Using CATA and Machine Learning to Operationalize Old Constructs in New Ways: An Illustration Using U.S. Governors’ COVID-19 Press Briefings

Jason D. Marshall1, Francis J. Yammarino2, Srikanth Parameswaran2, and Minyoung Cheong3

Abstract
Increased computing power and greater access to online data have led to rapid growth in the use of computer-aided text analysis (CATA) and machine learning methods. Using “big data”, researchers have not only advanced new streams of research, but also new research methodologies. Noting this trend and simultaneously recognizing the value of traditional research methods, we lay out a methodology that bridges the gap between old and new approaches to operationalize old constructs in new ways. With a combination of web scraping, CATA, and supervised machine learning, using labeled ground truth data (i.e., data with known inputs and outputs), we train a model to predict CIP (Charismatic-Ideological-Pragmatic) leadership styles from running text. To illustrate this method, we apply the model to classify U.S. state governors’ COVID-19 press briefings according to their CIP leadership style. In addition, we demonstrate content and convergent validity of the method.

Keywords
text analysis, CATA, web scraping, machine learning, leadership, COVID-19

1Heider College of Business, Creighton University, USA
2School of Management, Binghamton University, State University of New York, USA
3School of Graduate Professional Studies at Great Valley, Pennsylvania State University, USA

Current Affiliation: Minyoung Cheong is currently affiliated with School of Management, Kyung Hee University, Seoul, South Korea.

Corresponding Author:
Jason D. Marshall, Heider College of Business, Creighton University, 602 N 20th Street, Omaha, NE 68102, USA.
Email: jmmarshall6112@gmail.com
As the internet is crawling with data (pun intended), researchers have begun to leverage this “big data” source (Braun, Kuljanin, & DeShon, 2018) to advance theory in a variety of research contexts. Advanced computing capabilities and modern analytic techniques have made this possible. For example, with web scraping techniques, researchers can access data directly from a web page to be analyzed on their personal computer. In addition to numerical data, there is a plethora of textual data available on the internet. Accessing, extracting, and storing this data is as easy as executing a few lines of code in your favorite programming language (e.g., Python or R). This ease of access has corresponded with an increase in text analysis studies in recent years (Banks, Woznyj, Wesslen, & Ross, 2018). Specifically, computer-aided text analysis (CATA; McKenny, Aguinis, Short, & Anglin, 2018; McKenny, Short, & Payne, 2012; Short, Broberg, Cogliser, & Brigham, 2010) has grown more popular.

In brief, CATA provides rich insights into individual cognitions, values, and identities in ways that cannot be replicated via traditional research methods, such as self-report surveys (Pollach, 2012). CATA research typically uses deductive dictionary-based approaches to text analysis, which differs from open-language methods such as text mining (Kobayashi, Mol, Berkers, Kismihók, & Den Hartog, 2018), natural language processing (NLP; Pandey & Pandey, 2019), and neural network models (DeTienne, DeTienne, & Joshi, 2003; Minbashian, Bright, & Bird, 2010) that are also gaining traction. Whereas these open-language approaches to text analysis leverage machine learning algorithms and co-occurrence matrices to establish relationships among words in a bottom-up fashion, CATA predominantly takes a top-down approach, calculating word frequency within established categories or dictionaries. CATA dictionaries are comprised of words that are linked together in meaning, often theoretically developed, and psychometrically validated by subject matter experts (Short et al., 2010). Hence, with CATA’s psychometrically validated dictionaries, there is an untapped opportunity to utilize machine learning techniques, in conjunction with CATA, to gain some methodological advantages over open-language techniques.

There are two basic forms of machine learning: supervised and unsupervised. In unsupervised machine learning, the machine is not aware of any preexisting categories, dictionaries, or classifications of words, but merely uses a form of cluster analysis, based on co-occurrence matrices, to establish which groups of words cluster together in a corpus (i.e., a body of text). Text mining is an example of unsupervised machine learning applied to text analysis. Supervised machine learning, on the other hand, builds predictive models by learning from training examples where feature data is labeled according to its ground truth output (Zhou, 2018). Ground truth, in this case, refers to data that were human-coded or labeled in a prior research study, such as the CIP (Charismatic – Ideological – Pragmatic) labels assigned to U.S. Presidents in Yammarino and colleagues’ (2013) historiometric study. In other words, the researcher provides both input (independent) and output (dependent) variables, which allows the machine to learn how to best fit a model to match the two (Janasik, Honkela, & Bruun, 2009). We assert that supervised machine learning provides an avenue to leverage and extend existing ground truth output data to operationalize constructs in new ways.

With all the new advanced data analysis tools and techniques, and the relatively easy access to big data, one might be compelled to forget all about traditional research methods and move on to the shiny new toys. However, we offer an alternative perspective. Rather than take an “out with the old and in with the new” approach, we believe a better method is to leverage existing ground truth datasets (i.e., the old) and apply web scraping, CATA, and supervised machine learning techniques (i.e., the new) to advance new streams of research. This combination of techniques is an important methodological contribution to both micro- and macro-oriented research because it leverages and extends existing methodologies to allow for construct operationalization via text analysis. For example, the application of these techniques, in concert, would be particularly useful in upper echelons research (Hambrick & Mason, 1984) where traditional research methodologies
(e.g., self-report surveys, interviews) are more difficult to employ (Hambrick, 2007). As such, in this context, our operationalization could serve as a useful tool to test and/or develop theory in upper echelons research.

Of course, not all public-facing individuals occupy upper echelon positions in organizations. Fortunately, this approach could just as easily be applied at the individual level of analysis for movie stars, professional athletes, and social media influencers. Additionally, as we highlight throughout the manuscript, publicly available data and web scraping are not prerequisites for the execution of the CATA and machine learning portions of the method. As such, the method could be easily applied in conjunction with more traditional forms of research at the individual or group level. In other words, our recommended methodology could be applied at multiple levels of analysis and in a variety of contexts.

Another strength of this method, beyond convenience, is that it allows for a more nuanced examination of psychometric data over time. Aguinis and Bakker (2021) defined four aspects of time (duration, frequency, timing, and sequence) that can and should be specified when discussing time as an aspect of a construct. By operationalizing constructs through running text (e.g., speeches, interviews, tweets), the method we demonstrate in this manuscript allows researchers to capture each of these aspects of time. In other words, as running text often spans across time and contexts, this methodological combination makes it possible for researchers to examine underlying theoretical assumptions (e.g., stability, causal sequence) that were previously untestable or impractical to test. To demonstrate the utility of our recommended method, the current study illustrates how to use a combination of web scraping, CATA, and supervised machine learning techniques to classify categorical outcomes in an upper echelons context.

**CIP Style in U.S. Governors’ COVID-19 Press Briefings**

**Current Study**

The particular categorical outcome we have chosen to model is the CIP (Charismatic-Ideological-Pragmatic) outstanding leadership style approaches (Lovelace, Neely, Allen, & Hunter, 2019; Mumford, 2006) of U.S. state governors. With the current COVID-19 pandemic, the effects of leadership (or lack thereof) have been magnified. As Bligh and colleagues noted, “People become increasingly susceptible to the leader and his or her vision in the wake of a crisis” (Bligh, Kohles, & Meindl, 2004: 215). The pandemic is certainly a crisis and leaders (e.g., U.S. state governors) have been frequently communicating their visions, beliefs, and solutions via press briefings. Accordingly, in this study, we seek to categorize the U.S. state governors’ press briefings according to their CIP leadership style in hopes of not only advancing a new methodology, but also providing data analysis that may be of interest/value to the general public. The latter aspiration is a direct response to a call from Aguinis and colleagues (Aguinis, Suarez-Gonzalez, Lannelongue, & Joo, 2021) to reach an audience outside of academia.

**CIP Model**

Mumford (2006) developed the CIP model of outstanding leadership based on early work from Max Weber (1924). According to the model, there are three different leadership styles that are all equally likely to produce outstanding leadership, and one key distinction among the styles is the time-orientation of the leader (Mumford, 2006): Charismatic leaders (i.e., the C in CIP) are future-focused, ideological leaders (i.e., the I in CIP) are past-focused, and pragmatic leaders (i.e., the P in CIP) are present-focused. Not only does time-orientation play a key role in differentiating among CIP leadership styles, but for charisma specifically, it also distinguishes Mumford’s (2006) conception of the
construct from the more mainstream definitions of charisma (e.g., Antonakis, Bastardoz, Jacquart, & Shamir, 2016; House & Howell, 1992; Shamir, Arthur, & House, 1994). In addition to time orientation, several other factors are used to differentiate among CIP styles. For example, charismatic leaders use positive emotional imagery and are concerned with motivating followers. Regarding problem solving, charismatics are best during the middle problem-solving phase when their ability to generate possibilities is most valuable (Lovelace et al., 2019). Ideological leaders are values-driven and use negative emotion in their communication (Hunter, Cushenbery, Thoroughgood, Johnson, & Ligon, 2011). During times of crisis, ideologues perceive situations as the causal mechanism, focus on changing the system, and may rally their base constituency (i.e., like-minded individuals), rather than attempt to appeal to the masses (Griffith et al., 2018). With regard to their problem-solving value, ideologues tend to be best during the late problem-solving phase when their ability to implement solutions is most needed (Lovelace et al., 2019). Finally, pragmatic leaders communicate using rational appeals and are best during the early phase of problem-solving when data are being gathered and evaluated. During times of crisis, pragmatic leaders adopt an interactionist approach, focusing on how the situation affects people and their behavior (Griffith et al., 2018; Lovelace et al., 2019).

**Limits of Historiometric Method and Opportunity for New Methods**

Prior CIP studies have primarily used historiometric methods (Ligon, Harris, & Hunter, 2012) to assess the CIP style in outstanding historical leaders such as Presidents (Yammarino, Mumford, Serban, & Shirreffs, 2013), world leaders (Serban et al., 2018), and college/NFL football coaches (Hunter et al., 2011). In other words, these researchers have used historical documents, such as biographies, to code whether an individual is C, I, or P. Recently, Lovelace and colleagues (2019) stated that research on the CIP model has been over-reliant on the historiometric approach and claimed this reliance may be limiting the potential applications of the model. Although historiometric methods have a few limitations for the study of CIP in organizations, the biggest limitation is that historiometric methods are labor-intensive to employ (Lovelace et al., 2019). Researchers spend dozens of hours learning how to code the documents. Then, they spend countless hours reading over and manually coding documents. Then, the documents are checked for interrater reliability. All of this takes a tremendous amount of time and resources.

Thus, to address this limitation of the historiometric approach, upon which the CIP model has heavily relied, we advance a more granular, flexible, and timely operationalization of the CIP model that combines web scraping, computer-aided text analysis (CATA), and supervised machine learning techniques. Web scraping opens the door to numerous textual datasets, systematically displayed on publicly available websites. CATA not only allows researchers to code large volumes of text in a fraction of the time, but it also allows us to study running text (e.g., speeches, interviews, meeting transcripts, tweets) in current leaders, without waiting for biographies to be written after the fact. Finally, supervised machine learning, when used in conjunction with a ground truth dataset, can be used to train a model to predict outcome variables (e.g., CIP style in our case). Collectively, these methodologies provide a new operationalization for CIP research.

**New Methods in Combination**

To illustrate the operationalization of our method, we leverage web scraping, CATA, and supervised machine learning techniques to classify CIP style of U.S. state governors. To help guide the reader through the process and enable other researchers to execute this method for their research topics of interest, whether micro, meso, or macro in orientation, we provide the full step-by-step process, including recommended R and Python packages, in Table 1.
Sample and Selection

As we use multiple datasets from multiple sources, in Figure 1 we provide sample data configurations to help distinguish among the datasets and provide a visual overview of the supervised machine learning process.

Ground Truth Data. To build a model using supervised machine learning, we needed a dataset with labels assigned to the outcome variables. In supervised machine learning, this labeled data is often referred to as ground truth data. For the present study, our ground truth labels were derived from Yammarino and colleagues’ (2013) historiometric study of United States Presidents, in which each President was classified as either C, I, or P. We used Yammarino et al.’s (2013) classifications to build our CIP ground truth dataset for all U.S. Presidents prior to Donald Trump.

Training and Validation Data. Through web scraping techniques, we collected Presidential address transcripts from “The American Presidency Project”, hosted by the University of California (UC).

| Step | Action | Recommended Package/Program |
|------|--------|-----------------------------|
| 1    | Identify ground truth labels of interest (e.g., CIP style of U.S. Presidents, Yammarino et al., 2013) |   |
| 2    | Identify textual data source to generate features for individuals in the ground truth dataset (e.g., website) |   |
| 3    | Scrape textual data to generate features for a training dataset |   |
| 4    | Perform computer-aided text analysis (CATA) on the textual data to create numeric features | BeautifulSoup (Python) rvest (R) LIWC |
| 5    | Merge LIWC output with ground truth labels to build a dataset for supervised machine learning | merge (R) |
| 6    | Compare supervised machine learning algorithms | caret (R) |
|      | ▪ Certain algorithms are designed for categorical outcome variables and others for continuous outcome variables |   |
|      | ▪ Simply change the “model” in the caret script to run various algorithms |   |
|      | ▪ Compare model accuracy and kappa among algorithms |   |
| 7    | Model feature selection | caret::varImp (R) caret::rfe (R) |
|      | ▪ Let theory and statistics guide the feature selection |   |
|      | ▪ Depending on the algorithm, there are different ways to identify the important features, within the caret package |   |
|      | ▪ For MLR, we used the varImp function |   |
|      | ▪ For other algorithms (e.g., RandomForests), the rfe function in the caret package can be used |   |
| 8    | Train and validate a supervised machine learning model | caret::trainControl (R) |
|      | ▪ Various cross-validation methods are available in the caret package (e.g., k-folds, repeated k-folds, bootstrap, bootstrap .632, leave-one-out) |   |
| 9    | Scrape textual data for model application (e.g., CIP style in U.S. state governors) | BeautifulSoup (Python) rvest (R) |
| 10   | Perform CATA on the textual data you wish to classify | LIWC |
| 11   | Apply the model to predict outcomes in new data | predict (R) |
| 12   | Model validation (if possible) via SMEs |   |

Note. LIWC stands for Linguistic Inquiry and Word Count (Pennebaker et al., 2015).
Santabara at https://www.presidency.ucsb.edu/index.php, to use as the raw data for model training and validation. The addresses included inaugural addresses, state of the union addresses, commencement addresses, remarks at the White House correspondents’ dinner, holiday addresses, remarks to the United Nations, remarks to U.S. Congress, remarks to foreign governments, and remarks on foreign affairs. The addresses spanned from George Washington, in 1789, through Barack Obama, in 2016. Data were also collected from Donald Trump; however, they were not used in this dataset because he was not included in the Yammarino et al. (2013) CIP classifications.

As each President was classified by a single CIP style, in our ground truth data (Yammarino et al., 2013), we applied that same style to each Presidential address for a given President. For example, Barack Obama was classified as P, therefore, each of his Presidential addresses was labeled as P. In total, we collected 844 transcripts of U.S. Presidential addresses, which we analyzed using Linguistic Inquiry and Word Count (LIWC; Pennebaker et al., 2015) and merged with the labeled ground truth CIP labels from Yammarino et al. (2013). After accounting for language change over time, which we systematically assessed (see Appendix A), our final training and validation sample consisted of 377 Presidential addresses that occurred after 1980.

**Figure 1. Data sources and examples.**

*Note. CIP stands for charismatic, ideological, and pragmatic. LIWC stands for Linguistic Inquiry and Word Count (Pennebaker et al., 2015). Ground truth data were collected from Yammarino et al., 2013. Training-testing data were collected from “The American Presidency Project,” hosted by UC Santa Barbara (844 Presidential addresses total). Application data were collected from Rev.com (835 COVID-19 press briefings total).*

Santa Barbara at https://www.presidency.ucsb.edu/index.php, to use as the raw data for model training and validation. The addresses included inaugural addresses, state of the union addresses, commencement addresses, remarks at the White House correspondents’ dinner, holiday addresses, remarks to the United Nations, remarks to U.S. Congress, remarks to foreign governments, and remarks on foreign affairs. The addresses spanned from George Washington, in 1789, through Barack Obama, in 2016. Data were also collected from Donald Trump; however, they were not used in this dataset because he was not included in the Yammarino et al. (2013) CIP classifications. As each President was classified by a single CIP style, in our ground truth data (Yammarino et al., 2013), we applied that same style to each Presidential address for a given President. For example, Barack Obama was classified as P, therefore, each of his Presidential addresses was labeled as P. In total, we collected 844 transcripts of U.S. Presidential addresses, which we analyzed using Linguistic Inquiry and Word Count (LIWC; Pennebaker, Booth, Boyd, & Francis, 2015) and merged with the labeled ground truth CIP labels from Yammarino et al. (2013). After accounting for language change over time, which we systematically assessed (see Appendix A), our final training and validation sample consisted of 377 Presidential addresses that occurred after 1980.

**Application Data.** After a model has been trained and validated, it can be applied to predict outcomes in new data. Accordingly, our model was applied to a dataset containing 835 COVID-19 press briefing transcripts, representing 42 United States governors. Through web scraping techniques, we collected the press briefing transcripts from https://www.rev.com/blog/transcript-tag/coronavirus-update-transcripts. The press briefings spanned from February 27, 2020 to April 23, 2021. Regarding political affiliation, 549 of the briefings were from Democratic governors (65.74%) and
286 (34.25%) were from Republican governors. Females delivered 88 of the briefings (10.54%) and males delivered the other 747 (89.46%).

**Web Scraping**

Web scraping refers to the automated process of obtaining data directly from a webpage. While some companies develop specific APIs (Application Programming Interfaces) that allow researchers to quickly and easily access data (Braun et al., 2018), many times APIs are not available. For websites without an API, Kobayashi and colleagues recommend using web scraping techniques to acquire the data (Kobayashi et al., 2018). Prior to beginning the web scraping process, it is important to consult the terms of use for the website from which you wish to acquire the data. For example, the terms of use for the Rev website, where we scraped the governors’ press briefings, stated restrictions such that a user may not, “take any action that imposes an unreasonable or disproportionately heavy load on the Platform or its infrastructure or that negatively affects the ability of others to access or use the Platform” (Rev, 2020). This does not explicitly forbid web scraping per se, but rather necessitates the use of responsible web scraping practices, which we will describe in detail below. In addition to reviewing the terms of use, in this particular case, we also received written consent stating, “You’re welcome to use the content as you please” (Lawson, 2020).

For this study, we followed Braun and colleagues’ (2018) general web scraping process: a) identify the website address(es), b) identify the data on the website to be extracted, c) write a script to extract the data, and d) execute the script to download the data onto a computer.

**Identify the Website Address(es).** After you have identified the home page of the website, you must identify the specific pages that contain the data you wish to extract. For example, the website for the governors’ COVID-19 press briefings is https://www.rev.com/blog/transcript-tag/coronavirus-update-transcripts. Many times, the data are not all available on this single page, but rather several pages. For example, at the time of this writing, there are 108 pages of COVID-19 press briefings on the Rev.com site. Fortunately, for web scraping purposes, most websites establish a stable pattern for labeling these pages (e.g., https://www.rev.com/blog/transcript-tag/coronavirus-update-transcripts/page/2; https://www.rev.com/blog/transcript-tag/coronavirus-update-transcripts/page/3; https://www.rev.com/blog/transcript-tag/coronavirus-update-transcripts/page/4). These patterns enable you to write a programming script, using your preferred programming language (e.g., R or Python), to access the web pages for scraping.

**Identify the Data on the Website to be Extracted.** Most web pages consist of some combination of text, images, tables, and/or hyperlinks to other websites or pages. On the front-end, the contents are often displayed in some aesthetically pleasing manner. On the backend, however, it is merely a combination of HTML (HyperText Markup Language) and CSS (Cascading Style Sheets). A full review of HTML and CSS is beyond the scope of this manuscript (for a detailed overview, see Mitchell, 2018), thus, we simply want to note that most websites use patterns, in the HTML and CSS, that enable researchers to write a programming script to extract the data from the web pages. There is an easy way to access the HTML and CSS to identify these patterns in the web pages: While viewing a web page on a web browser (e.g., Google Chrome), you can highlight a portion of the page you wish to extract, right click, and select “inspect”. This will give you access to the HTML and CSS for that particular section of the web page. There you will find the HTML tags and classes needed to write the programming script. For a more detailed explanation, we provide a step-by-step tutorial of this process in Appendix B.
Write a Script to Extract the Data. After you have identified the web page(s) you want to visit and identified patterns in the HTML/CSS, you can write a script to extract the data. We used the Beautiful Soup package (Richardson, 2020) in Python; however, there are options in R (R Core Team, 2022) as well (for a summary of R packages for web scraping, see Braun et al., 2018). Put simply, Beautiful Soup reads and parses HTML so that data can be extracted from a website. For reference, two examples of our web scraping Python scripts are provided in Appendix B (governors’ COVID-19 press briefings) and Appendix C (Presidents’ state of the union addresses).

Writing the script follows the same general steps as the overall scraping process. After loading your packages (e.g., Beautiful Soup, time, requests, pandas) you create a list of the web pages you wish to visit. If the website has multiple pages, like our Rev.com example, you may have to create an empty list and use a for loop to recreate the pattern of the website’s pagination to populate your website list. If the pages you visit merely hold preview content with links to the actual data (like our Rev.com example), you must create another empty list, write a for loop to visit each summary page, retrieve the hyperlinks to the pages with the actual data you wish to extract, and store the hyperlinks in the new list. Next, you write a for loop to visit each hyperlink that contains your desired data, retrieve the data from the specified HTML tags/classes, and store the data.

This is the phase of the process where responsible web scraping is critical. As noted in the Rev.com terms of service, websites do not want you to overload their servers with rapid requests. Doing so may not only violate the terms of use, but it may also result in your IP (internet protocol) address being blocked from accessing the website. This would hamper your data collection process. To avoid overloading the server, it is important to incorporate time delays in your web scraping script. The time package in Python allows you to put your script to “sleep” for a specified number of seconds before resuming the scraping process. This allows you to avoid overloading the server and, more pragmatically, avoid getting blocked from the website. Of course, this was a cursory overview of the script writing process. For a detailed overview of the entire process, using Beautiful Soup, see Mitchell’s (2018) book.

Execute the Script to Download the Data onto a Computer. The final step in the web scraping process is executing the script and downloading the data onto your computer. This phase of the process will be largely dependent on the type of data you are scraping and the format in which the data is received from the website. Using a combination of lists to store the data and post processing techniques (e.g., strip text, split text, relabel variables), you can prepare your data in Python (or R) to be downloaded/exported in a common data file format such as CSV (Comma-Separated Values) or XLSX (i.e., Microsoft Excel spreadsheet). In Python, the pandas and numpy packages can be used in conjunction to create a data frame and export the data in your desired format. A similar process could be followed in R, storing the data in data frames and exporting to .csv or .xlsx files via the openxlsx package.

Computer-Aided Text Analysis (CATA)

There are two broad categories of CATA: inductive and deductive. Inductive CATA, is often used to assist in qualitative research, such as grounded theory or thematic analysis (Short, McKenny, & Reid, 2018). Researchers may use computer-aided qualitative data analysis software (CAQDAS; O’Kane, Smith, & Lerman, 2021) specifically designed for inductive CATA, such as ATLAS.ti, NVivo, or QDA Miner, to visualize relationships between themes, capture text segments, or to simply assist in the organization of the coding process. As the name suggests, inductive CATA is an exploratory approach that is guided by human coding.

Our focus, however, is on deductive CATA, which takes a top-down approach and transforms textual data into quantitative data, often based on word frequency in specific dictionaries or dictionary categories (McKenny et al., 2018). The dictionaries are developed based on the premise that
specific words are related to certain higher-order concepts/constructs (Pang & Ring, 2020). These dictionaries are typically developed and validated using some variation of Short and colleagues’ (2010) recommendations to 1) create a working definition of the construct of interest, using a priori theory, 2) assess construct dimensionality based on existing literature, 3) develop an exhaustive list of key words that capture the construct of interest, and 4) consult with content experts to validate the word lists and assess rater reliability. If the research question does not necessitate the development of a new dictionary, researchers may use pre-existing validated dictionaries that are often available through published research or software programs.

There are several deductive CATA software programs available, including CAT Scanner (McKenny, Short, & Newman, 2012), DICTION (Hart & Carroll, 2012), and Linguistic Inquiry and Word Count (LIWC; Pennebaker et al., 2015). Additionally, there are several CATA packages in R, such as sentimentr, tm, stm, syuzhet, and quanteda. For this study, we chose LIWC for three reasons: 1) there are specific validated dictionaries that map onto our CIP construct very well (e.g., focuspresent, focusfuture, focuspast), 2) LIWC’s theoretically-derived dictionaries can be used to establish content validity of our operationalization and 3) LIWC has been used in high-quality journals to operationalize language in the political leadership context (e.g., Sergent & Stajkovic, 2020).

LIWC. We used LIWC2015 (Pennebaker et al., 2015) to analyze both the Presidential addresses (for model training and validation) and the governors’ COVID-19 press briefings (for model application). LIWC2015 is capable of analyzing text in a variety of formats, including plain text, PDF, RTF, .csv, and .xlsx. The program processes words sequentially throughout the text, adding incremental value to the dictionary category (or categories) to which the word belongs. If a word is present in more than one dictionary category, it is counted in all applicable categories. There are approximately 90 variables calculated throughout this process, including 41 dictionary categories designed to tap psychological constructs, as well as grammar/punctuation variables and other descriptive variables (e.g., word count, words per sentence, percentage of words longer than six letters). For reference, in Table 2, we provide a list of the variables we used in this study, along with sample words and the total number of words in each dictionary category.

Establishing validity. While CATA offers many psychometric advantages over traditional measures, such as self-report surveys and archival data (McKenny et al., 2018), the validity of the measure must be established (Short et al., 2010). To establish content validity of our CATA-based measure, after analyzing the Presidential addresses via LIWC2015, we assessed the differences among CIP style, based on theoretically relevant dictionary categories. For example, we conducted a one-way ANOVA and subsequent post-hoc Tukey test (Tukey, 1977) to assess whether C, I, and P differed, as theory suggests, in time orientation (Mumford, 2006). As you can see in Figure 2, C is significantly higher than I and P in the focusfuture variable, I is significantly higher than C and P in the focuspast variable, and P is significantly higher than I and C in the focuspresent variable. As time orientation is one of the key differentiating aspects among CIP styles, the fact that these LIWC2015 variables also captured these differences suggests that the text-based measure is appropriately differentiating among the CIP styles, thus, providing supporting evidence for the content validity.

Supervised Machine Learning

Supervised machine learning uses labeled input and output data (i.e., ground truth data) to build a model capable of predicting outcomes (e.g., classifications, labels) in unlabeled data (Janasik et al., 2009). In other words, with the labeled data, the model learns how predictor variables, referred to as features in supervised machine learning, map onto the known outcomes, such that those features can be used to predict unknown classifications in new data. Although the algorithms may vary based on the outcome variable of interest, supervised machine learning can be used to predict continuous
and categorical outcome variables. There are many dedicated packages in R that allow for the prediction of specific types of outcome variables (e.g., \textit{neuralnet, e1071, kernlab}). For this study, however, we used the \textit{caret} package (Kuhn, 2021) as it provides a highly flexible interface that allows for the execution of numerous algorithms, within the package, to predict whichever type of outcome variable is relevant to the study of interest. In other words, although we used \textit{caret} to predict a categorical outcome variable (CIP), the same package can be used to predict continuous outcome variables, as well, by simply changing the “method” within the \textit{caret} code. As such, our

| Feature Name (case sensitive) | Dictionary Category | Sample Words | Total Words in Category |
|------------------------------|---------------------|--------------|------------------------|
| achieve                      | Achievement         | Goal, able, proud | 213                    |
| adverb                       | Common adverbs      | Very, really | 140                    |
| affect                       | Affective processes | Best, loss, success | 1393                   |
| article                      | Articles            | A, an, the   | 3                      |
| assent                       | Assent              | Agree, OK, cool | 36                     |
| auxverb                      | Auxiliary verbs     | Am, will, have | 141                    |
| cogproc                      | Cognitive processes | Wisest, doubtful, mistaken | 797                   |
| Dash                         | Dashes              | NA (punctuation) | NA                     |
| differ                       | Differentiation     | Hasn’t, but, else | 81                     |
| filler                       | Fillers             | I mean, you know | 14                     |
| focusfuture                  | Future focus        | Hopefully, ahead, someday | 97                     |
| focuspast                    | Past focus          | Accepted, knew, remember | 341                   |
| focuspresent                 | Present focus       | Present, depends, currently | 424                   |
| hear                         | Hear                | Listen, hearing | 93                     |
| home                         | Home                | Roommates, chores, families | 100                   |
| informal                     | Informal language   | NA           | 380                    |
| insight                      | Insight             | Think, know | 259                    |
| leisure                      | Leisure             | Travel, play, parties | 296                   |
| negate                       | Negations           | No, not, never | 62                     |
| nonflu                       | Non-fluencies       | Er, hm, umm | 19                     |
| number                       | Numbers             | Second, thousand | 36                     |
| prep                         | Prepositions        | To, with, above | 74                     |
| quant                        | Quantifiers         | Few, many, much | 77                     |
| risk                         | Risk                | Hesitant, doubt, unsure | 103                   |
| see                          | See                 | View, saw, seen | 126                    |
| sexual                       | Sexual              | Passion, sex, lover | 131                   |
| social                       | Social processes    | Party, danced, team | 756                   |
| space                        | Space               | Open, locals, border | 360                   |
| swear                        | Swear words         | Hell, damn, prick | 131                   |
| time                         | Time                | Date, today, current | 310                   |
| Tone                         | Emotional tone      | NA (summary category) | NA                     |
| verb                         | Common verbs        | Eat, come, carry | 1000                   |
| WC                           | Word count          | NA (summary category) | NA                     |
| we                           | 1\textsuperscript{st} person plural | We, us, our | 12                     |
| work                         | Work                | Company, startup, sector | 444                   |
| WPS                          | Words per Sentence  | NA           | NA                     |
| you                          | 2\textsuperscript{nd} person pronoun | You, your, ya’ll | 30                     |

Note. Dictionary categories, sample words, and totals were identified in Pennebaker et al. (2015).
example (and R code) is generalizable to nearly any supervised machine learning model. The caret package is essentially a one-stop shop for supervised machine learning in R, as it is possible to complete each phase of the machine learning process within the package, including data pre-processing, prediction algorithms, feature selection, and cross-validation.

**Data Pre-Processing.** The overarching objective of supervised machine learning is to train a model capable of accurately predicting outcomes in unseen data. To minimize noise in the data, and maximize the accuracy of the model, there are certain pre-processing steps that can be taken. For example, centering is a pre-processing technique that subtracts the mean score from each feature (i.e., predictor variable). Scaling is another pre-processing technique that normalizes the feature data, often using either decimal scaling, min-max normalization, log transformation, or z-score normalization (Nwanganga & Chapple, 2020). In the caret package, the scale function simply divides each feature by its standard deviation, therefore, in the present study, we used both the center and scale options in the caret package to effectively implement a z-score normalization.

Another pre-processing consideration, for classification problems, such as predicting CIP, is to determine if the sample is balanced among the classes (e.g., CIP). In other words, is the sample relatively equally distributed among the classes? This is important because using supervised machine learning on a dataset with a large imbalance in the classes may result in high accuracy scores, however, the results can be misleading (Mohammed, Rawashdeh, & Abdullah, 2020). For instance, a poor performing model might assign every observation to the most represented category, and thereby look accurate, but the accuracy would purely be a byproduct of the data imbalance, not

**Figure 2.** Boxplots with Tukey post hoc of time orientation by CIP style. Note. C, I, and P stand for charismatic, ideological, and pragmatic, respectively.
model performance. Hence, if there is substantial imbalance, researchers often use data manipulation techniques, such as over- or under-sampling (referred to as upsampling or downsampling in the caret package) to balance the data among the classes (Nguyen, Cooper, & Kamei, 2012). Over-sampling is a process of randomly sampling instances from a minority class and then duplicating them until the minority class has the same number of instances as the majority class. Under-sampling is a process of selecting a number of random instances from the majority class that is equal to the number of instances in the minority class. For the present study, our sample had some imbalance among the CIP classes with 74, 83, and 220 instances per class, respectively; however, after comparing model performance metrics with and without over- and/or under-sampling, we determined that the data manipulation techniques did not improve the model. Thus, we did not use either over- or under-sampling. We did, however, ensure that the k-folds were stratified (see Cross-Validation Methods below for a description of k-folds), such that each fold had a relatively equal number of instances for each category (Kohavi, 1995).

**Prediction Models/Algorithms.** There are dozens of prediction algorithms used in supervised machine learning. Some of the algorithms are specifically used for continuous outcome variables, such as linear regression, lasso regression, and ridge regression; whereas others are used specifically for categorical outcomes, such as linear discriminant analysis, logistic model trees, naïve Bayes, and multinomial logistic regression; and others can predict both continuous and categorical outcomes. As CIP is a categorical outcome, we tested the predictive accuracy of eight algorithms that can classify categorical outcomes: 1) Linear Discriminant Analysis, 2) Logistic Model Trees, 3) Multinomial Logistic Regression, 4) Regularized Discriminant Analysis, 5) Neural Networks, 6) k-Nearest Neighbors, 7) Random Forests, and 8) Naïve Bayes. A summary of these eight algorithms is provided in Table 3 and a comparison of model performance is provided in Table 4 and Figure 3.

**Feature Selection.** When it comes to predictor variables or features, as they are referred to in supervised machine learning, having more is not always better. The decision on how many features to include in the model is often driven by sample size guidelines and model performance (e.g., accuracy). To reduce the number of features, researchers use a process referred to as feature selection, which is a process of identifying relevant features (i.e., predictor variables) and removing those that may be redundant or unimportant to the prediction model (Yu & Liu, 2004). Feature selection can be driven by theory, data, or, as is often the case, some combination of the two. For example, based on CIP theory, all time orientation variables (focusfuture, focuspast, focuspresent) should be important for the differentiation among CIP styles. Based on theory alone, this would suggest that these three features should be included in the model. However, including only these three features results in a poor performing model (mean accuracy = .58; mean kappa1 = .10), so we needed to determine which other features to include.

As we chose to use multinomial logistic regression (MLR), we followed feature selection guidelines for MLR, which recommend having a minimum of ten instances per feature (Starkweather & Moske, 2011). Accordingly, with a sample size of 377 instances, we were only able to select a maximum of 37 features to include in our model. To identify which 37 features to include, we used the varImp function, of the caret package in R, to rank the features from most important to least important for classifying the CIP outcomes. As shown in Figure 4, our theoretically relevant time-orientation features are the 2nd (focuspresent), 8th (focuspast), and 60th (focusfuture) most important features in our model. Although focusfuture falls outside of the top 37, rather than dismiss a theoretically important feature, we instead chose the top 36 features and added focusfuture as the 37th feature in our final model.

**Cross-Validation Methods.** There are various approaches to cross-validation that can be used in supervised machine learning. For example, Leave One Out Cross-Validation (LOOCV) is the simplest cross-validation approach in which the data is divided into two non-overlapping parts and these two parts are used for training and validating the model, respectively (Yadav & Shukla,
The strength of this approach is that it reduces the likelihood of overfitting a model, however, it may not provide optimum model accuracy, as the training data is limited to one subset of the data. Bootstrap Cross-Validation, or Repeated Hold Out Cross-Validation, is another form of LOOCV in which the process of sub-setting the data into training and validation sets is repeated several times, with replacement (Kim, 2009). In other words, the training and validation data may be recycled as the data is randomly subset with each iteration. The strength of this approach is that it provides more training data, which can improve the accuracy of the model, however, there is an increased risk of overfitting the model to the idiosyncrasies of dataset. Similarly, Bootstrap .632 Cross-Validation random samples and resamples the data, with replacement, however the procedure adjusts for upward bias in the error rate (Efron & Tibshirani, 1997). Finally, k-fold Cross-Validation (k-fold CV) is a process in which the data is split into k equal parts from which the model is trained on k – 1 parts and validated on the remaining one part (Kohavi, 1995). This process is then repeated k number of times, such that each fold is used exactly one time as the validation subset. With k-fold CV, the number of folds is arbitrary, however, 10 folds is the most popular choice (Kim, 2009). For this study, we chose to use k-fold CV with 10 folds and repeated the process three times, for a total of 30 iterations of training and validation. With each repeat of the k-fold CV, the decay rate, a hyper-parameter used to minimize overfitting (Ismail, Ahmad, Soh, Hassan, & Harith, 2019), was

Table 3. Classification Algorithm Descriptions.

| Algorithm                 | Description                                                                                                                                                                                                 |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Linear Discriminant Analysis (LDA) | "The LDA technique is developed to transform the features into a lower dimensional space, which maximizes the ratio of the between-class variance to the within-class variance, thereby guaranteeing maximum separability" (Tharwat, Gaber, Ibrahim, & Hassani, 2017, p. 170). |
| Logistic Model Trees (LMT)  | "...consists of a standard decision tree with logistic regression at the leaves" (Landwehr, Hall, & Frank, 2005, p.173).                                                                                     |
| Multinomial Logistic Regression (MLR) | MLR is similar to the more oft-used logistic regression, with a primary difference being that it is used when there are more than two categories in the dependent variable. The goal of MLR is to model the odds of a particular categorical outcome, as a function of the model covariates (Hosmer & Lemeshow, 2015). |
| Regularized Discriminant Analysis (RDA) | RDA is an extension of LDA that is particularly useful for smaller samples with a large number of features, as it incorporates regularization parameters to decrease larger eigenvalues and increase smaller eigenvalues, which counteracts the bias inherent in sample-based estimation (Friedman, 1989). |
| Neural Networks (NNET)     | "Neural networks are a wide class of flexible nonlinear regression and discriminant models, data reduction models, and nonlinear dynamical systems. They consist of an often large number of ‘neurons,’ i.e. simple linear or nonlinear computing elements, interconnected in often complex ways and often organized into layers” (Sarle, 1994, p. 1). |
| k-Nearest Neighbors (kNN)  | "kNN classifier is to classify unlabeled observations by assigning them to the class of the most similar labeled examples” (Zhang, 2016, p. 2).                                                             |
| Random Forests (RF)        | "Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest” (Breiman, 2001, p. 5). |
| Naïve Bayes (NB)           | Bayesian classifiers assign the most likely class to a given instance, as it is described by its feature vector. Naïve bayes classifiers simplify this process by assuming that features are independent given class (Rish, 2001). |
automatically manipulated within the caret package (0, .0001, and .1, respectively) to identify an optimum decay rate for model accuracy. As our decision was purely data-driven, based on the model performance, an optimum decay rate of .1 was selected for our model. The results of all 30 training-validation iterations are presented in Table 5. To better illustrate the k-fold CV process, in Figure 5, we provide a visual depiction of the process with 10 folds.

Multinomial Logistic Regression (MLR)

After comparing the performance of eight different machine learning algorithms, we settled on using multinomial logistic regression (MLR) for our model, as it had the highest accuracy and kappa values, on average. Commonly referred to as a discrete choice model, MLR is similar to the more oft-used logistic regression, with a primary difference being that it is used when there are more than two categories in the dependent variable. In general, the goal of MLR is to model the odds of a particular categorical outcome, as a function of the covariates (i.e., features) in the model (Hosmer & Lemeshow, 2015). Like binary logistic regression, MLR uses maximum likelihood to estimate the probability of category membership (Starkweather & Moske, 2011). More precisely, we used MLR, in this study, to construct a model capable of explaining the relationship between the features (LIWC categories) and the labeled outcomes (CIP styles), so that the model could accurately predict CIP in the governors’ COVID-19 press briefings, for which we did not have a CIP classification. As previously noted, although there are multiple R packages available to conduct MLR (e.g., mnlogit, VGAM, maxent, nnet), we used caret

Table 4. Algorithm Accuracy and Kappa Performance.

|           | Min   | Median | Mean   | Max   | SD   |
|-----------|-------|--------|--------|-------|------|
| MLR       | 0.789 | 0.865  | 0.867  | 0.974 | 0.061|
| NNET      | 0.789 | 0.851  | 0.862  | 0.946 | 0.055|
| LMT       | 0.757 | 0.867  | 0.857  | 0.946 | 0.061|
| RDA       | 0.784 | 0.855  | 0.852  | 0.946 | 0.055|
| LDA       | 0.737 | 0.868  | 0.846  | 0.919 | 0.070|
| KNN       | 0.730 | 0.842  | 0.843  | 0.919 | 0.054|
| RF        | 0.711 | 0.769  | 0.785  | 0.892 | 0.065|
| NB        | 0.649 | 0.746  | 0.753  | 0.892 | 0.074|

|           | Min   | Median | Mean   | Max   | SD   |
|-----------|-------|--------|--------|-------|------|
| MLR       | 0.647 | 0.760  | 0.768  | 0.957 | 0.105|
| NNET      | 0.647 | 0.735  | 0.762  | 0.900 | 0.094|
| LMT       | 0.577 | 0.767  | 0.750  | 0.904 | 0.105|
| RDA       | 0.602 | 0.749  | 0.738  | 0.900 | 0.096|
| LDA       | 0.519 | 0.776  | 0.731  | 0.859 | 0.125|
| KNN       | 0.552 | 0.721  | 0.727  | 0.859 | 0.085|
| RF        | 0.451 | 0.583  | 0.597  | 0.793 | 0.126|
| NB        | 0.366 | 0.579  | 0.572  | 0.801 | 0.129|

Note. Each algorithm was run using all available features (i.e., all LIWC dictionaries). MLR = multinomial logistic regression, NNET = neural network, LMT = logistic model trees, RDA = regularized discriminant analysis, LDA = linear discriminant analysis, KNN = k-nearest neighbors, RF = random forests, NB = naive Bayes.
Model Specification. Prior to running an MLR, the researcher has to establish which variables to include in the model. There are multiple ways to select variables and, ultimately, the researcher uses a combination of theory, statistics, and discretion to choose the final model. As described above, we used the varImp function, of the caret package in R, to identify the most important features for classifying the CIP outcomes, ultimately choosing 37 features to include in the final model.
Following notation from Hastie and colleagues (2009), the resulting model can be expressed in the following two equations:

$$\log \left( \frac{\Pr(CIP = I)}{\Pr(CIP = C)} \right) = \beta_{10} + \beta_{11}(achieve) + \beta_{12}(adverb) \ldots \beta_{137}(you) + \epsilon$$  \hfill (1)

$$\log \left( \frac{\Pr(CIP = P)}{\Pr(CIP = C)} \right) = \beta_{20} + \beta_{21}(achieve) + \beta_{22}(adverb) \ldots \beta_{237}(you) + \epsilon$$  \hfill (2)

Equation 1 represents the logit, where a one-unit increase (or decrease) in the independent variable(s) is associated with an increase (or decrease) of the log odds for the model selecting $I$ versus $C$. Similarly, Equation 2 represents the logit, where a one-unit increase (or decrease) in the independent variable(s) is associated with an increase (or decrease) of the log odds for the model selecting $P$ versus $C$. The final model had high mean accuracy (87.7%) and kappa (78.3%) values with relatively stable performance.
across cross-validation samples (accuracy $SD = .06$, kappa $SD = .11$). To view the full model, including the coefficients, standard errors, and Wald statistics for each independent variable, see Table 6.

**Application to New Data.** After systematically building, training, and validating the model, we finally applied the model to the governors’ COVID-19 press briefing dataset. Unlike the Presidential dataset, there were no known CIP outcomes in the governor dataset. Hence, there were no accuracy scores to be calculated. Instead, we relied upon the training/validating procedure in the Presidential address dataset to build our confidence in the model’s ability to accurately assess CIP in the governor dataset. Ultimately, the model classified the press briefings based on the CIP category with the highest probability of being correct. A subset of the modeled probabilities is provided in Table 7.

**Model Validation.** After applying the model to the governors’ press briefings, we consulted with three subject matter experts (SME), who were also familiar with some of the governors’ press briefings, to

---

**Table 5.** Fit and Accuracy Summary of Final Multinomial Logistic Regression Model.

| Accuracy | Kappa  | decay | Resample |
|----------|--------|-------|----------|
| 0.730    | 0.545  | 0     | Fold01   |
| 0.784    | 0.623  | 0.1   | Fold01   |
| 0.757    | 0.583  | 0.0001| Fold01   |
| 0.895    | 0.814  | 0     | Fold02   |
| 0.921    | 0.856  | 0.1   | Fold02   |
| 0.921    | 0.861  | 0.0001| Fold02   |
| 0.872    | 0.785  | 0     | Fold03   |
| 0.949    | 0.914  | 0.1   | Fold03   |
| 0.897    | 0.827  | 0.0001| Fold03   |
| 0.757    | 0.560  | 0     | Fold04   |
| 0.784    | 0.601  | 0.1   | Fold04   |
| 0.757    | 0.560  | 0.0001| Fold04   |
| 0.811    | 0.670  | 0     | Fold05   |
| 0.865    | 0.765  | 0.1   | Fold05   |
| 0.838    | 0.712  | 0.0001| Fold05   |
| 0.892    | 0.815  | 0     | Fold06   |
| 0.838    | 0.713  | 0.1   | Fold06   |
| 0.919    | 0.859  | 0     | Fold07   |
| 0.892    | 0.808  | 0.1   | Fold07   |
| 0.865    | 0.765  | 0.0001| Fold07   |
| 0.974    | 0.957  | 0     | Fold08   |
| 0.949    | 0.913  | 0.1   | Fold08   |
| 0.974    | 0.957  | 0.0001| Fold08   |
| 0.842    | 0.725  | 0     | Fold09   |
| 0.921    | 0.860  | 0.1   | Fold09   |
| 0.816    | 0.684  | 0.0001| Fold09   |
| 0.789    | 0.628  | 0     | Fold10   |
| 0.868    | 0.775  | 0.1   | Fold10   |
| 0.816    | 0.686  | 0.0001| Fold10   |

*Note. Accuracy is the percentage of predictions that matched with the known outcomes. Kappa is the percentage of predictions that matched with known outcomes, controlling for the expected accuracy, based on class distribution. Decay is a regularization parameter used to avoid overfitting.*
establish face validity of the measure of CIP in governors. Each SME we consulted has published multiple scholarly journal articles and book chapters on CIP. Each SME conversation was separate, and we did not define any criteria for agreement. After showing the CIP results to the SMEs, and having a discussion, there were two points of contention: 1) lack of charismatic representation in the governors’ press briefings and 2) lack of stability in the CIP style over time. For example, one scholar stated, “I remember ‘learning’ that charismatic leadership tends to emerge during times of crisis – when people look for inspiration and direction, so I was surprised to not see as much charisma, given COVID.” However, after discussing the problem solving strengths of different CIP styles (Griffith et al., 2018; Lovelace et al., 2019) and the fact that COVID-19 is not a typical crisis event (i.e., it is ongoing and people are dying every day), the scholar reconsidered her initial reaction and acknowledged, “That makes sense and is interesting.”

Regarding stability of CIP overtime, the initial push back (only from one SME) was alleviated after we discussed the limitations of the historiometric approach, used in prior empirical studies, to monitor the fine-grained detail of CIP overtime. Further, we discussed the anecdotal evidence of a situational aspect of CIP, referring to Barack Obama’s shift from a charismatic campaigner to a pragmatic President (Yammarino, Sotak, & Serban, 2020), which helped to validate the findings for that SME. Another SME stated, “I always thought CIP stuff was stable,” but added, “validation evidence [is] good.” Ultimately, after comparing our results with their expected results, the SMEs indicated that the measure was assessing CIP in the governors with relative accuracy. As such, this step provided further support for content validity (Short et al., 2010).

As a reminder, the CIP styles of the Presidents were established in a human-coded historiometric study (Yammarino et al., 2013). Thus, in addition to establishing support for content validity, these results also suggest convergent validity (Campbell & Fiske, 1959). Regarding external validity, we chose Presidential addresses to develop our text-based model because of the similarity in context (i.e., political leadership) and content (addresses and briefings) to our test sample of U.S. state governors’ COVID-19 press briefings.
| Variable Name | I Coef. | I Std. Err. | I z | P Coef. | P Std. Err. | P z |
|---------------|---------|-------------|----|---------|-------------|----|
| Intercept     | 0.56    | 1.21        | 0.47 | 4.87    | 0.93        | 5.24 |
| achieve       | -1.53   | 0.80        | -1.92 | -0.49   | 0.62        | -0.79 |
| adverb        | 2.56    | 1.36        | 1.88 | 1.81    | 1.14        | 1.58 |
| affect        | -0.11   | 1.10        | -0.10 | -1.77   | 0.77        | -2.29 |
| article       | 0.30    | 0.88        | 0.35 | -1.32   | 0.74        | -1.79 |
| assent        | 0.99    | 1.05        | 0.94 | 1.45    | 0.94        | 1.54 |
| auxverb       | 0.86    | 1.20        | 0.71 | -0.88   | 0.89        | -0.99 |
| cogproc       | 0.28    | 1.56        | 0.18 | 0.02    | 1.23        | 0.01 |
| Dash          | 1.11    | 1.30        | 0.86 | 2.29    | 1.18        | 1.94 |
| differ        | 0.38    | 1.16        | 0.33 | 0.55    | 0.94        | 0.58 |
| filler        | -1.66   | 0.96        | -1.73 | -0.16   | 0.53        | -0.30 |
| focusfuture   | 0.52    | 0.74        | 0.71 | -0.29   | 0.60        | -0.48 |
| focuspast     | 1.08    | 1.36        | 0.79 | -0.09   | 1.17        | -0.08 |
| focuspresent  | -2.12   | 2.33        | -0.91 | 1.16    | 1.94        | 0.60 |
| hear          | 1.37    | 0.99        | 1.38 | 0.68    | 0.79        | 0.86 |
| home          | -0.45   | 0.79        | -0.57 | 0.20    | 0.58        | 0.34 |
| informal      | 1.45    | 1.41        | 1.03 | 0.45    | 1.08        | 0.41 |
| insight       | 1.03    | 1.08        | 0.96 | -0.61   | 0.84        | -0.72 |
| leisure       | 0.81    | 0.87        | 0.93 | -0.62   | 0.64        | -0.96 |
| negate        | 1.56    | 1.05        | 1.48 | -0.08   | 0.78        | -0.10 |
| nonflu        | 3.48    | 1.37        | 2.54 | -0.15   | 1.09        | -0.14 |
| number        | 1.70    | 0.87        | 1.96 | -0.16   | 0.71        | -0.22 |
| prep          | 2.46    | 0.86        | 2.85 | 1.08    | 0.70        | 1.53 |
| quant         | -1.24   | 0.80        | -1.55 | -0.88   | 0.66        | -1.34 |
| risk          | -0.64   | 0.93        | -0.69 | -1.16   | 0.67        | -1.74 |
| see           | -1.22   | 0.83        | -1.47 | -0.60   | 0.65        | -0.92 |
| sexual        | -2.72   | 1.28        | -2.13 | 0.03    | 0.64        | 0.05 |
| social        | -0.91   | 1.19        | -0.76 | -1.55   | 0.95        | -1.62 |
| space         | -0.53   | 0.79        | -0.67 | -0.74   | 0.63        | -1.17 |
| swear         | -0.24   | 1.87        | -0.13 | -1.23   | 1.38        | -0.89 |
| time          | 1.25    | 0.91        | 1.37 | 1.50    | 0.76        | 1.99 |
| Tone          | 0.23    | 0.93        | 0.25 | -0.53   | 0.66        | -0.81 |
| verb          | -1.00   | 2.69        | -0.37 | -0.70   | 2.29        | -0.31 |
| WC            | 1.09    | 0.77        | 1.42 | -0.99   | 0.58        | -1.70 |
| we            | 2.25    | 1.12        | 2.01 | 1.94    | 0.95        | 2.04 |
| work          | 0.30    | 0.82        | 0.37 | -0.53   | 0.65        | -0.81 |
| WPS           | 0.55    | 0.92        | 0.60 | 1.25    | 0.82        | 1.52 |
| you           | -1.53   | 1.07        | -1.43 | 0.14    | 0.69        | 0.20 |

Note. Coeff. = coefficient, Std. Err. = standard error, and z = Wald statistic.
Results for Illustration

A summary of the results is provided in Table 8. In brief, the breakdown of the press briefing CIP categorization was as follows: 3 charismatic (< 0.01%), 174 ideological (20.84%), and 658 pragmatic (78.80%). This means that less than 1% of the press briefings used language associated with a charismatic approach, such as a future-oriented and positive emotional expression of the governor’s vision. With thousands of people dying every day, a charismatic approach, as defined by the CIP model (Mumford, 2006), would have likely been socially inappropriate. With nearly 80% of the briefings classified as pragmatic, the data suggests that governors, in general, communicated a data-driven approach to solving the present issues, without dwelling on the past or focusing on the future, in response to the COVID-19 pandemic. This makes sense as there have been, and continue to be,
Table 8. Total Governor Press Briefings Classified by CIP Category.

| Governor Name     | State | C | I | P |
|-------------------|-------|---|---|---|
| Governor Abbott   | TX    | 0 | 8 | 18|
| Governor Baker    | MA    | 0 | 7 | 19|
| Governor Beshear  | KY    | 0 | 0 | 12|
| Governor Brown    | OR    | 0 | 0 | 9 |
| Governor Carney   | DE    | 0 | 0 | 1 |
| Governor Cooper   | NC    | 0 | 0 | 25|
| Governor Cuomo    | NY    | 3 | 36| 121|
| Governor DeSantis | FL    | 0 | 12| 50|
| Governor DeWine   | OH    | 0 | 24| 60|
| Governor Ducey   | AZ    | 0 | 0 | 6 |
| Governor Edwards | LA    | 0 | 2 | 16|
| Governor Evers    | WI    | 0 | 0 | 2 |
| Governor Gordon   | WY    | 0 | 0 | 1 |
| Governor Grisham  | NM    | 0 | 0 | 1 |
| Governor Herbert | UT    | 0 | 0 | 2 |
| Governor Hogan    | MD    | 0 | 5 | 11|
| Governor Holcomb  | IN    | 0 | 1 | 3 |
| Governor Hutchinson | AR | 0 | 0 | 4 |
| Governor Inslee   | WA    | 0 | 1 | 17|
| Governor Ivey     | AL    | 0 | 3 | 4 |
| Governor Justice  | WV    | 0 | 0 | 1 |
| Governor Kelly    | KS    | 0 | 1 | 3 |
| Governor Kemp     | GA    | 0 | 3 | 5 |
| Governor Lamont   | CT    | 0 | 2 | 8 |
| Governor Lee      | TN    | 0 | 0 | 7 |
| Governor Little   | ID    | 0 | 1 | 1 |
| Governor McMaster | SC   | 0 | 1 | 6 |
| Governor Murphy   | NJ    | 0 | 15| 33|
| Governor Newsom  | CA    | 0 | 47| 44|
| Governor Noem     | SD    | 0 | 0 | 3 |
| Governor Northam  | VA    | 0 | 0 | 37|
| Governor Parson   | MO    | 0 | 0 | 2 |
| Governor Polis    | CO    | 0 | 0 | 6 |
| Governor Pritzker | IL    | 0 | 2 | 39|
| Governor Raimondo | RI    | 0 | 0 | 3 |
| Governor Reeves   | MS    | 0 | 1 | 2 |
| Governor Reynolds | IA    | 0 | 1 | 13|
| Governor Sisolak  | NV    | 0 | 0 | 2 |
| Governor Stitt    | OK    | 0 | 0 | 1 |
| Governor Walz     | MN    | 0 | 0 | 6 |
| Governor Whitmer  | MI    | 0 | 1 | 46|
| Governor Wolf     | PA    | 0 | 0 | 8 |
| TOTAL             |       | 3 | 174| 658|

Note. C, I, and P stand for charismatic, ideological, and pragmatic, respectively. Predictions were made using the Multinomial Logistic Regression algorithm.
ample data collected and analyzed throughout the pandemic. Finally, 20% of the briefings were classified as *ideological*, which suggests that some governors occasionally used a communication strategy focused on the past and filled with negative emotion. Again, with thousands of people dying every day and a longing for life as we knew it (i.e., the past), it makes sense that some governors used an ideological style.

An important aspect of these results is the within-person variability of the CIP classifications over time, which is shown in Figure 6. Prior theorizing and empirical work on the CIP model have found and/or assumed CIP to be a trait-like, fixed quality in individuals. Perhaps the lack of granularity in the historiometric methodology, which has dominated the study of CIP to date (Lovelace et al., 2019), led researchers to perceive and theorize that CIP style is relatively stable in individuals. With our CATA and machine learning operationalization, however, we

![Figure 6. Governors’ press briefings classified as C, I, or P over time.](image)

*Note.* C, I, and P stand for *charismatic*, *ideological*, and *pragmatic*, respectively. Predictions were made using a multinomial logistic regression algorithm.
were able to capture the within-person change over time that has previously remained untested. Although there was variability, most governors tended to have a dominant style. For example, Governor Cooper (North Carolina) had all 25 briefings classified as \( P \), Governor Beshear (Kentucky) had all 12 briefings classified as \( P \), and Governor Northam (Virginia) had all 37 briefings classified as \( P \). Others were more balanced between two styles. For example, Governor Newsom (California) had 44 briefings classified as \( P \) and 47 classified as \( I \), and Governor DeWine (Ohio) had 60 briefings classified as \( P \) and 24 classified as \( I \). We attribute the relative lack of charismatic press briefings to the fact that these briefings all took place during a global pandemic. The positive emotional imagery and focus on the future that characterizes the charismatic style of leadership may have been difficult to portray, and likely ill-advised, when thousands of individuals were dying on a daily basis.

Discussion

We introduced a method, that combines web scraping, CATA, and supervised machine learning techniques to advance new streams of research. Leveraging labeled ground truth data, from a historiometric study (Yammarino et al., 2013), we illustrated how this method and operationalization can be used to predict categorical outcomes. In addition to classifying categorical outcomes, a similar approach could be used to assess continuous variables, such as personality (Harrison, Thurgood, Boivie, & Pfarrer, 2019) or implicit motives (Schultheiss, 2013), as well. Hence, this methodological combination could be applied in a variety of research disciplines and contexts. The process, as we have described, may be especially useful to researchers who study populations that are difficult to assess via traditional research methods (e.g., upper echelons). Not all aspects of the method, however, need to be used in conjunction, as we have done. Instead, we envision each aspect of our method (web scraping, CATA, machine learning) as a stand-alone module that can aide micro-, meso-, and macro-oriented researchers as they expand their research tool kits. In brief, not only does this text-based approach open doors to new data sources, but it may also spur new streams of theoretical inquiries.

Method Limitations and Future Research

There are certainly limitations to our study and this method. First, although 87% seems to be high accuracy percentage for predicting CIP, based on running text, there is still a 13% chance that the model predicted the CIP style incorrectly. The second limitation to this method is that it purely relies upon words and dictionary categories to classify individuals, without any other behavioral or contextual information factored into the model. Although this method appears to explain a large portion of the variance among CIP styles, it may be missing some valuable information that cannot be fully captured from the words alone. A third limitation, to this particular study, is the fact that all of the press briefings took place during a global pandemic. The gravity of the situation may have overpowered an individual leader’s natural style and necessitated a certain style of communication. Although this provides additional theoretical considerations for the study of CIP, it may place a constraint on our method’s ability to generalize to less situationally constrained times (i.e., normal, non-crisis life). On the other hand, this limitation could also be viewed as a strength, as restricting our sample to a consistent context may minimize situational confounds and allow for CIP variance to be attributed to the individuals, rather than the context.

We put forth this method in hopes of stimulating and generating new streams of future research. By combining web scraping, CATA, and supervised machine learning techniques there are many possibilities. For example, researchers can use other ground truth datasets in
which data are classified by certain categorical variables and apply the approach demonstrated in this study. Similarly, researchers could identify ground truth datasets in which individuals are scored on continuous scales of other psychological constructs (e.g., personality, goal orientation) and apply a similar, modified approach. With a continuous variable as an outcome, rather than using multinomial logistic regression, the researcher would simply train a linear model using one of the various algorithms available in the caret package. With access to paid textual datasets, such as LexisNexis Full Disclosure Wire or Wharton Research Data Service (WRDS), one could skip the web scraping portion of the method and jump right into the CATA and machine learning. Beyond the use of ground truth datasets and archival textual data, new online communication platforms (e.g., Zoom) provide another avenue for using the methods we put forth in this manuscript. For example, researchers could collect psychometric data from participants and have the participants engage in some sort of online Zoom activity (or perhaps use an already recurring Zoom activity, such as a college course), save the transcripts from the Zoom session, and proceed with the above-mentioned methods. This method could also be applied in mixed methods studies in which researchers collect both survey data and qualitative data from participants. Put simply, this method is broadly applicable to a variety of research topics and disciplines.

**Theoretical Implications for CIP**

While the primary focus of this manuscript was to advance a research method for using textual data to classify outcomes, there are some theoretical implications as well. Similar to the way in which experience sampling methods (ESM; Gabriel et al., 2019) allowed researchers to capture the within-person variation of personality over time (Fleeson, 2001, 2004), our text-based operationalization of CIP will allow researchers to further examine the stability assumptions of CIP that were previously untestable using historiometric methods (Lovelace et al., 2019). Stated differently, our method of assessing CIP from running text allows researchers to examine CIP from multiple levels of analysis (e.g., within and between individuals), over time, and across situations, rather than being limited to merely aggregating CIP across time and situations, assuming any within-person fluctuations to be error.

**Conclusion**

There is a blue ocean of text available on the internet. There is also a plethora of research studies conducted on public figures. We put forth a method that combines web scraping, CATA, and supervised machine learning techniques to leverage these data sources to classify categorical outcomes. To illustrate the implementation of this methodology, we classified U.S. governors’ COVID-19 press briefings according to their categorical CIP leadership styles. Additionally, as this method is highly flexible, we opened the door for other researchers to build their research streams as well.

**Appendix A: Accounting for Language Change Over Time**

To assess whether the Presidents’ language changed over, we first subset the data into four categories: pre-1900, 1900 – 1950, 1950 – 2000, and post-2000. Then, we separated the data by CIP type, which allowed us to create boxplots with post-hoc Tukey significance tests to determine whether the LIWC dictionary categories differed among time periods, which would suggest language change over time. As shown in Figures A1, A2, and A3, most of the LIWC dictionary categories showed significant differences among the time periods, suggesting that the language did change over time.
**Figure A1.** LIWC Dictionaries for Charismatic Presidential Addresses Over Time.
Note. Green boxes indicate non-significant differences among time periods.

**Figure A2.** LIWC Dictionaries for Ideological Presidential Addresses Over Time.
Note. Green boxes indicate non-significant differences among time periods.
As the language appeared to change over time, we systematically tested which dates provided the most accurate predictions. Starting with 1900, we stepped through each decade, reducing the sample to only those speeches that occurred after 1910, 1920, 1930, etc. As shown in Table A1 below, post-1970 and 1980 provided similar results, which were better than those samples that included older speeches. Instinctively, we were inclined to use the post-1970 sample, however, upon further inspection, we saw that the additional 101 speeches included in the 1970 data (compared to 1980) were all labeled as “P”, which would have exacerbated the imbalance of that data (this can be problematic in supervised machine learning). Therefore, we ultimately chose to use the post-1980 dataset as our final training-validation data.

Table A1. Comparing Model Accuracy Over the Years.

| Model   | Sample Size | Accuracy | Kappa | Accuracy (SD) | Kappa (SD) |
|---------|-------------|----------|-------|---------------|------------|
| Mod1900 | 795         | .775     | .566  | .041          | .082       |
| Mod1910 | 780         | .778     | .567  | .038          | .067       |
| Mod1920 | 743         | .803     | .584  | .068          | .143       |
| Mod1930 | 721         | .807     | .585  | .029          | .066       |
| Mod1940 | 694         | .833     | .630  | .057          | .118       |
| Mod1950 | 640         | .847     | .663  | .039          | .085       |
| Mod1960 | 568         | .818     | .629  | .037          | .080       |
| Mod1970 | 478         | .857     | .715  | .045          | .102       |
| Mod1980 | 377         | .867     | .768  | .061          | .105       |

Note. Performance is based on the multinomial logistic regression algorithm, using k-folds crossvalidation with 10 folds and 3 repeats. Results reported used a decay equal to .10. All 92 LIWC features were used in each model.
Appendix B: Web Scraping Process and Python Script for Rev.com

Figure B1. Example of Pagination on Rev.com Website.

Figure B2. Highlight Relevant Section, Right Click, and Select “Inspect” to Access HTML.
Python Markdown Script for Scraping COVID-19 Press Briefings on Rev.com

Import Libraries

```python
import requests
from bs4 import BeautifulSoup
import time
```

**Figure B3.** HTML for Highlighted Section of Rev.com Website.

**Figure B4.** Sample Rev.com Transcript from Governor Cuomo, with HTML Displayed.

**Python Markdown Script for Scraping COVID-19 Press Briefings on Rev.com**

**Figure B4.** Sample Rev.com Transcript from Governor Cuomo, with HTML Displayed.
import pandas as pd
import numpy as np

Create a header for the scraper (your user agent can be found here: https://www.whatismybrowser.com/detect/what-is-my-user-agent)

headers = {
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
}

Create a list for storing all the main pages

main_page_list = []

Write a for loop to create a list of all main page urls

for i in range(1, 109):  # From page 1 to the total number of pages + 1
    main_str1 = 'https://www.rev.com/blog/transcript-tag/coronavirus-update-transcripts' + '/page/' + str(i)
    print(main_str1)
    main_page_list.append(main_str1)
    print(main_page_list)

Create a list to store the hyperlinks for the transcript sources

hyperlinks = []

Create a for loop to retrieve the urls of the transcript pages

for link in main_page_list:
    print("Now scraping from the main page:", link)
    time.sleep(3)  # Important for responsible web scraping – pause scraping for 3 seconds
    main_page = requests.get(link, headers)
    main_soup = BeautifulSoup(main_page.content, "html.parser")
    speakers = main_soup.find_all("div", class_="fl-post-column")
    for speaker in speakers:
        try:
            hyperlink = speaker.a['href']
        except:
            hyperlink = "NA"
        print(hyperlink)
        hyperlinks.append(hyperlink)

Create more lists to store transcript page data

data = []
page_count = 0  # Used to numerically identify the page from which the row of data was scraped

Create a for loop to scrape transcript pages

for hyperlink in hyperlinks:
    print("Now scraping:", hyperlink)
    time.sleep(3)  # Important for responsible web scraping – pause scraping for 3 seconds
link_page = requests.get(hyperlink, headers)
link_soup = BeautifulSoup(link_page.content, "html.parser")

try:
    date = link_soup.find("div", class_="fl-rich-text").get_text().strip()
except:
    date = "NA"

text = link_soup.find("div", class_="fl-callout-text")
status = "Found"
p_tags = text.find_all("p") # Each speech is contained in a <p> tag.
Extracting all <p> tags

for p in p_tags:
    p_children = [] # Empty list to store the children of each <p> tag
    for child in p.children: # Obtaining all the children of each <p> tag
        p_children.append(child)
    speaker = p_children[0]
    print(speaker)
    try:
        speaker_cleaned = speaker[ :-3]
    except:
        speaker_cleaned = speaker
    print(speaker_cleaned)
    try:
        time_stamp = p_children[1].get_text().strip()
    except:
        time_stamp = "NA"
    print(time)
    try:
        speech = p_children[4].strip()
    except:
        speech = "NA"
    print(speech)
data.append((page_count, hyperlink, status, date, time_stamp, 
speaker, speaker_cleaned, speech))
page_count = page_count + 1
print(data)

Combine data into a dataframe
df = pd.DataFrame(np.array(data))

Identify columns to include in the dataframe
df.columns = ['Page Count', 'Hyperlink', 'Status', 'Date', 'Time',
'Speaker', 'Speaker Clean', 'Speech Text']

Print dataframe
print(df)

Export data to Excel file on the local desktop
df.to_excel('~/Desktop/RevData.xlsx', index=False, header=True)
Appendix C: Python Markdown Script for Scraping U.S. Presidents’ State of the Union Addresses from “The American Presidency Project”

Import Libraries

```python
import requests
from bs4 import BeautifulSoup
import time
import pandas as pd
import numpy as np
```

Create a header for the scraper (your user agent can be found here: https://www.whatismybrowser.com/detect/what-is-my-user-agent)

```python
headers = {
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.88 Safari/537.36"
}
```

Create lists to store transcript page data

```python
dates = []
names = []
titles = []
texts = []
```

Create a list for storing all the main pages

```python
main_page_list = []
```

Page zero does not have a page number, so it must be added to the list before adding others

```python
page_zero = 'https://www.presidency.ucsb.edu/documents/app-categories/written-messages/presidential/state-the-union-messages'
main_page_list.append(page_zero)
print(main_page_list)
```

Write a for loop to create a list of all main page urls

```python
for i in range(1,15):
    main_str1 = 'https://www.presidency.ucsb.edu/documents/app-categories/written-messages/presidential/state-the-union-messages?' + "page=" + str(i)
    print(main_str1)
    main_page_list.append(main_str1)
print(main_page_list)
```

Create a list to store the hyperlinks for the transcript sources

```python
hyperlinks = []
```

Create a for loop to retrieve the URLs of the transcript pages

```python
for link in main_page_list:
    print("Now scraping from the main page:", link)
    main_page = requests.get(link, headers=headers)
    main_soup = BeautifulSoup(main_page.content, "html.parser")
```
urls = main_soup.find_all("div", class_ = 'field-title')
for url in urls:
    try:
        hyperlink = url.a['href']
    except:
        hyperlink = "NA"
    hyperlinks.append(hyperlink)
    time.sleep(3) # Important for responsible web scraping - pause scraping for 3 seconds
print(hyperlinks)

Create a for loop to scrape transcript pages
for hyperlink in hyperlinks:
    print("Now scraping:", hyperlink)
    link_page = requests.get('https://www.presidency.ucsb.edu' + str(hyperlink), headers=headers)
    link_soup = BeautifulSoup(link_page.content, "html.parser")
    try:
        date = link_soup.find("span", class_ = 'date-display-single').get_text().strip()
    except:
        date = "NA"
    dates.append(date)
    try:
        name = link_soup.find("h3", class_ = 'diet-title').get_text().strip()
    except:
        name = "NA"
    names.append(name)
    try:
        title = link_soup.find("div", class_ = 'field-ds-doc-title').get_text().strip()
    except:
        title = "NA"
    titles.append(title)
    try:
        text = link_soup.find("div", class_ = "field-docs-content").get_text().strip()
    except:
        text = "NA"
    texts.append(text)
    time.sleep(3) # Important for responsible web scraping - pause scraping for 3 seconds

Print lists
print(dates)
print(names)
print(titles)
print(hyperlinks)
print(texts)
Combine data into a dataframe

data_output = pd.DataFrame({"date": dates, "name": names, "titles": titles, "text": texts})

Print dataframe

print(data_output)

Export data to Excel file in the same folder as the Python script

data_output.to_excel('PresidentsSOTU_messages.xlsx', index=False, header=True)

Note: you could also specify a location to export the file

Appendix D: R Markdown Script for Supervised Machine Learning

STEP ONE: TRAIN AND VALIDATE A MODEL

Import the Ground Truth Data

mydata<-readxl::read_excel("~/Desktop/CIP Project/Text Data/LIWC2015 Results (Merged_Pres_12.19.2020(no Trump)).xlsx")

Set Your Outcome Variable as Factor

mydata$CIP <- as.factor(mydata$CIP)

Preview the Data - The CIP column is the outcome variable that we will use to train a model

head(mydata[1:6,c(1,3,6)])

## # A tibble: 6 x 3
## #> name titles CIP
## #> <chr> <chr> <fct>
## #> 1 Abraham Line... Inaugural Address C
## #> 2 Abraham Line... Inaugural Address C
## #> 3 Andrew Jacks... Inaugural Address I
## #> 4 Andrew Jacks... Inaugural Address I
## #> 5 Barack Obama Commencement Address at the United States Air Force
## #> Academy. P
## #> 6 Barack Obama Remarks to the United Nations General Assembly in New
## #> York. P

Load Required Packages

Format the Date Column, Create a New Column Called “Year”, Use Only Data After 1980 – See Appendix A for Details on Why We Did This

# Convert the Date to Date Format
library(lubridate)

mydata$date <- mdy(mydata$date)

#Create a New Column Called "Year"
mydata$year <- lubridate::year(mydata$date)
# Create New Dataset with Only Data After 1980
v1980 <- subset(mydata, mydata$year > 1980)

Remove Non-LIWC Variables from the Dataset

# Create a New Dataset Called "newdata" Excluding Non-LIWC Variables
newdata <- subset(v1980, select = -c(name, date, titles, link, text, 'Power Orientation (S or P), year, 'function'))

# Make Sure CIP is a Categorical Variable in the New Dataset
newdata$CIP <- as.factor(newdata$CIP)

Stratify Subsamples (for Categorical Outcome Variables) and Set Cross Validation Method

#library(caret)
set.seed(123)
stratindex <- createFolds(factor(newdata$CIP), 10, returnTrain = T)  # Create an index to stratify folds so they are balanced across subsamples
train.control <- trainControl(method = "repeatedcv", index = strat.index, returnResamp = "all",
                              number = 10, repeats = 3)  # Method = Cross Validation Method, Number = # of folds, Repeats = # of repeated k-fold repeats

Compare Algorithm Performance

#### Run Multiple Algorithms Using All Features (i.e., LIWC variables)
####
# This Section Was Adapted from Brownlee (2016)

# Random Forest
fit.rf <- train(CIP ~ ., data = newdata, method = "rf",
               preProcess = c("center", "scale"),
               trControl = train.control)

# Naive Bayes
fit.nb <- train(CIP ~ ., data = newdata, method = "nb",
               preProcess = c("center", "scale"),
               trControl = train.control)

# k-Nearest Neighbors
fit.knn <- train(CIP ~ ., data = newdata, method = "kknn",
                 preProcess = c("center", "scale"),
                 trControl = train.control)

# Latent Discriminant Analysis
fit.lda <- train(CIP ~ ., data = newdata, method = "lda",
                 preProcess = c("center", "scale"),
                 trControl = train.control)

# Multinomial Logistic Regression
fit.mlr <- train(CIP ~ ., data = newdata, method = "multinom",
                 preProcess = c("center", "scale"),
                 trControl = train.control)
# Regularized Discriminant Analysis
```
fit.rda <- train(CIP ~ ., data = newdata, method = "rda",
                preProcess = c("center", "scale"),
                trControl = train.control)
```

# Logistic Model Trees
```
fit.lmt <- train(CIP ~ ., data = newdata, method = "LMT",
                preProcess = c("center", "scale"),
                trControl = train.control)
```

# Neural Network
```
fit.nnet <- train(CIP ~ ., data = newdata, method = "nnet",
                 preProcess = c("center", "scale"),
                 trControl = train.control)
```

Compare Performance Results Among Algorithms

# Collect Results for Each Fold (aka "resamples") for Each Algorithm
```
results <- resamples(list(RF=fit.rf, NB=fit.nb, KNN=fit.knn, LDA=fit.lda, 
                         MLR=fit.mlr, RDA=fit.rda, LMT=fit.lmt, NNET=fit.nnet))
```

# Create Summary Results Table
```
summary(results)
```

## Call:
```
  summary.resamples(object = results)
```

## Models: RF, NB, KNN, LDA, MLR, RDA, LMT, NNET

## Number of resamples: 10

## Accuracy
```
  Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
RF 0.6842105 0.7583570 0.7692308 0.7774592 0.8077881 0.8648649 0
NB 0.6486486 0.7123819 0.7464438 0.7532699 0.7801455 0.8918919 0
KNN 0.7297297 0.8378378 0.8421053 0.8434840 0.8794452 0.9189189 0
LDA 0.7368421 0.7972973 0.8683299 0.8619767 0.8967611 0.9189189 0
MLR 0.7894737 0.8120555 0.8648649 0.8668381 0.9135482 0.9743590 0
RDA 0.7837838 0.8001012 0.8548164 0.8518018 0.8918919 0.9459459 0
LMT 0.7567568 0.8083215 0.8666430 0.8567951 0.910256 0.9459459 0
NNET 0.7631579 0.8186344 0.8818434 0.8593555 0.8967611 0.9189189 0
```

## Kappa
```
  Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
RF 0.3903743 0.5501576 0.5773474 0.5798525 0.6253294 0.7562582 0
NB 0.3657143 0.5112838 0.5788534 0.5723725 0.6140850 0.8008075 0
KNN 0.5520581 0.7069303 0.7208596 0.7269607 0.7773974 0.8587786 0
LDA 0.5189873 0.6261849 0.7758347 0.7314123 0.8233079 0.8585987 0
MLR 0.6465116 0.6753768 0.7601227 0.7679404 0.8489453 0.9569536 0
RDA 0.6021505 0.6709856 0.7489733 0.7378934 0.8043552 0.9004038 0
LMT 0.5774112 0.6802900 0.7666478 0.7499499 0.8205207 0.9041451 0
NNET 0.5952663 0.6892834 0.8024946 0.7559225 0.8244062 0.8587786 0
```
Create a Box Plot to Visually Compare Performance Among Algorithms

```
# Create Box Plots of Results
scales <- list(x=list(relation="free"), y=list(relation="free"))
bwplot(results, scales=scales, col = "red")
```

Note: Multinomial Logistic Regression Has the Highest Mean Accuracy (and Kappa), so We’ll Use That Going Forward

Feature Selection for MLR

```
Top_Feat <- varlmp(fit.mlr)
print(Top_Feat, 10) # Look at the top 10 most important
## multinom variable importance
##
## only 10 most important variables shown (out of 92)
##
## Overall
## adverb   100.00
## focuspresent 92.89
## prep     84.34
## nonflu   78.48
## assent   56.81
## informal 55.12
## quant    52.60
## focuspast 49.47
## insight  47.95
## WC      47.91
```
Create a Plot of the Top Features in Order of Importance

\begin{verbatim}
plot(Top_Feat)
\end{verbatim}

MLR Guidelines Recommend 10 Instances per Feature. With 377 Instances, We Could Only Have 37 Features.

\begin{verbatim}
# MLR Model with 37 Features (top 36 + focusfuture)
fit.mlr37 <- train(CIP ~ achieve + adverb + affect + article + assent + auxverb +
                    cogproc + Dash + differ + filler + focusfuture + focuspast +
                    focusesent + hear + home + informal + insight + leisure + negate + nonflu +
                    number +
                    prep + quant + risk + see + sexual + social + space + swear + time +
                    Tone + verb + WC + we + work + WPS + you, data = newdata, method =
                    "multinom",
                    preProcess = c("center", "scale"),
                    trControl = train.control)
\end{verbatim}

Capturing coefficients, standard errors, and z-values of the Final Model

\begin{verbatim}
# Code Adapted from https://datasciencebeginners.com/2018/12/20/multinomial-logistic-regression-using-r/
# Coefficients
coefficients <- summary(fit.mlr37[ [ "finalModel" ] ])$coefficients
# Show Coefficients
coefficients
\end{verbatim}
# Standard Errors

```r
std.err <- summary(fit.mlr37[['fmalModel']])$standard.errors

# Show Standard Errors
std.err
```

```
## (Intercept) achieve adverb affect article assent auxverb
## I 1.2054669 0.7971325 1.410476 1.0759832 0.8668030 0.8877197 1.2238263 -0.9987197 1.0912367
## P 0.9286142 0.6249246 1.1439996 0.7740976 0.7352607 0.9400907 0.9443751 -0.9987197 1.0912367
## cogproc Dash differ filler focus future focus past focus present
## I 1.557941 1.298565 1.1625151 0.9598461 1.364940 2.325465
## P 0.7380863 1.2348619 1.504751 -0.5346349 -0.6983755 -0.9859869 1.943301
## hear home informal insight leisure negate nonflu number
## I 0.9910954 0.7912180 1.410476 1.0759832 0.8668030 1.0522846 1.2238263 -0.9987197 1.0912367
## P 0.7903262 0.5822706 1.081582 0.8403637 0.6424737 0.7809309 0.8674989 0.7118399
## prep quant risk see sexual social space
## I 0.8629786 0.7956257 0.9316140 0.8345318 1.2774302 1.190461 0.7878365
```

# Calculate z-values

\[
\text{zvalues} \leftarrow \frac{\text{summary(fit.mlr37["fmalModel"])}\text{coefficients}}{\text{summary(fit.mlr37["fmalModel"])}\text{standard.errors}}
\]

# Show z-values - AKA Wald Statistic #

\[
zvalues
\]

## (Intercept) achieve adverb affect article assent auxverb
### I 0.4660534 -1.9193174 1.880011 -0.1007889 0.3473227 0.936175 0.7145292
### P 5.2409130 -0.7905148 1.580284 -2.2871159 -1.7944430 1.540767 -0.9937647
## cogproc Dash differ filler focus future focus past
### I 0.1785554 0.8561643 0.3256985 -1.7328811 0.7061713 0.78863695
### P 0.0134672 1.9395991 0.5809341 -0.2987596 -0.4787425 -0.0798659
## focus present hear home informal insight leisure negate
### I -0.9128218 1.3823445 -0.5656500 1.0283553 0.9558978 0.9308950 1.4833240
### P 0.5966303 0.8594856 0.3386335 0.4122955 -0.7226342 -0.9625314 -0.1032291
## nonflu number prep quant risk see sexual
### I 2.5427439 1.9628624 2.850432 -1.553244 -0.6917349 -1.4664826 -2.13203676
### P 0.01398657 -0.2196387 1.532370 -1.339215 -1.7372056 -0.9178463 0.04868727
## social space swear time Tone verb WC
### I -0.7644919 -0.6700712 -0.1288549 1.368223 0.2505282 -0.3715112 1.422239
### P 0.16217865 -1.1746591 -0.8940920 1.987716 -0.8113358 -0.3055547 -1.701742
## we work WPS you
### I 2.005235 0.3700507 0.5971225 -1.4251634
### P 2.037637 -0.8050929 1.5242498 0.2001102

**STEP TWO: APPLY THE MODEL TO PREDICT OUTCOMES IN UNLABELED DATA (CIP in this case)**

Import Data for Classification

```r
Full_Gov<-readxl::read_excel("~/Desktop/CIP Project/Text Data/Full_Gov_Data_June2021.xlsx")
```
Convert the Date Column for Future Use

```r
# Convert the Date to Date Format
library(lubridate)
Full_Gov$Date <- mdy(Full_Gov$Date)
```

Run the Model

```r
# Predicting CIP in the Full Gov Dataset
Full_Gov$predicted <- predict(fit.mlr37, newdata = Full_Gov, "raw")
summary(Full_Gov$predicted)
```

Create a Summary of the Predictions by Governor

```r
# Summary Table by Governor
gov_CIP <- table(Full_Gov$ 'Speaker Clean', Full_Gov$predicted) gov_CIP
```
Create a Plot of CIP of Governors’ Press Briefings Over Time

```r
# ggplot(data = Full_Gov) +
  geom_point(mapping = aes(x = Date, y = 'Speaker Clean', color = predicted)) +
  ylab("Governor Names")
```

```
# Optional
# Export Table as txt File
# write.table(gov_CIP, file = "~/Desktop/Full_Gov_Predictions.txt", sep = ",", quote = FALSE, row.names = T)
```

```
## Governor Parson 0 0 2
## Governor Polis 0 0 6
## Governor Pritzker 0 2 39
## Governor Raimondo 0 0 3
## Governor Reeves 0 1 2
## Governor Reynolds 0 1 13
## Governor Sisolak 0 0 2
## Governor Stitt 0 0 1
## Governor Walz 0 0 6
## Governor Whitmer 0 1 46
## Governor Wolf 0 0 8
```
Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iD

Jason D. Marshall https://orcid.org/0000-0001-9899-9307

Note

1. Kappa is the percentage of predictions that matched with known outcomes, controlling for the expected accuracy, based on class distribution. In other words, kappa is a weighted accuracy metric.

References

Aguinis, H., & Bakker, R. M. (2021). Time is of the essence: Improving the conceptualization and measurement of time. Human Resource Management Review, 31(2). https://doi.org/10.1016/j.hrmr.2020.100763
Aguinis, H., Suarez-Gonzalez, I., Lannelongue, G., & Joo, H. (2021). Scholarly impact revisited. Academy of Management Perspectives, 26(2), 105–132. https://doi.org/10.5465/amp.2011.0088
Antonakis, J., Bastardoz, N., Jacquart, P., & Shamir, B. (2016). Charisma: An ill-defined and ill-measured gift. Annual Review of Organizational Psychology and Organizational Behavior, 3(1), 293–319. https://doi.org/10.1146/annurev-orgpsych-041015-062305
Banks, G. C., Woznyj, H. M., Wesslen, R. S., & Ross, R. L. (2018). A review of best practice recommendations for text analysis in R (and a user-friendly app). Journal of Business and Psychology, 33(4), 445–459. https://doi.org/10.1007/s10869-017-9528-3
Bligh, M. C., Kohles, J. C., & Meindl, J. R. (2004). Charisma under crisis: Presidential leadership, rhetoric, and media responses before and after the September 11th terrorist attacks. Leadership Quarterly, 15(2), 211–239. https://doi.org/10.1016/j.leaqua.2004.02.005
Braun, M. T., Kuljajanin, G., & DeShon, R. P. (2018). Special considerations for the acquisition and wrangling of big data. Organizational Research Methods, 21(3), 633–659. https://doi.org/10.1177/10944281177690235
Breiman, L. (2001). Random forests. Machine Learning, 45, 5–32. https://doi.org/10.1023/A:1010933404324
Brownlee, J. (2016). Machine learning mastery with R: Get started, build accurate models and work through projects step-by-step. Melbourne, Australia. https://machinelearningmastery.com/machine-learning-with-r/
Campbell, D. T., & Fiske, D. W. (1959). Convergent and discriminant validation by the multitrait-multimethod matrix. Psychological Bulletin, 56(2), 81–105. https://doi.org/10.1037/h0046016
DeTienne, K. B., DeTienne, D. H., & Joshi, S. A. (2003). Neural networks as statistical tools for business researchers. Organizational Research Methods, 6(2), 236–265. https://doi.org/10.1177/1094428103251907
Efron, B., & Tibshirani, R. (1997). Improvements on cross-validation: The .632+ bootstrap method. Journal of American Statistical Association, 92(438), 548–560. Retrieved from http://www.jstor.org/stable/2965703.
Fleeson, W. (2001). Toward a structure- and process-integrated view of personality: Traits as density distributions of states. Journal of Personality and Social Psychology, 80(6), 1011–1027. https://doi.org/10.1037/0022-3514.80.6.1011
Fleeson, W. (2004). Moving personality beyond the person-situation debate. Current Directions in Psychological Science, 13(2), 83–87. https://doi.org/10.1111/j.0963-7214.2004.00280.x
McKenny, A. F., Aguinis, H., Short, J. C., & Anglin, A. H. (2018). What doesn’t get measured does exist: improving the accuracy of computer-aided text analysis. *Journal of Management, 44*(7), 2909–2933. https://doi.org/10.1177/0149206316657594

McKenny, A. F., Short, J. C., & Newman, S. M. (2012). CAT Scanner. Retrieved from http://www.catscanner.net/.

McKenny, A. F., Short, J. C., & Payne, G. T. (2012). Using computer-aided text analysis to elevate constructs: an illustration using psychological capital. *Organizational Research Methods, 16*(1), 152–184. https://doi.org/10.1177/1094428112459910

Minbashian, A., Bright, J. E. H., & Bird, K. D. (2010). A comparison of artificial neural networks and multiple regression in the context of research on personality and work performance. *Organizational Research Methods, 13*(3), 540–561. https://doi.org/10.1177/1094428109335658

Mitchell, R. (2018). *Web scraping with python* (A. MacDonald & J. Billing, Eds.) (2nd ed.). O’Reilly Media, Inc.

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine learning with oversampling and undersampling techniques: Overview study and experimental results. 2020 11th International Conference on Information and Communication Systems, ICICS 2020, 243–248. https://doi.org/10.1109/ICICS49469.2020.239556.

Mumford, M. D. (2006). *Pathways to outstanding leadership: A comparative analysis of charismatic, ideological, and pragmatic leaders*. Lawrence Erlbaum Associates.

Nguyen, H. M., Cooper, E. W., & Kamei, K. (2012). A comparative study on sampling techniques for handling class imbalance in streaming data. 6th International Conference on Soft Computing and Intelligent Systems, and 13th International Symposium on Advanced Intelligent Systems, SCIS/ISIS 2012, 1762-1767. https://doi.org/10.1109/SCIS-ISIS.2012.6505291.

Nwanganga, F., & Chapple, M. (2020). *Practical machine learning in R*. Wiley & Sons, Inc. https://doi.org/10.1002/9781119591542.

O’Kane, P., Smith, A., & Lerman, M. P. (2021). Building transparency and trustworthiness in inductive research through computer-aided qualitative data analysis software. *Organizational Research Methods, 24*(1), 104–139. https://doi.org/10.1177/10944281119865016

Pandey, S., & Pandey, S. K. (2019). Applying natural language processing capabilities in computerized textual analysis to measure organizational culture. *Organizational Research Methods, 22*(3), 765–797. https://doi.org/10.1177/1094428117745648

Pang, J. S., & Ring, H. (2020). Automated coding of implicit motives: A machine-learning approach. *Motivation and Emotion, 44*(4), 549–566. https://doi.org/10.1007/s11031-020-09832-8

Pennebaker, J. W., Booth, R. J., Boyd, R. L., & Francis, M. E. (2015). *Linguistic inquiry and word count: LIWC2015 operator’s manual*. Pennebaker Conglomerates. Retrieved from www.LIWC.net

Pollach, I. (2012). Taming textual data: The contribution of corpus linguistics to computer-aided text analysis. *Organizational Research Methods, 15*(2), 263–287. https://doi.org/10.1177/1094428111417451

R Core Team (2022). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing.

Rev. (2020). Terms of service. Retrieved from https://www.rev.com/about/terms.

Richardson, L. (2020). Beautiful soup documentation.

Rish, I. (2001). An empirical study of the naive Bayes classifier. In *IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence* (pp. 41–46).

Sarle, W. S. (1994). Neural networks and statistical models. In *Nineteenth Annual SAS Users Group International Conference, April, 1994* (pp. 1–13). https://doi.org/10.5023/jappstat.24.77.

Schultheiss, O. C. (2013). Are implicit motives revealed in mere words? Testing the marker-word hypothesis with computer-based text analysis. *Frontiers in Psychology, 4*(OCT), 1–20. https://doi.org/10.3389/fpsyg.2013.00748

Serban, A., Yammarino, F. J., Sotak, K. L., Banoeng-Yakubo, J., Mushore, A. B. R., Hao, C., & Mumford, M. D. (2018). Assassination of political leaders: The role of social conflict. *Leadership Quarterly, 29*(4), 457–475. https://doi.org/10.1016/j.leaqua.2018.01.003
Sergent, K., & Stajkovic, A. D. (2020). Women’s leadership is associated with fewer deaths during the COVID-19 crisis: Quantitative and qualitative analyses of United States governors. *Journal of Applied Psychology, 105*(8), 771–783. https://doi.org/10.1037/apl0000577

Shamir, B., Arthur, M. B., & House, R. J. (1994). The rhetoric of charismatic leadership: A theoretical extension, a case study, and implications for research. *The Leadership Quarterly, 5*(1), 25–42. https://doi.org/10.1108/S1479-357120180000009016

Short, J. C., Broberg, J. C., Cogliser, C. C., & Brigham, K. H. (2010). Construct validation using computer-aided text analysis (CATA). *Organizational Research Methods, 13*(2), 320–347. https://doi.org/10.1177/1094428109335949

Short, J. C., McKenny, A. F., & Reid, S. W. (2018). More than words? Computer-aided text analysis in organizational behavior and psychology research. *Annual Review of Organizational Psychology and Organizational Behavior, 5*, 415–435. https://doi.org/10.1146/annurev-orgpsych-032117-104622

Starkweather, J., & Moske, A. K. (2011). Multinomial logistic regression. Retrieved January 2, 2021, from http://www.unt.edu/rss/class/Con/Benchmarks/MLR_JDS_Aug2011.pdf.

Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. (2017). Linear discriminant analysis: A detailed tutorial. *AI Communications, 30*(2), 169–190. https://doi.org/10.3233/AIC-170729

Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley Pub. Co.

Weber, M. (1924). *The theory of social and economic organizations*. Free Press.

Yu, L., & Liu, H. (2004). Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research, 5*, 1205–1224.

Zhang, Z. (2016). Introduction to machine learning: K-nearest neighbors. *Annals of Translational Medicine, 4*(11), 1–7. https://doi.org/10.21037/atm.2016.03.37

Zhou, Z. H. (2018). A brief introduction to weakly supervised learning. *National Science Review, 5*(1), 44–53. https://doi.org/10.1093/nsr/nwx106

**Author Biographies**

**Jason D. Marshall** is an incoming Assistant Professor of Management in the Heider College of Business at Creighton University. He earned his PhD in Management, with a concentration in Leadership & Organizational Science, from Binghamton University, State University of New York. With a broad interest in strategic leadership, Marshall’s research largely centers on using computer-aided text analysis (CATA) and machine-learning techniques to assess psychological constructs in new ways and in new contexts. His work has been published in *The Leadership Quarterly* and *American Journal of Health Education*.

**Francis J. Yammarino** is SUNY Distinguished Professor of Management and Director, Bernard M. & Ruth R. Bass Center for Leadership Studies, at Binghamton University, State University of New York. He was a senior editor of *The Leadership Quarterly* and a co-editor of *Research in Multi-Level Issues*, has served on 10 scholarly journal editorial review boards, and is a Fellow of the SIOP, APA, APS, and SOB. His research focuses on leadership and levels of analysis issues and has been funded by several federal agencies. In 2010, Dr. Yammarino
received the Distinguished Leadership Scholar Career Award from *The Leadership Quarterly* and *Academy of Management Network of Leadership Scholars*, and in 2021 he received the Sage Publications/RMD Distinguished Career Award from the *Academy of Management Research Methods Division*.

**Srikanth Parameswaran** is an Assistant Professor of Management Information Systems in the School of Management at Binghamton University. His research interests are in the areas of IT innovation, user-generated and web content, and technology-mediated health outcomes. His research has been published in *Information and Management (I&M), Journal of the American Medical Informatics Association (JAMIA), Information Systems Frontiers (ISF)*, and other outlets. He earned his MS in management information systems and PhD in Management Science and Systems from the State University of New York at Buffalo. He received his bachelor’s degree in Computer Science and Engineering from Anna University. He was featured on the Poets & Quants list of the “Top 50 Undergraduate Business Professors of 2021.”

**Minyoung Cheong** is an Assistant Professor of Management and Organization in the School of Graduate Professional Studies at Great Valley, Pennsylvania State University. He earned his PhD in Management from Binghamton University, State University of New York. His research interests relate to leadership with a primary interest in empowering leadership, organizational paradox, and levels of analysis and multilevel issues in leadership. His work has been published in *The Leadership Quarterly, Journal of Organizational Behavior, and Group & Organization Management*. 