Populism monitoring: web (text) data collection, cleaning and analysis.

George Makris and Ioannis Andreadis

Aristotle University of Thessaloniki

DEFINING POPULISM

During the last decades there has been an intensification on the study of populism. Studies of the populist phenomenon take place both on policy level (e.g. rise of populist parties and movements, effects of those actors on the party system etc.) and on discourse level.

The dominant scientific definition in the field of populism is given by the "ideational approach" (Kaltwasser et al., 2017; Mudde & Kaltwasser, 2017). According to this definition, the populist ideology considers society to be separated into two antagonistic groups: the people and the elite. More specifically, people is a homogeneous group that is morally good and pure, and elite is a group that is corrupted and opposes the people placing their own interest over the people's. Also, according to the populist ideology, politics should be an expression of the people's general will.

Although, there is still a debate in the scientific community regarding a precise conceptual clarification of the term populism, it seems like the aforementioned definition is the most capable to make up a methodological starting point for empirical studies and research on the populist phenomenon (Mudde, 2004; Mudde & Rovira Kaltwasser, 2018).The reason for that is the fact that with this particular definition it is easier to classify texts as populist (or not) as well as to detect populism, in general (Mudde & Rovira Kaltwasser, 2018). Also, the ideational approach has provided the research community with the capacity to study populism both from the supply and the demand side, using various methods (Hawkins & Rovira Kaltwasser, 2017).

So in the last few years, populism has begun to be studied on the supply side: media content (TV programs, newspapers etc.), party manifestos, political speeches etc., as well as on the demand side through survey questions on a common scale which effectively measure populism on the citizen level. For example, the 5th module of the CSES, a project which coordinates the collection of national election studies with a common core questionnaire in different countries, includes questions to measure populist attitudes[1]. Andreadis and Stavrakakis (2017) using data from the "Greek Candidates Survey of 2015" and the "Greek Voters Survey of 2015" compare the populist attitudes of candidates with those of the electorate through common questions.

The ideational approach for the study of populism doesn't take into account other ideologies which can be combined with populism and/or include it, such as socialism,

[1] https://cses.org/data-download/cses-module-5-2016-2021/

nationalism etc. (Hawkins & Rovira Kaltwasser, 2017). This is the reason why populism has been characterized as a "thin-centered ideology" (Mudde, 2004). Also, scholars using the ideational approach have not sufficiently examined the role of party organization and leadership in the rise of populist actors, as well as the reaction of mainstream political parties to the rise of the last (Hawkins & Rovira Kaltwasser, 2017). However, the ideational approach is, as we said before, the dominant paradigm so far, for the study of populism, as it is frequently employed by scholars and constitutes a minimum basic definition of a complex phenomenon, which for the first time provides scholars of populism with a common conceptual basis for conducting comparative research, quantitative and qualitative analyses etc. Therefore, this will be the approach followed in the present paper for the detection of populism on the internet.

There are other approaches to the populist phenomenon as well, which are not part of this paper such as the political-strategic and the socio-cultural (Mudde & Rovira Kaltwasser, 2018). The first (Weyland, 2017) argues that populism is an organized strategy of some charismatic leaders in order to rise to power and the second (Ostiguy, 2017) argues that populism is a style of policy-making where the leader tries to break social "taboos", behaving "weirdly" and against the dominant way of political communication.

BIG DATA AND COMPUTATIONAL SOCIAL SCIENCE

In recent years, there has been an explosion in the volume of generated data worldwide. This has led to the use of the term Big Data. Big Data can be structured, semi-structured or unstructured. Until the beginning of the big data era, data was predominantly structured, i.e. in a tabular format with rows (observations) and columns (variables). Since around 2008 unstructured data has dominated, such as words and text. Also, in terms of the size of the data in the respective databases, there has been a transition from terabytes to zettabytes. Accordingly, new technologies have been invented to support these huge amounts of data, such as "data lakes". Data lakes are a technology used by many enterprises (either through their own data centers or through cloud services from providers such as Google, Amazon, etc.) that own and manage big data in order to increase their data processing speed as well as to improve their data storage capacity. Accordingly, many new techniques have been created for the exploitation, transformation and analysis of big data, such as natural language processing and machine learning. A more thorough review of the second will take place below in the paper.

As a result of the developments that we have described above, many businesses, as well as the scientific community have recently made use of techniques for extracting text from the internet (text mining) and analyzing it (text analytics). One real-life application of text mining and analysis is the Google search engine, which uses artificial intelligence techniques such as machine learning and knowledge graphs, to enhance its search engine results. . Also applications are made by various companies, which, in order to improve the quality of services provided to their customers, they collect and analyze reviews (in text form) of them for various products.

As it is well known, the 2008 US presidential election campaign established a new paradigm for text mining and analysis for voter profiling and more effective and personalized political advertising. Accordingly, the 2016 presidential election campaign saw the same practices. These developments have inevitably reached the social sciences as well. As we said before, the management and analysis of a large amount of data requires

the use of tools by scientists that come from the field of IT and technology in general. For this reason, in recent years the term "computational social science" has been established and is frequently used[2].

Computational social science is somewhat of a neologism used to describe an emerging interdisciplinary field in which computer science and the social sciences converge. However, it has not yet been formally established as a separate field due to many obstacles, such as the very different scientific paradigms and models that the social sciences have with computer science, the insufficient training of social scientists in computational methods, citizens' concerns about the protection of their personal data when those are collected with automated methods such as mobile tracking, data collection from social media etc. (Lazer et al., 2009).

In the social sciences, computers are used mainly for two reasons: First, to analyze social behavior through simulations and social network analysis, and second, to analyze the content of mass media and especially social media. In the field of political science, the use of computers for the collection and analysis of big data has mainly been applied to the academic study of political communication (computational communication science) through the development of software for the collection and analysis of data from online websites and social media, through the tracking of internet users with the help of the digital footprint of each of them etc. (Theocharis & Jungherr, 2021). Also, apart from academia, computational methods are now increasingly used (especially in the USA as mentioned above) to effectively design candidates' election campaigns through targeted, personalized advertising (Tufekci, 2014), and to assist political decision-making (Kim, 2020). This tendency has even been called: computational politics (Haq et al., 2020; Tufekci, 2014).

In addition to the above, there are also other branches of political science that have begun to use computational social science methods. In this paper, references will be made to several computational social science methods such as the collection and analysis of text data and machine learning.

COMPUTER SCIENCE AND POLITICAL SCIENCE

In recent decades the greatest innovations at the intersection between political science and computer science have been presented in the "Political Analysis" journal, which is the official journal of the Society for Political Methodology and the Political Methodology section of the American Political Science Association (APSA). For the purposes of this paper, reference will be made to textual data collection and analysis, and machine learning. In fact, two issues have been published that summarize all the relevant articles of the journal on each of these issues (Cranmer, 2017; Roberts, 2016).

Content analysis (newspaper articles, political speeches, party manifestos etc.) is one of the long-established methodologies in political science. Though, the use of computers in recent years has led to new methodologies for automated content analysis. This particular innovation has helped (and will help much more in the future) political scientists to analyze huge volumes of text data, as it is inherently impossible for a researcher to read very large corpora of text, as well as very expensive to hire multiple

[2] https://computationalsocialscience.org/

coders to read and code huge volumes of text. Therefore, automated analysis enables any researcher to analyze and draw reliable results on large volumes of textual data easily and quickly on their computer, without necessarily being part of a highly expensive research project (Grimmer & Stewart, 2013).

According to Grimmer and Stewart (2013), there are four principles for automated content analysis: First, due to the complexity of languages, no automated computational model is completely correct and cannot accurately capture many subtle meanings of a language and easily draw causal conclusions (causal inference problem). However, it can be quite helpful in searching for useful information. Second, automated computational methods are useful to augment rather than completely replace human effort. Moreover, research has shown that human coders are more likely to discover more subtle meanings by reading the texts, compared to the computer, concluding that the ideal scenario is the researchers to find the most appropriate way to use a joint human and computational effort (Conway, 2006). Third, there is not just one method for applying automated text analysis to all situations. Fourth, all the results of the used models for text analysis always need manual verification to a greater or lesser extent. And correspondingly, based on the verification, a modification of the analysis method is also needed. For this very reason, automated text analysis is better done programmatically (that is, using a programming language, usually R or Python) and not with off-the-shelf commercial quantitative text analysis software where it is not easy to modify the analysis methods used.

As far as text analysis is concerned, there are many different computational methods. One method is supervised machine learning (Grimmer & Stewart, 2013). In this case, the researchers face a classification problem, where he needs to predict to which category each text document belongs. Thus, they read a part of the text themselves, classify it manually and train an algorithm on the labelled data (training set), in order for the algorithm to then predict the rest of the text by itself (test set). Regarding the evaluation of the model, this can be done in two ways. Either by the researcher coding some more texts manually and evaluating the already trained model on them (and then applying it to the rest) or by dividing the initial data (training set) into k random subsets (k resampling folds) training the algorithm on k-1 of them and evaluating it on the last one.

There are additional methods for text analysis such as dictionary methods (Grimmer & Stewart, 2013) in which there are dictionaries that assign each word a specific tone score and then the percentage of each word is calculated in the text in order to calculate an average tone score of the text. The most frequent application of this method is for sentiment analysis, which will be discussed below.

So far, in the literature there are not enough scientific articles on the study of populism through computational methods. However, some of the most interesting contributions are two. First, Cocco and Monechi (2021), , using machine learning on 268 election manifestos from 99 parties, classify sentences as populist or not and based on this classification calculate a populism score for various parties internationally. Second, Hawkins and Silva (2018) attempt to detect populist discourse in the election manifestos of 144 parties in 27 countries using two methods: one using the human factor and one automated method using machine learning; they conclude that the second can be very useful as long as there is enough data (in this case enough manifestos).

The present paper aims to detect populist discourse in well-known Greek news websites and their Twitter accounts, through computational methods for text analysis, a

methodological field that, as mentioned before, is characterized by the general title "automated content analysis". For this purpose, data was collected from the Internet using the methods of web scraping and Twitter API. The data was then cleaned and three methods were applied to the analysis in order to detect populism: sentiment analysis, word frequencies and machine learning. Finally, some analyses were conducted on the populist articles found.

DATA

Our research effort aims to extract data related to populism from news websites and from their respective Twitter accounts, and then to analyze the common corpus of those two. The data of this paper are articles from six well-known news websites: "kathimerini.gr", "efsyn.gr", "avgi.gr", tovima.gr", "tanea.gr", "protothema.gr". One of the websites from which articles were also collected is "real.gr", however it was subsequently removed from the analysis, as it was not possible to collect data from its Twitter account, as the API returned no tweets before the 25th of February. The aforementioned websites were chosen due to their high user traffic. The period during which the articles were collected was February 1-27, 2022. The articles were collected from two sources using two methods: from the websites using the web scraping method and from the Twitter accounts of the same websites using the Twitter API method. Web scraping was carried out three times a day with a time interval of 6 hours between them (10:00, 16:00, 22:00). Text mining via the Twitter API was performed once on March 3 and all tweets from the accounts of the above websites for the entire month of February were collected. All of the above procedures were performed by writing code in the R programming language. Below we present an analysis of these two methods.

Web Scraping

Web scraping is a method for extracting data from web pages. In political science, it is not such a widespread method, although it has been used (Jackman, 2006). All web pages consist of three elements. HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets) and Javascript. These elements make up the front-end part of the web pages, i.e. what the user sees. The server of a website sends these three elements to the user's browser and the browser in turn processes the code it receives to create the website so that the user can see it. HTML is used to create the basic structure of the website, CSS for its stylistic decoration and Javascript for creating its interactive elements.

More specifically, for web scraping we are mainly interested in the HTML and CSS code of a web page. The HTML code has the form of a tree with various branches (nodes) and various tags. By using the appropriate tags each time, we can extract from the website the text (or any other information) that interests us. On the other hand, the CSS code also uses tags to determine which HTML elements will be decorated. A typical CSS file contains objects called CSS Selectors (elements, id's, classes etc.) that contain certain morphological elements such as color, size etc. Thus, in order to decorate the elements inside the HTML code, one makes a connection with a CSS file and uses the selectors he/she wants in order to decorate different elements of the web page. We can use either HTML tags or CSS Selectors to locate and extract the data we want from the web page.

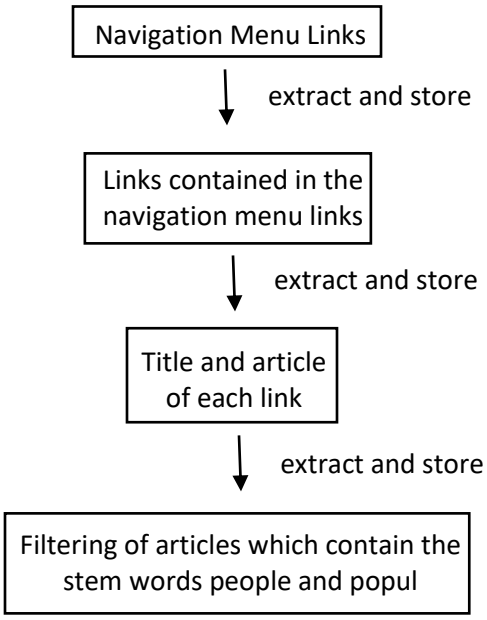
There is a tool as a plug-in of the Google Chrome browser, the selector gadget, which allows the users to interactively navigate with their mouse to the website they want,

select the information they want to extract and find easily and quickly the CSS Selector or HTML element that leads to this information. In this paper, we used the selector gadget to identify CSS selectors and HTML elements that link to articles and article titles. Then using those elements and the "rvest" library in R, we developed an algorithm that consists of four steps: 1) first it collects and stores all the links from the navigation menu of a web page, 2) then, it opens each of these links and collects and stores all the links contained in each, 3) finally, opens each of these links and collects and stores their content (title and article), 4) from the collected articles it filters only those that contain the stem words "λαο" and "λαϊκ" ("people" and "popul").

It should be noted that the program we have written for the above tasks is not applicable to all web pages, and this is one of the limitations of web scraping. Each web scraping script applies only to the website for which it was written, which has a specific HTML structure that differentiates it from others. Nevertheless, the logic of the algorithm presented in this paper can be easily applied to other websites as well, as long as the necessary adjustments are made for each one separately. The basic logic of the algorithm is visualized in the figure below (Figure 1).

Also, another limitation of web scraping is that even a small change or update to the source code of a web page can completely break the web scraping code related to it. For this very reason, the code presented in this paper cannot be used long-term. Anyone who wants to collect data for a long time needs to constantly monitor the code and make the necessary modifications whenever they need to be made. However, during the period in which the data for this paper were collected, there were no changes to the HTML code of the websites under consideration. The only errors that occurred were common errors such as 404 error, which occurs when the website server cannot recognize the client's request, most likely because the specific url requested by the client has been removed. However, these errors did not affect the course of the collection, as "exception handling" was performed through the "try" command in R, and any faulty links were presented in the R console and removed manually, without affecting the collection of the rest.

Figure 1: Steps of the web scraping algorithm



Twitter API

API's are software that consist of different endpoints where clients, whether they are people or software, can request and then receive specific information. In this way, an API acts as a "messenger" between a user and a database, receiving his request and returning the information he wants through a text file (.xml, .csv, .json).

To collect the tweets from the Twitter accounts of the websites, we used the TwitteR library in R. Before collecting the data, we created a developer account on Twitter in order to obtain the credentials that would allow us to collect the desired data. The data was collected on March 3 using the maximum limit of tweets that the Twitter API allows, 3200. Then we filtered only those posted in February (as some more were returned from the end of January and from the first days of March).

Of the tweets collected only 24 contained the stem words "λαο" and "λαϊκ". Since this number is very small we decided to combine them with the articles collected by the web scraping method (all of which contain the stem words "λαο" and "λαϊκ") in order to detect more. Thus, in an indirect way, using the common urls[3] between the two methods, we identified those tweets whose urls lead to website articles containing the stem words "λαο" and "λαϊκ") but don't mention them on the tweet text. The result was a new dataset with 527 articles (common ones). The following analysis was carried out on this dataset.

ANALYSIS

Before the analysis, the data was cleaned. Cleaning as a process involves removing all "noise" such as stopwords (e.g. articles, pronouns etc.) and less common words, punctuation, and capitalization. All of the above was performed for the data of this research. To remove stopwords we used a list containing Greek stopwords[4].

The main goal of the data analysis was to detect populism in a large volume of data (articles) as automated and fast as possible (always keeping in mind the limitations of automated content analysis described above), so that the researcher does not have to read and code all texts, but only a part of them. The analysis was done using three methods: counting the frequency of the words "λαο" and "λαϊκ", sentiment analysis and machine learning. The main goal was to find the best method which produces accurate results (correctly detects populist articles) and limits the number of articles the researcher needs to read. However, no method can completely replace reading for the reasons we mentioned in the previous section.

SENTIMENT ANALYSIS AND WORD FREQUENCY ANALYSIS

As most articles containing the words "λαο" and "λαϊκ" are not necessarily populist (based on the populism scheme we described in the first section) it is necessary to investigate which of them are indeed populist and which are not. The first two methods we applied to detect populism are counting the frequency of the words "λαο" and "λαϊκ" and sentiment analysis. Each method was applied to all of the articles and to verify their results we drew

[3] At the tweets of the news sites' accounts there is an initial text followed by a short url that links to the respective website article. To make the join between the articles from the web scraping and the tweets, the short urls were converted to long urls with the help of the "longurl" library in R

[4] https://github.com/stopwords-iso/stopwords-el

a small random sample of ten articles from each method, which we read and manually coded.

Sentiment analysis is a technique for detecting emotions in volumes of texts (e.g. anger, fear, joy etc.) but also polarity. Polarity is usually measured on a scale of -1 (negative) to 1 (positive) and is used to show whether the general sentiment of a text (or set of texts) is positive or negative. Sentiment analysis belongs to the so-called dictionary methods (Grimmer & Stewart, 2013), as presented above. There are dictionaries where each word is assigned a polarity score, usually from -1 to 1. Then, the words in the dictionary are counted in the text for which the researcher wants to calculate the sentiment, and an average sentiment score is calculated based on the score of each word.

In the present paper, the reason why sentiment analysis was chosen to detect populism is that it was hypothesized that an article in which the scheme of populism ("pure" people vs "corrupt" elite) is present is likely to contain more negativity due to the hostility towards the elites who "conspire" against the people. Also, it should be noted that sentiment analysis was not performed for the detection of individual sentiments, but to calculate the overall polarity of the articles.

A dictionary for the Greek language[5] was used to carry out the sentiment analysis. This dictionary was created by four reviewers and each one coded each word with one of the following categories "POS", "NEG", "N/A", i.e. positive, negative and neutral, respectively. In order to make the analysis easier, we converted the categories to numbers -1, 1 and 0 respectively and took an average of the scores of the four raters for each word. Then, we matched the words in dictionary with the words in the articles we collected[6]. The words in the dictionary are pretty much restrictive because they are only in the first person of the present tense if it is a verb (e.g. "κάνω") and only in the masculine gender if it is an adjective (e.g. "καλός") and only in the single form if it is and noun (e.g. "πράγμα").

So in order to make the matching more effective, we used a relatively recent and very effective method, the "Adaptive Fuzzy String Matching" method by Kaufman and Klevs (2021). This method was created to solve the problem of matching two databases based on only one common variable which, however, is incomplete and very different between the two databases, a common problem in the field of data analysis. Therefore, for example, the algorithm is able to decide that the observation "JP Morgan" is shared with the observation "JPM". The way this is achieved is through the recruitment of a series of indicators such as cosine similarity, Jaccard etc. and feeding of them into a machine learning model in order to predict the likelihood that two words match or not. This algorithm is available in R through the "stringmatch" library. In the supplementary material, we provide the code we have used to join the dictionary with the articles we collected, using the Kaufman and Levs method. We need to note that this code takes some time to run due to the many (unique) words that need to be matched, so it takes a lot of resources (CPU and RAM). However, at the supplementary material we provide an excel file named "matched_words", which contains the result of the code.

After the words were matched, we took the average of the word scores for each article. The result was that the articles got an average score on a scale from -3.25 to 3.25.

[5] https://github.com/MKLab-ITI/greek-sentiment-lexicon

[6] To accomplish this, we converted the words of the articles into a tabular format so that each row is also one word (one-word-per-row format).

We then selected a random sample of 10 articles from those that received a very negative score (<-1) for human coding. Of these 10 articles, 5 were coded as populist.

The second method for the detection of populism that we used is the word frequency method. This is a simple method that counts the frequency of occurrence of the words "λαο" and "λαϊκ". This method has been applied by Stavrakakis and Katsambekis to detect populism in the speeches of the former prime minister of Greece, Alexis Tsipras (Stavrakakis & Katsambekis, 2014). Thus, after calculating the frequency of the words "λαο" and "λαϊκ" in the articles we took a random sample of 10 of the articles with the most frequent occurrence of these words (> 5 times). According to human coding, four out of these10 articles, were found to be populist.

It, therefore, appears that both methods have similar results and can be used interchangeably or even complementarily, although sentiment analysis seems to be somewhat more effective (however the difference is not big enough to give a definitive statement). Nevertheless, none of the two methods is able to identify all populist texts, except to possibly limit enough the volume of texts and target the researcher to articles that are more likely to be populist (either sentimentally negative articles, or articles with a high frequency of the words "λαο" and "λαϊκ"). The most effective method, which is able to detect (almost) all of the populist articles seems to be the third one, machine learning, which we discuss below.

MACHINE LEARNING

Machine learning is a relatively new discipline in political science, borrowed from computer science (and specifically from the field of artificial intelligence). The logic is that a computer is trained on a set of data in order to be able to solve some problems by itself. The applications of machine learning are many, e.g., forecasting business sales and profits, targeted advertising, financial fraud detection, and even self-driving cars.

In political science, most applications of machine learning are in the field of natural language processing, i.e. the field of analyzing and processing textual data with the aim of prediction (Cocco & Monechi, 2021; Denny & Spirling, 2018; Grimmer & Stewart, 2013; Kaufman & Klevs, 2021). Its applications in other branches of political science are limited. Notable examples are the innovative work of Schrodt (1990), who uses a machine learning algorithm to predict the outcomes of wars between states. Another example is the use of machine learning to predict election rates and voting results in the United States Supreme Court (Montgomery et al., 2012), as well as its use to detect electoral fraud (Cantú & Saiegh, 2017).

Machine learning is divided into three categories, supervised, unsupervised and reinforcement learning. The first is performed in cases where the algorithm is trained on labeled data in order to predict the categories of unlabeled data. The second is performed in cases where there are no labeled data and the algorithm is left on its own to discover patterns in the data. Cluster analysis is one such example. Finally, the third is used in cases where the algorithm is trained on some initial data, but not corrected, and is left alone to make decisions within some environment seeking to increase its "reward" and limit its "punishment" (Cranmer, 2017).

The two main methods used in machine learning are classification and regression. Classification is a method that predicts into which distinct category each observation will fall. Regression is used to predict continuous variables. Regression in machine learning

doesn't have to rely only on statistical models (such as linear, logistic etc.), but also to deep learning methods i.e. neural networks etc. Therefore, in the language of machine learning, the concept of regression is interpreted in a broader way than in the language of statistics, and generally means the prediction of a continuous result (Cranmer, 2017).

For the detection of populism in this paper, we used supervised machine learning. First we used a random sample of 27 articles which we read and coded as populist (1) or not (0) as a training set. The goal was for the computer to use a statistical model to train itself on the initial data and then classify the rest on its own. The statistical model that we used was logistic regression, as it is most suitable for cases of binary classification. The logistic function is as follows:

$$\varphi(z) = \frac{1}{1 + e^{-z}}$$

The φ function of z takes values in the range from 0 to 1. The above function is also called sigmoid because when visualized on an X/Y plot it forms an S letter. In this function, in the place of z we plug in the linear regression model (that results from the logistic regression i.e. that which maximizes the likelihood function) as follows:

$$p = \frac{1}{1 + e^{-(a+bx)}}$$

In this way, for each value of x we have a probability p that the particular observation belongs to category 1 (rather than category 0). The usual threshold we set for classifying an observation into one of the two categories is 0.5. If the observed value is above this threshold we classify it in category 1, otherwise in category 0.

Based on the above statistical model we wrote an algorithm in R for the automated coding of articles as populist or not, based on the initial, coded by us, 27 articles. As the training set in this case is very small, a machine learning model cannot be used by itself, as such models need a large amount of data to be able to give accurate results. Thus, we created a HITL (Human-In-The-Loop) algorithm (Zanzotto, 2019). HITL algorithms are based on human-machine interaction, i.e. human creates a program so that the computer repeats (loop) an operation, and when it completes this operation it "requests" a correction from human in order to repeat the same operation again with a more effective way etc. This cycle continues in this interactive way until the computer manages to achieve the desired percentage of accuracy. Such algorithms are suitable when there are not enough data to train the computer (such as in our case), and help to achieve the desired accuracy of a rare training set without needing a lot of time (or years) to create a large enough training set[7]. In summary, we developed an HITL algorithm based on a logistic regression statistical model. Below we proceed with the presentation of the algorithm.

THE ALGORITHM

The first step is the entry of the coded data (training set). The second step (and here the loop begins) is to transform them through a function we called "text_preprocessing" so that

[7] https://levity.ai/blog/human-in-the-loop

they are ready to be read by the computer. The transformation involves segmenting the text into words (one-word-per-row) and then converting them into numerical form with the tf-idf method. The tf-idf method (Silge, 2022) is a common method of presenting textual data in which words are weighted based on their rarity. It results as a multiplication of two indicators: tf (term frequency), i.e. the frequency of a word in a document (in this case an article) and idf (inverse document frequency). Idf decreases the weight for words that are very frequent and increases it for words that are less frequent in a collection of documents (in this case articles). The idf formula is as follows:

$$\text{idf(term)} = \ln\left(\frac{\text{ndocuments}}{\text{ndocuments containing term}}\right)$$

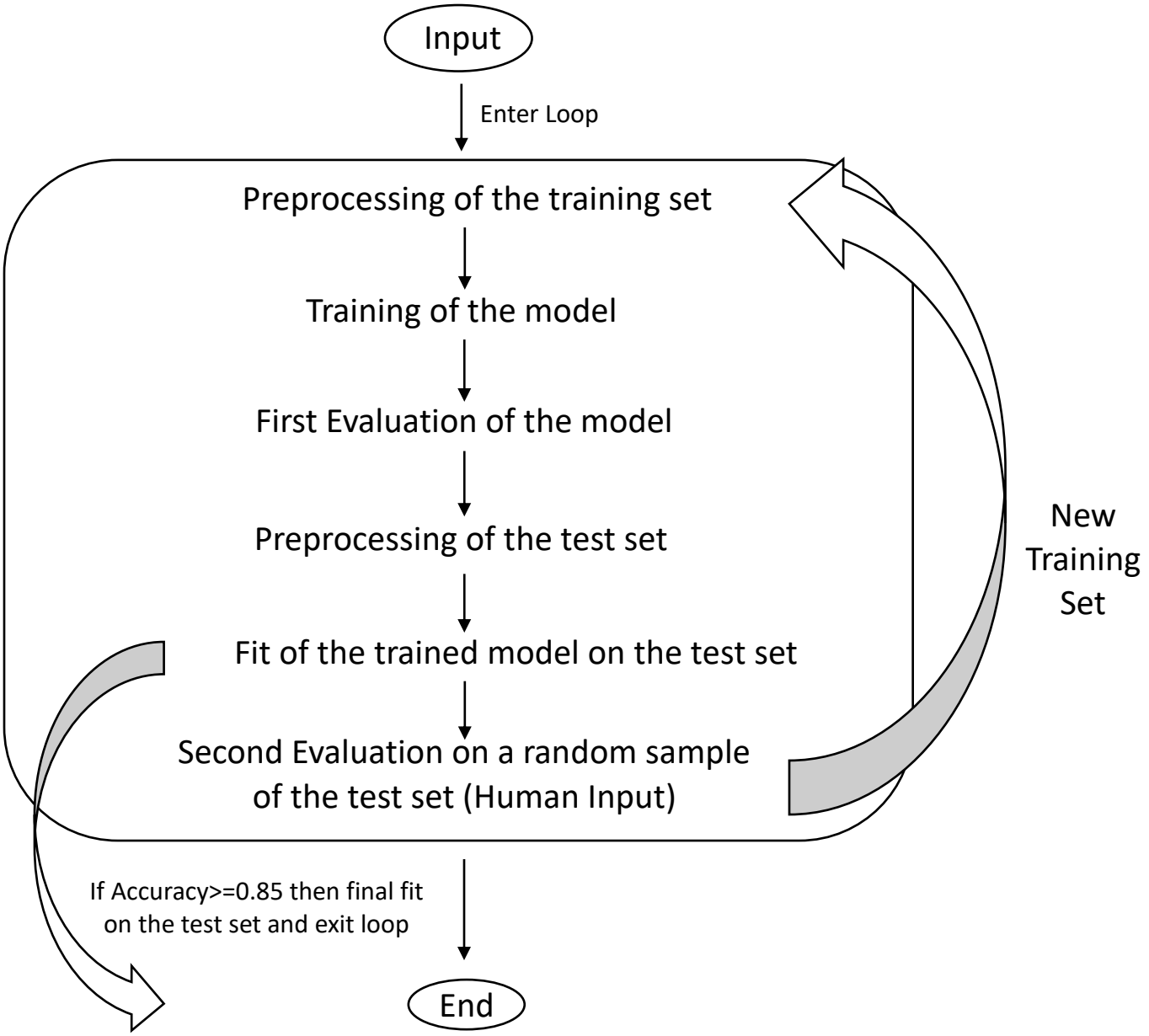
In the third step, the computer is trained using the logistic regression statistical model and the variables used are the website names and the tf-idf variable. In the fourth step, the first evaluation of the model is done using the "cross-validation" method) (Grimmer & Stewart, 2013). Based on this method, the training set is randomly divided into k resampling folds and the model is evaluated on each of them. More specifically, it is trained on all of the rest (k-1) and the predictions are made to the last one (k-th), and the same process is performed for each one of the folds.

Then, in the fifth step, the rest of the data (test set) is prepared (in the same way as described above). In the sixth step, the trained model is applied to the now prepared test set. In the seventh step, the researchers intervene to correct the algorithm manually. At this point, they see in front of them a random sample of 10 articles from the predicted test set and the predicted categories by the computer, read the articles and verify whether they were correctly classified as populist or not and then make the necessary corrections if needed. Then, in the eighth step, this sample of the corrected articles is joined with the training set and thus we have a new training set, larger than the first one. Then the loop starts again from the second step until the eighth and so on. The loop ends when the accuracy rate resulting from cross-validation reaches or exceeds 85%. This number is somewhat arbitrary, but it is a fairly significant percentage. When the loop ends, the final, a bit larger, training set (as it has resulted from all the previous iterations) is applied one last time to the test set and the loop ends. The figure below (Figure 2) visualizes all the aforementioned steps.

After applying the aforementioned machine learning algorithm to the data, we drew a random sample of 10 of the articles classified as populist by the algorithm. Of these, after reading them, 8 were found to be populist. So it seems that machine learning is the best method among the three that we presented in this paper, to detect populism in website articles.

To summarize, the basic logic of the algorithm, as described above, is to classify with the highest possible accuracy a large volume of articles, starting from a very small training set. For this reason, a HITL algorithm is used. In this effort, two evaluations occur consecutively, one by the computer and one by the human, until the loop ends.

Figure 2: HITL Algorithm for the classification of articles.



ANALYSIS OF THE POPULIST ARTICLES

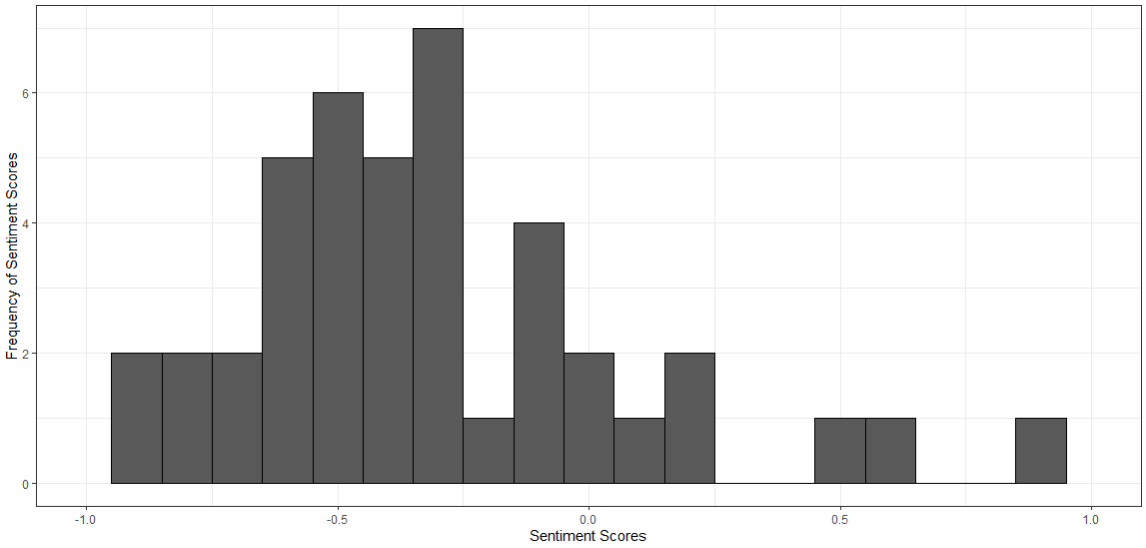
Below we present some analyses we performed on the articles predicted as populist by the algorithm. It should be noted here that the results may be biased, as the initial coding and reading of the articles was based solely on our subjective opinion and perception. This is one of the limitations of machine learning. Since all machine learning algorithms learn only from training data, it is easy for them to reproduce social biases of the researcher developing them (Cranmer, 2017). Especially when it comes to text analysis, the goal of

supervised machine learning is to reproduce human codings automatically (Grimmer & Stewart, 2013).

Nevertheless, there was an effort on our part to be as objective as possible in coding the articles with the methodology of the ideational approach that we described above. However, it would be more correct for the researcher to establish specific coding rules before reading the articles, and then classify them based on these rules before proceeding with the machine learning process. Given this, we proceed below to the presentation of some analyses of the articles predicted as populist by the algorithm.

Firstly, out of a total of 572 articles, 52 were found to be populist, i.e. a percentage of 10%. The following analysis was made on these populist articles. In the first graph (Graph 1) we watch the populist articles' sentiment distribution. We can see that the majority of populist articles have a negative sentiment score. This result seems to confirm the claim we made in the previous section that populism is more likely to be detected in articles characterized by negative sentiments, and also argues that sentiment analysis can be an effective method for detecting populist articles.

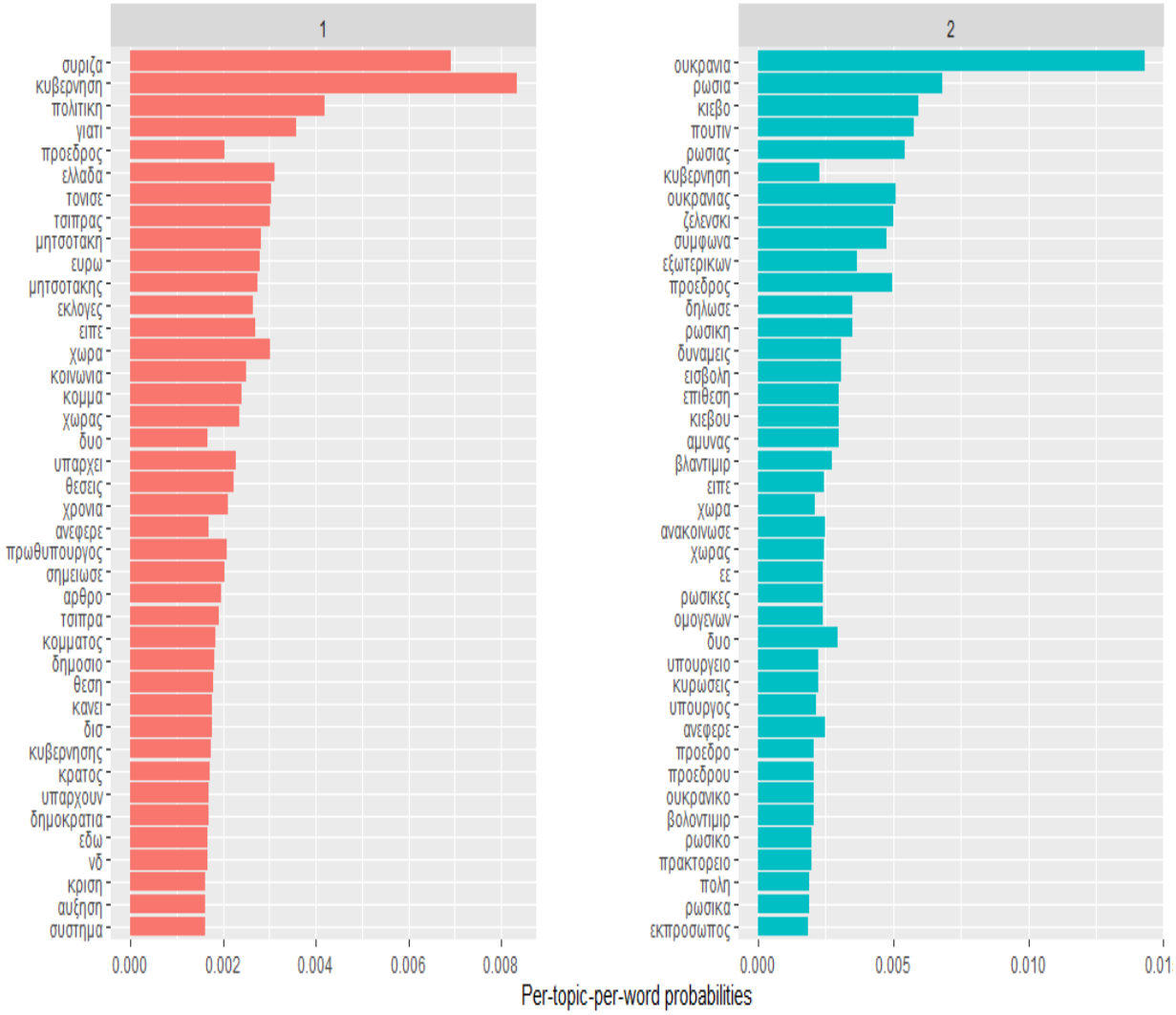
Graph 1: Sentiment distribution of the populist articles



In the next graph (Graph 2) we see two basic topics that emerged from an LDA analysis. LDA is a very frequently used unsupervised machine learning method in the field of text analysis which aims to extract topics in a volume of texts and calculate the probability of each word belonging to either topic (Grimmer & Stewart, 2013). In the graph, we see the 40 words with the highest probability of occurring in each topic. Based on these, we see that in the first topic words like "government", "syriza", "mitsotaki", "party", "politics" etc. have the highest probability of occurring in this topic. Therefore, we can say it's a mainly political topic. In the second topic we see words with the highest probability like "ukraine", "russia", "putin", "ee", "kiev" etc. so we can say that this topic

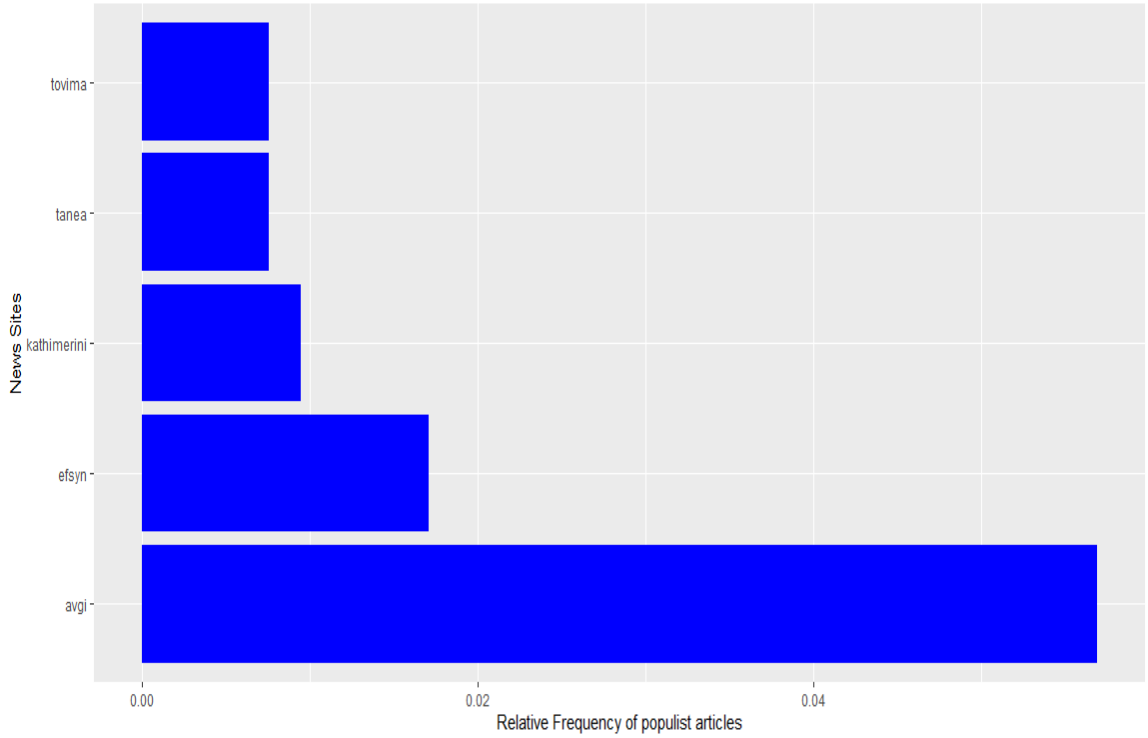
mainly concerns the war in Ukraine which fiercely marked the month of February, when the data were collected.

Graph 2: LDA topics of the populist articles



In Graph 3 we see the relative frequency of populist articles per news site. It seems that “Avgi” has the highest frequency of populist articles followed by “Efsyn”.

Graph 3: Relative Frequency of populist articles per News Site



"Avgi" and "Efsyn" are indeed more populist in content than the other news sites. "Avgi" is a media company owned by the radical left and populist party, "SYRIZA", which also served as Greece's governing party from 2015 to 2019. "Efsyn" is a media company with a left-wing orientation. Although, it doesn't belong to a specific party, "Efsyn" produces populist content as well, deriving from various left parties and movements. On the other hand, the rest of the news sites are mainstream media companies, not well-known about their populist content (in fact, they frequently adopt, especially "kathimerini", a clear anti-populist stance in their opinion articles).

In the next graph (Graph 4), we visualize a word cloud of the 100 most frequent words in the populist articles (colored with darker blue). Among the most frequent words we see the words "government", "syriza", and "ukraine". Also visible, although less frequent is the word "mitsotaki" as well as the words "president" and "politics".

Graph 4: Word cloud of the 100 most frequent words in the populist articles

In conclusion, we can say in general that the populist articles are sentimentally negative, they consist of two main topics, one political and one related to the war in Ukraine and the most frequent words are “government”, “syriza” and “Ukraine”. In general, populism seems to be present in political news articles and political commentary articles, mainly of left-wing political parties and politicians. Thematically those articles concern populist criticisms of the governmental policies by opposition parties, usually in an ostentatious tone, on current issues such as the war in Ukraine, the economic crisis etc. And the usual cause of criticism is the “deception” of the people from the government in order to serve financial interests such as government-friendly media etc. Correspondingly, there is also criticism at the international level for the “imperialistic” stance of the USA, NATO etc. Finally, the news sites where most of the populist articles were found are “Avgi” and “Efsyn”.

As we mentioned at the beginning, the whole analysis was performed on the articles that were common between the two methods (scraping and API), so it now remains to detect the most frequent words of the populist articles in the corresponding Twitter posts of the news sites. In the graph below (Graph 5) we see that the most frequent words in the tweets leading to the corresponding populist website articles are “tsipras” and “syriza”. Other less common words are “kinal”, “nd” and “mitsotakis”. Therefore, the tweets of the news

websites' Twitter accounts leading to populist articles do not seem to have the words "λαο" and "λαϊκ", but mostly names of parties and politicians, possibly because populist speech is mainly driven by them or concerns them. Also, we can see from the above that the populist articles mainly have a political dimension, that is they concern political actors and parties. Therefore, we conclude that the most effective possible way for someone to collect potentially populist tweets from the news websites' Twitter accounts is not so much through the words "λαο" and "λαϊκ", but mainly through searching for politicians' and/or parties' names.

Graph 5: Word cloud of the 100 most frequent words in the tweets of the news websites leading to the corresponding populist website articles

CONCLUSION

In this paper, three computational methods were presented for the effective detection of populism in a set of 572 articles from a sample of six well-known news websites and their corresponding Twitter accounts. The articles were collected using two methods, the common ones between the two were filtered and the analysis was performed on those common ones. The aim was to facilitate, through the use of the computer, those who wish to study populism in a large volume of texts, without having to read and code all the texts,

but only a part of them. These methods were sentiment analysis, word frequency analysis, and machine learning. The first two appeared to be equally effective in terms of their capacity to identify potential populist texts and narrow down a large volume of texts effectively, targeting the researcher to potential populist articles. They can be used alternately or simultaneously. However, they are not capable of identifying all (or most of the) populist articles. So the third method was a HITL machine learning algorithm that we developed, which appeared to be the best of the three, being able to correctly identify the most of the populist articles (always replicating the researcher's logic of what is populist and what is not). This particular algorithm can provide high accuracy as it involves human-computer interaction and double evaluation, one by the computer and one by the human. In this particular application of the algorithm to the data that we collected, the maximum accuracy we reached was 88%. However, researchers can set this threshold even higher and thus read and correct more articles if they have the time and desire to detect all the populist articles in a text corpus. Finally, it was found that populist tweets did not mainly contain the words "λαο" and "λαϊκ", but mainly names of politicians and political parties.

Acknowledgements

The research work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant" (Project Number: 3572).

BIBLIOGRAPHY

Cantú, F., & Saiegh, S. M. (2017). Fraudulent Democracy? An Analysis of Argentina's Infamous Decade Using Supervised Machine Learning. *Political Analysis*, *19*(4), 409–433. https://doi.org/10.1093/pan/mpr033

Cocco, J. D., & Monechi, B. (2021). How Populist are Parties? Measuring Degrees of Populism in Party Manifestos Using Supervised Machine Learning. *Political Analysis*, 1–17. https://doi.org/10.1017/pan.2021.29

Conway, M. (2006). The Subjective Precision of Computers: A Methodological Comparison with Human Coding in Content Analysis. *Journalism & Mass Communication Quarterly*, *83*(1), 186–200. https://doi.org/10.1177/107769900608300112

Cranmer, S. J. (2017). *Introduction to the Virtual Issue: Machine Learning in Political Science*. 9.

Denny, M. J., & Spirling, A. (2018). Text Preprocessing For Unsupervised Learning: Why It Matters, When It Misleads, And What To Do About It. *Political Analysis*, *26*(2), 168–189. https://doi.org/10.1017/pan.2017.44

Grimmer, J., & Stewart, B. M. (2013). Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts. *Political Analysis*, *21*(3), 267–297. https://doi.org/10.1093/pan/mps028

Haq, E. U., Braud, T., Kwon, Y. D., & Hui, P. (2020). A Survey on Computational Politics. *IEEE Access*, *8*, 197379–197406. https://doi.org/10.1109/ACCESS.2020.3034983

Hawkins, K. A., & Rovira Kaltwasser, C. (2017). What the (Ideational) Study of Populism Can Teach Us, and What It Can't. *Swiss Political Science Review*, *23*(4), 526–542. https://doi.org/10.1111/spsr.12281

Hawkins, K. A., & Silva, B. C. (2018). *A Head-to-Head Comparison of Human-Based and Automated Text Analysis for Measuring Populism in 27 Countries*. 42.

Jackman, S. (2006). *Data from Web into R.* The Political Methodologist 14(2). https://thepoliticalmethodologist.files.wordpress.com/2013/09/tpm_v14_n21.pdf

Kaltwasser, C. R., Taggart, P. A., Espejo, P. O., & Ostiguy, P. (2017). *The Oxford Handbook of Populism*. Oxford University Press.

Kaufman, A. R., & Klevs, A. (2021). Adaptive Fuzzy String Matching: How to Merge Datasets with Only One (Messy) Identifying Field. *Political Analysis*, 1–7. https://doi.org/10.1017/pan.2021.38

Kim, S. (2020, September 28). *Computational Models of Political Decision Making*. Oxford Research Encyclopedia of Politics. https://doi.org/10.1093/acrefore/9780190228637.013.881

Lazer, D., Pentland, A., Adamic, L., Aral, S., Barabási, A.-L., Brewer, D., Christakis, N., Contractor, N., Fowler, J., Gutmann, M., Jebara, T., King, G., Macy, M., Roy, D., & Van Alstyne, M. (2009). Computational Social Science. *Science*, *323*(5915), 721–723. https://doi.org/10.1126/science.1167742

Montgomery, J. M., Hollenbach, F. M., & Ward, M. D. (2012). Improving Predictions using Ensemble Bayesian Model Averaging. *Political Analysis*, *20*(3), 271–291. https://doi.org/10.1093/pan/mps002

Mudde, C. (2004). The Populist Zeitgeist. *Government and Opposition*, *39*(4), 541–563. https://doi.org/10.1111/j.1477-7053.2004.00135.x

Mudde, C., & Kaltwasser, C. R. (2017). *Populism: A Very Short Introduction*. Oxford University Press.

Mudde, C., & Rovira Kaltwasser, C. (2018). Studying Populism in Comparative Perspective: Reflections on the Contemporary and Future Research Agenda. *Comparative Political Studies*, *51*(13), 1667–1693. https://doi.org/10.1177/0010414018789490

Ostiguy, P. (2017, October 26). *Populism: A Socio-Cultural Approach*. The Oxford Handbook of Populism. https://doi.org/10.1093/oxfordhb/9780198803560.013.3

Roberts, M. E. (2016). Introduction to the Virtual Issue: Recent Innovations in Text Analysis for Social Science. *Political Analysis*, *24*(V10), 1–5. https://doi.org/10.1017/S1047198700014418

Schrodt, P. A. (1990). Predicting Interstate Conflict Outcomes Using a Bootstrapped ID3 Algorithm. *Political Analysis*, *2*, 31–56. https://doi.org/10.1093/pan/2.1.31

Silge, E. H. and J. (2022). *Supervised Machine Learning for Text Analysis in R*. https://smltar.com/

Stavrakakis, Y., & Katsambekis, G. (2014). Left-wing populism in the European periphery: The case of SYRIZA. *Journal of Political Ideologies*, *19*(2), 119–142. https://doi.org/10.1080/13569317.2014.909266

Theocharis, Y., & Jungherr, A. (2021). Computational Social Science and the Study of Political Communication. *Political Communication*, *38*(1–2), 1–22. https://doi.org/10.1080/10584609.2020.1833121

Tufekci, Z. (2014). Engineering the public: Big data, surveillance and computational politics. *First Monday*. https://doi.org/10.5210/fm.v19i7.4901

Weyland, K. (2017, October 26). *Populism: A political-strategic approach.* The Oxford Handbook of Populism. https://doi.org/10.1093/oxfordhb/9780198803560.013.2

Zanzotto, F. M. (2019). Viewpoint: Human-in-the-loop Artificial Intelligence. *Journal of Artificial Intelligence Research*, *64*, 243–252. https://doi.org/10.1613/jair.1.11345