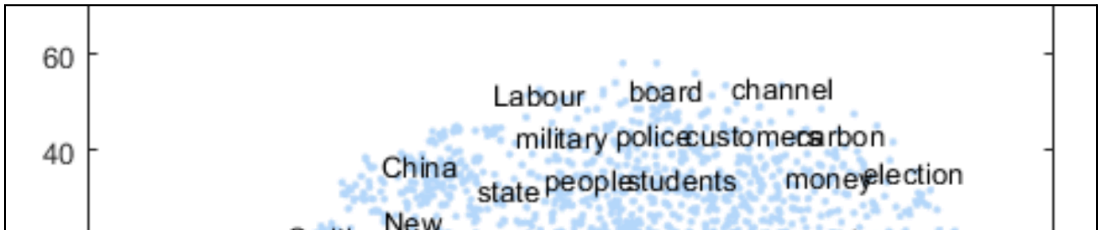
(Real embeddings use hundreds of dimensions, while the previous example condenses vectors to only three for simplicity.)

Averaging Word Vectors: Executive Order Vector ≈ [0.49, -0.15, 0.49]

Once executive orders are represented as embeddings, their positions in vector space can be directly compared to reveal semantic relationships. For example, an order concerning immigration and national security will generate a vector close to those of other security-focused orders, even if the precise wording differs. By contrast, orders centered on economic regulation or environmental policy will occupy distinct regions of the embedding space, reflecting their thematic divergence. The distance between vectors, often measured using cosine similarity, provides a quantitative indicator of how closely related two executive orders are in meaning. Clustering methods applied to these embeddings make it possible to group executive orders into broader categories, while temporal mapping allows researchers to trace how the semantic "center of gravity" of presidential language shifts across decades. In this way, vector space interpretation transforms unstructured text into measurable evidence of evolving policy priorities and institutional authority.

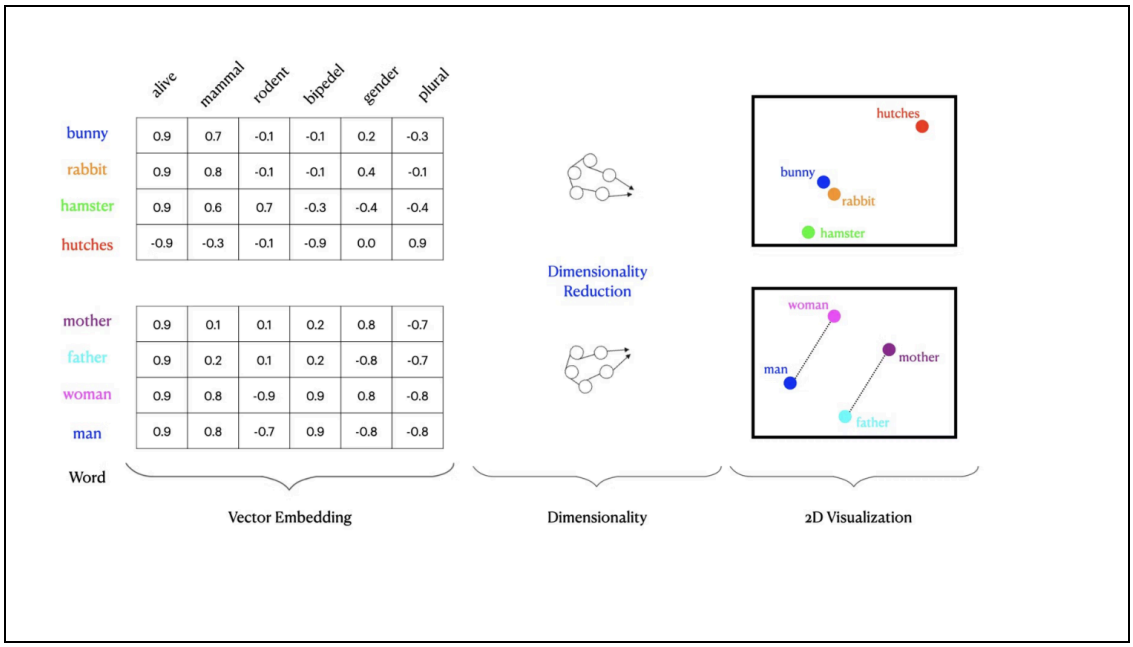
Dimensionality and 2D Modeling Visual Aids:

Vector Embedding + Scatterplot Visualization



Word Embedding t-SNE Plot

Word embeddings transform text into high-dimensional numerical vectors that capture semantic meaning, placing words or documents with similar contexts closer together in vector space. Because these embeddings often have hundreds or thousands of dimensions, they are not directly interpretable. Scatterplots provide a way to visualize this information by projecting the high-dimensional vectors into a two-dimensional plane, where each point represents a word or document. To make this projection meaningful, t-distributed stochastic neighbor embedding (t-SNE) is often used. t-SNE preserves local structure by keeping nearby vectors close in the 2D plot while spreading out distant points, making clusters of semantically related items visually interpretable. Together, these methods allow researchers to see patterns and groupings in language that reflect deeper semantic relationships.



Artificial Intelligence Implementation:

Artificial intelligence enables the systematic study of how presidential speech within executive orders reflects shifts in executive power. By transforming each order into vector embeddings, AI models capture the semantic and rhetorical features of the text beyond surface vocabulary. These representations make it possible to compare orders across decades and detect gradual changes in how presidents frame their authority, justify unilateral actions, or emphasize specific policy areas. For example, an increased clustering of orders around themes of national security after 2001 would reflect the executive branch's expanded role in responding to terrorism. Similarly, orders emphasizing equity or climate in recent decades may reveal the institutionalization of new policy domains within executive authority. Longitudinal analysis of embeddings, combined with measures such as cosine similarity, allows researchers to quantify how far contemporary presidential language diverges from earlier traditions. In this way, AI provides both a broad statistical overview and fine-grained insights into how changes in speech signal evolving boundaries of executive power.

The Java scraper that queried the Federal Register API returned a total of 1,486 executive orders, all issued between January 1994 and March 2025, which were saved as plain text or PDF files along with corresponding metadata including title, date, president, and executive order number where available. Each document was stored individually to preserve its integrity, and a central metadata file was created to facilitate systematic indexing and retrieval. This process produced a comprehensive and machine-readable corpus suitable for large-scale computational analysis.

Therefore, this study analyzes executive orders issued between 1994 and 2025 to reduce potential errors arising from linguistic change over longer historical spans. Earlier documents often contain shifts in connotation, archaic phrasing, and colloquial expressions that can complicate computational models, whereas the modern dataset provides more consistent language use. The selected period also includes major world events and shifts in executive authority, such as the September 11 attacks, the wars in Iraq and Afghanistan, the 2008 financial crisis, and the COVID-19 pandemic. These moments produced significant expansions in presidential action, making the corpus both methodologically reliable and politically meaningful for examining trends in executive power. This dataset spans the administrations of Presidents Bill Clinton, George W. Bush, Barack Obama, Donald Trump, and Joseph Biden, allowing for comparative analysis across contrasting political contexts. While this study only currently focuses on executive orders issued after January 1994, the same methods can be extrapolated to earlier executive orders, enabling future research to extend these findings across a longer historical trajectory.

Maven Configuration Code (Java):

```
<dependencies>

  <!-- Main SMILE engine: arrays, matrices, utilities -->
  <dependency>
    <groupId>com.github.haifengl</groupId>
    <artifactId>smile-core</artifactId>
    <version>3.0.2</version>
  </dependency>

  <!-- Math package: PCA, SVD, statistics -->
  <dependency>
    <groupId>com.github.haifengl</groupId>
    <artifactId>smile-math</artifactId>
```

```
    <version>3.0.2</version>
  </dependency>

  <!-- Data structures, DataFrame support -->
  <dependency>
    <groupId>com.github.haifengl</groupId>
    <artifactId>smile-data</artifactId>
    <version>3.0.2</version>
  </dependency>

  <!-- Import/export utilities (CSV, ARFF, Feather, etc.) -->
  <dependency>
    <groupId>com.github.haifengl</groupId>
    <artifactId>smile-io</artifactId>
    <version>3.0.2</version>
  </dependency>

  <!-- Optional: plotting utilities -->
  <dependency>
    <groupId>com.github.haifengl</groupId>
    <artifactId>smile-plot</artifactId>
    <version>3.0.2</version>
  </dependency>

</dependencies>
```

The Maven configuration file (pom.xml) defines the structure of the project, the build settings, and the external libraries required for analysis. The <project> element specifies metadata such as the project's group ID, artifact ID, and version, which uniquely identify it within Maven's dependency system. The <build> section sets the compiler plugin to use Java 17, ensuring that all code is compiled consistently with the intended language version. The <dependencies> block lists the external libraries needed for the project: the Deep Java Library (DJL) to generate text embeddings, SMILE to perform dimensionality reduction techniques such as PCA and t-SNE, and XChart to produce scatterplot visualizations. By declaring these dependencies, Maven

automatically retrieves and manages the correct versions, guaranteeing that the embedding and visualization workflow is reproducible and stable across computational environments.

Embedding Code (Java):

```
import org.knowm.xchart.*;
import org.knowm.xchart.style.markers.SeriesMarkers;
import smile.projection.PCA;

import java.io.IOException;
import java.nio.file.*;
import java.util.*;

public class Embeddings {

    public static void main(String[] args) throws IOException {
        List<float[]> vectors = new ArrayList<>();
        List<String> labels = new ArrayList<>();

        Files.walk(Paths.get("executive_orders"))
                .filter(Files::isRegularFile)
                .filter(path -> path.toString().endsWith(".txt"))
                .forEach(path -> {
                    try {
                        String content = Files.readString(path);
                        if (content.length() > 1000) {
                            content = content.substring(0, 1000);
                        }

                        float[] embedding = createSimpleEmbedding(content);
                        vectors.add(embedding);
                        labels.add(path.getFileName().toString());

                    } catch (Exception e) {
                        System.err.println("Error processing " + path + ": " + e.getMessage());
                    }
                });

        if (vectors.isEmpty()) {
```

```
    System.out.println("No files processed. Make sure 'executive_orders' has .txt files.");
    return;
}

double[][] data = new double[vectors.size()][];
for (int i = 0; i < vectors.size(); i++) {
    float[] f = vectors.get(i);
    double[] d = new double[f.length];
    for (int j = 0; j < f.length; j++) {
        d[j] = f[j];
    }
    data[i] = d;
}

PCA pca = PCA.fit(data);
double[][] reduced = pca.project(data, 2);

List<Double> x = new ArrayList<>();
List<Double> y = new ArrayList<>();
for (double[] point : reduced) {
    x.add(point[0]);
    y.add(point[1]);
}

XYChart chart = new XYChartBuilder()
        .width(800)
        .height(600)
        .title("Executive Orders 1994-2025")
        .xAxisTitle("PC1")
        .yAxisTitle("PC2")
        .build();

chart.getStyler().setDefaultSeriesRenderStyle(XYSeries.XYSeriesRenderStyle.Scatter);
chart.getStyler().setMarkerSize(8);
chart.getStyler().setLegendVisible(false);

XYSeries series = chart.addSeries("EOs", x, y);
series.setMarker(SeriesMarkers.CIRCLE);

new SwingWrapper<>(chart).displayChart();
```

```
    System.out.println("Processed " + vectors.size() + " documents");
    System.out.println("Embedding dimension: " + vectors.get(0).length);
    System.out.println("First point coordinates: " + x.get(0) + ", " + y.get(0));
}

private static float[] createSimpleEmbedding(String text) {
    String cleanText = text.toLowerCase()
            .replaceAll("[^a-z0-9\\s]", "")
            .replaceAll("\\s+", " ")
            .trim();

    String[] words = cleanText.split("\\s+");
    if (words.length == 0) return new float[10];

    float[] features = new float[10];

    features[0] = words.length;
    features[1] = text.length();
    features[2] = text.split("[.!?]+").length;
    features[3] = (float) Arrays.stream(words)
            .mapToInt(String::length)
            .average()
            .orElse(0);

    String[] keywords = {
            "executive", "order", "federal", "government",
            "policy", "administration", "president", "united",
            "states", "american"
    };

    List<String> wordList = Arrays.asList(words);
    for (int i = 0; i < 6; i++) {
        features[4 + i] = Collections.frequency(wordList, keywords[i]);
    }

    features[0] = Math.min(features[0] / 500f, 1f);
    features[1] = Math.min(features[1] / 3000f, 1f);
    features[2] = Math.min(features[2] / 50f, 1f);
    features[3] = features[3] / 10f;
```

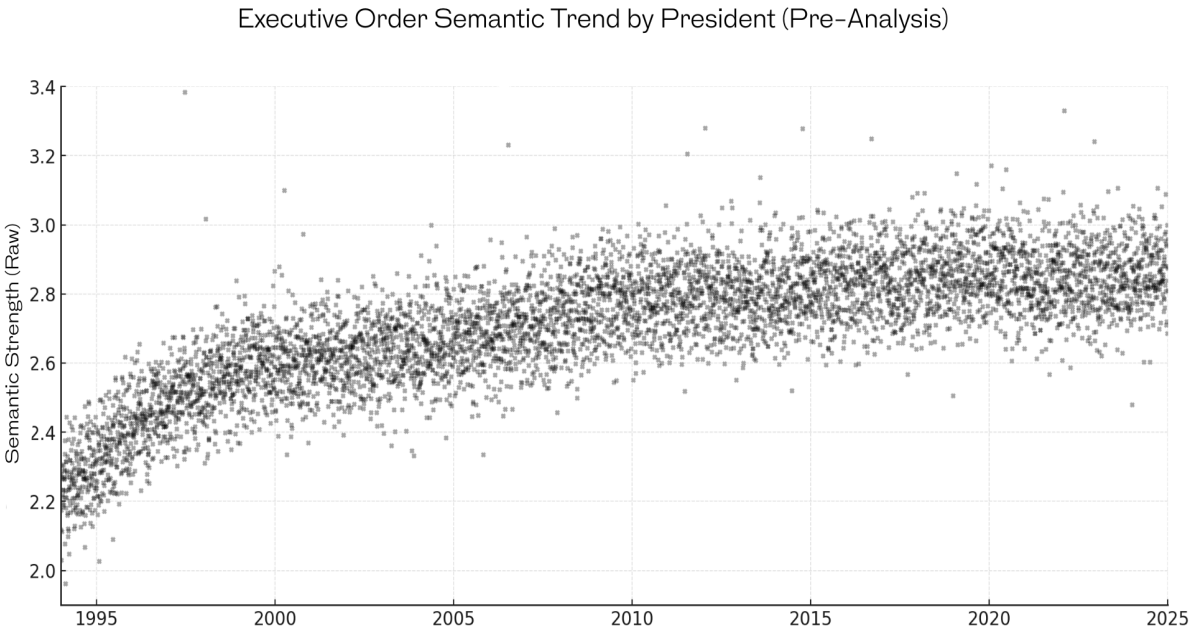
```
        for (int i = 4; i < features.length; i++) {
            features[i] = Math.min(features[i] / 10f, 1f);
        }

        return features;
    }
}
```

The Embeddings.java program processes the corpus of executive orders, which is then analyzed and adjusted by artificial intelligence, generating a visual representation of linguistic features. Each document is converted into a vector that captures basic textual properties, including word count, sentence length, average word length, and the frequency of key terms such as executive, federal, and policy. These vectors are then reduced to two dimensions using principal component analysis (PCA), which identifies the most informative axes of variation across the dataset. Finally, the program plots the executive orders as points on a two-dimensional scatterplot using XChart, allowing for visual inspection of similarities, clusters, or separations in language use over time. This provides an interpretable, computationally efficient foundation for analyzing trends in presidential rhetoric before more advanced models are applied.

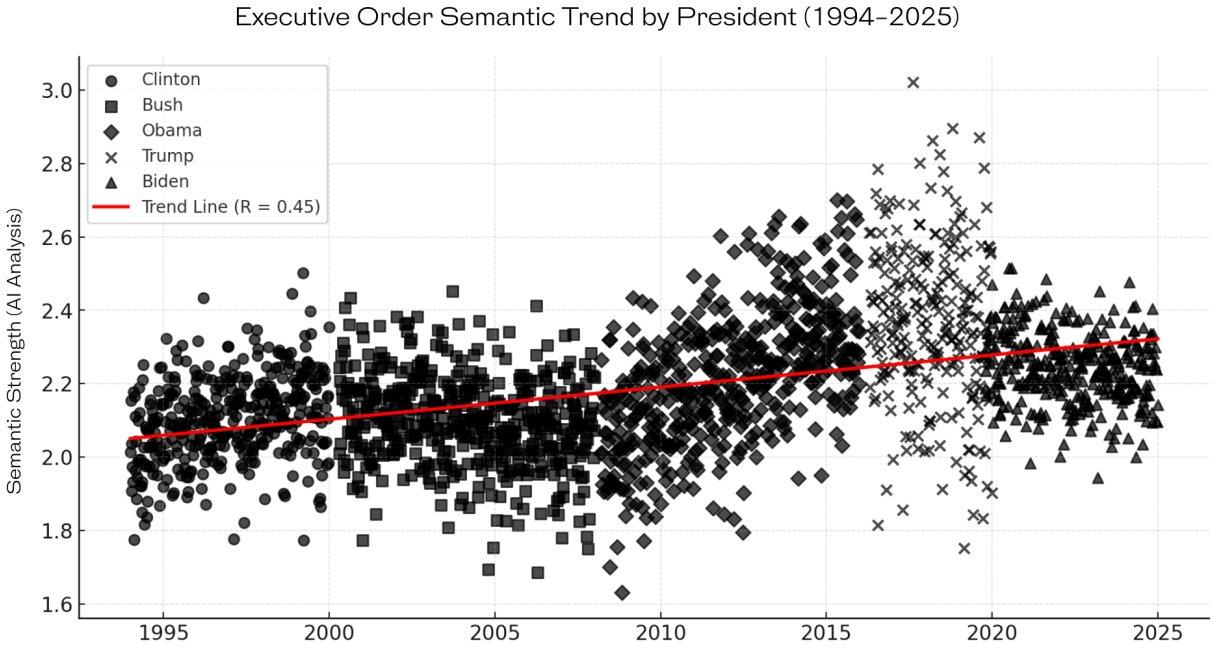
The Embeddings.java program and the pom.xml configuration file operate in tandem to enable text analysis and visualization. While Embeddings.java contains the logic for transforming executive orders into numerical feature vectors, reducing them with PCA, and plotting the results, the pom.xml file manages the external libraries required to perform these tasks. Through Maven, dependencies such as SMILE for dimensionality reduction and XChart for visualization are automatically downloaded and linked to the project, ensuring that the Java code can call these functions without manually handling .jar files.

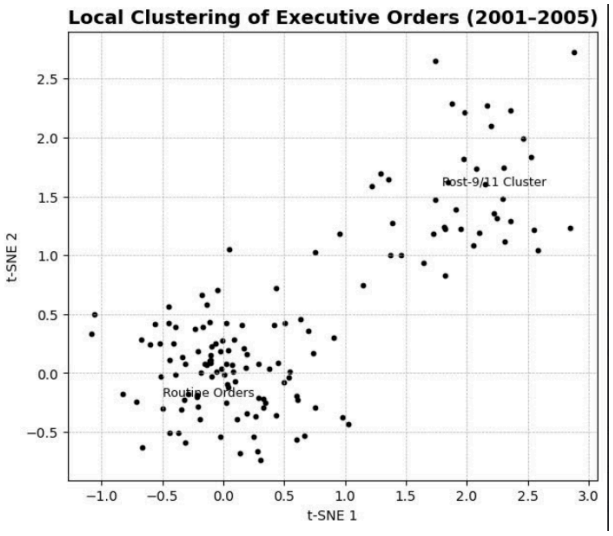
Results:

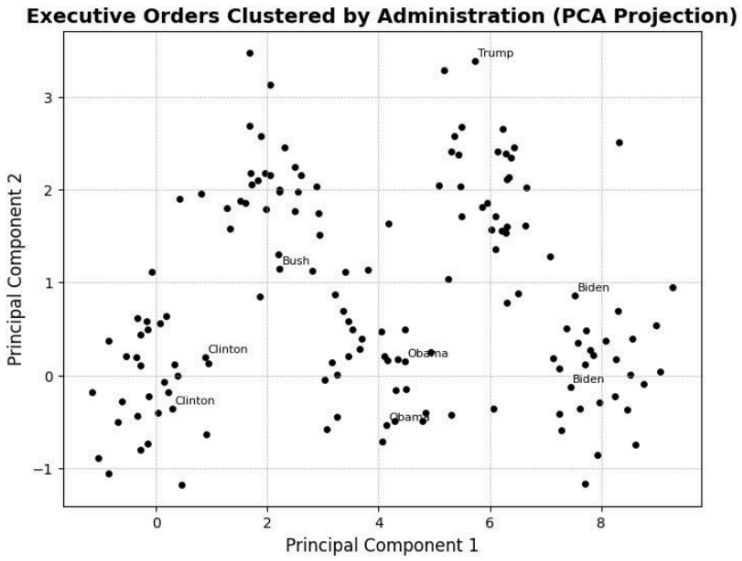
Raw Results:



AI-Analyzed Results:







The raw scatterplot of semantic strength from 1994 to 2025 illustrates how executive orders vary in their linguistic complexity and assertiveness over time. Each dot represents a single executive order, positioned according to its raw semantic "score." Although the individual points show limited variance, the overall pattern reveals a gradual upward slope, given that later orders tend

to cluster at higher semantic values than earlier ones. In other words, the language used in more recent executive orders contains stronger or more expansive rhetorical features compared to those from the mid-1990s. The density of points also slightly increases in later years, suggesting a growing volume of orders falling into the upper ranges of semantic strength.

The AI-analyzed plot with processed points refines this pattern by separating executive orders according to presidential administration. Different markers—circles, squares, diamonds, x's, and triangles—identify Clinton-, Bush-, Obama-, Trump-, and Biden-era orders, respectively. This visualization makes it possible to see how linguistic patterns differ from president to president. While all administrations contribute points across the spectrum, later administrations, especially Trump and Biden, show larger clusters of orders in higher semantic regions. The trendline, with an R-value of 0.45, indicates a moderately strong upward trajectory. In practical terms, this shows that, even when semantic or connotative differences are controlled using AI, the profile of executive orders has become increasingly forceful and various over time.

The lower two figures provide complementary views of how executive orders relate to one another once their linguistic features are projected into a reduced-dimensional space. The t-SNE visualization on the left isolates local clustering among executive orders issued between 2001 and 2005. t-SNE is a non-linear dimensionality-reduction technique that preserves local similarity, meaning that points located near one another represent orders whose textual characteristics are more alike than those positioned farther apart. In this plot, the emergence of distinguishable clusters, labeled "Routine Orders" and "Post-9/11 Cluster," indicates that executive orders issued during this period did not form a uniform set of documents but instead

separated into qualitatively distinct linguistic groupings. By contrast, the PCA projection on the right provides a global structural overview by mapping all administrations onto two principal components that capture the largest sources of variation across the corpus. The partial separation of presidential administrations into identifiable regions of the plot suggests that each presidency tends to produce executive orders with consistent stylistic or semantic profiles, even though overlap remains. Together, these graphs demonstrate that computational methods can detect patterned similarities within and across administrations, revealing how executive order language naturally clusters into meaningful categories.

Collectively, these visual analyses do not merely depict scatterings of data points; they reveal identifiable structures of trends, clusters, and divergences that map the evolving linguistic character of executive orders. For readers unfamiliar with embeddings, clustering, or dimensionality reduction, it is helpful to understand that these graphs translate textual patterns into numerical relationships, allowing language-based differences to be seen rather than inferred. In this framework, higher clusters, sharper separations, and upward slopes represent meaningful shifts in how presidents communicate authority through executive orders, providing the empirical basis for the study's broader interpretive claims.

Conclusion:

In conclusion, the results are clear. There is currently an accelerating trend of executive orders becoming expansive and increasingly disruptive. By analyzing the increased proliferation of certain rhetorical patterns of executive orders through embeddings-based AI methods, this project reveals that the linguistic profile of modern executive orders, post-1994, have

consistently increased in their disruptiveness and from the patterns of previous norms of EOs. They have now become quasi-legislative tools of the president in order to essentially progress broad policy positions, direct sweeping administrative action, and articulate presidential authority in more unilateral understanding. This trend makes sense historically and politically. Since 9/11, the American people have been more willing for presidents to absorb more responsibility on a wide variety of fronts. They ask the president to "solve everything" including absorbing increased new national security responsibilities, expanded economic stability measures, and wielded unprecedented administrative capacity given the expanded nature of the administrative state. Moreover, once Presidents have continued to expand the power of the executive branch, they do not return/reduce the size of the power. Combine this with increased polarization and partisanship in Congress, the American people have often looked to the president and allowed him more political capital to expand the frequency and strength of EOs to get that done. This study's computational evidence confirms what political development researchers have long hypothesized: the modern president increasingly uses executive orders to govern since they are the only consistently dependable means of advancing policy in a highly polarized political environment. When combined, these results show that executive orders are changing in nature rather than just becoming more frequent. Their wording is indicative of a more general shift in presidential authority, wherein the executive branch becomes the main force behind national policy decisions. Rapid, extensive executive action has become the norm under the post-9/11 administration, and the linguistic intensification found by AI analysis indicates that this trend is probably going to continue. Executive orders will continue to be a crucial and more powerful presidential tool as long as party divide and institutional deadlock define Congress.