UNIVERSITY OF CALIFORNIA

Los Angeles

Vox Populism: Analysis of the Anti-Elite Content of Presidential Candidates' Speeches

A thesis submitted in partial satisfaction

of the requirements for the degree

Master of Science in Statistics

by

Caleb Ziolkowski

2021

ABSTRACT OF THE THESIS

Vox Populism: Analysis of the Anti-Elite Content of Presidential Candidates' Speeches

by

Caleb Ziolkowski
Master of Science in Statistics
University of California, Los Angeles, 2021
Professor Yingnian Wu, Chair

Often social scientists want to label whether text is populist or anti-elite in some sense. Traditional methods of content analysis tend to run into one of two problems. Labeling text by hand is taxing, limiting the scope of the analysis. Alternately, labeling text based on political-party affiliation elides variation within political parties and does not tend to work well for two-party systems. I use recent breakthroughs in natural language processing (NLP) combined with supervised learning to explore an alternative way of labeling text as anti-elite that avoids these constraints, allowing sentence-level categorization at scale.

The thesis of Caleb Ziolkowski is approved.

Chad Hazlett

Jeffrey B. Lewis

Yingnian Wu, Committee Chair

University of California, Los Angeles

2021

To my family and friends, who seem to still like me even though I have spent more time with this thesis than them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

Anti-elitism is an important topic in the social sciences. Anti-elitism is central to most studies of populism, a phenomenon present in political movements and parties across the ideological spectrum, as well as in developed and developing nations (Aslanidis, 2016; Bonikowski & Gidron, 2015; Di Tella, 1965; Germani, 1978; Moffitt, 2020; Mudde, 2004; Mudde & Rovira Kaltwasser, 2018; Müller, 2017; Stavrakakis & Katsambekis, 2014). Some argue that anti-elitism (and populism more generally) shares an inextricable connection with democracy—only if people doubt the right or ability of the elite to rule are they likely to demand greater levels of representation (Müller, 2016). Similarly, anti-elitism seems to stoke at least some revolutionary movements—important democratic revolutions, such as the American and French revolutions, had large anti-elite components (K. Hawkins, Read, & Pauwels, 2017). The effect of economic development on democracy is sometimes understood in terms of how democracy can reduce tensions between the elites and the people by making the elites' concessions credible; windows of heightened anti-elite sentiments are likely to correspond to periods of democratization in such a model (Acemoglu & Robinson, 2000, 2006). Anti-elitism seems to be a core part of the backlash against globalization (Rodrik, 2020) and discontent about economic inequality (Lindh & McCall, 2020).

While anti-elitism may be important, studying the subject poses measurement challenges. For example, how can we know if an actor is, in some sense, anti-elite? One approach has been to look at their words—written or spoken. For instance, one study analyzed opinion articles in newspapers in five countries to see how populism—which consists of "people-centrism" and "anti-elitism"—changed from roughly 1990 to 2007 (Rooduijn, 2014). The limitation of this approach is made clear by the amount of data analyzed; for the

five countries four election periods were examined; for each election period, only four weeks of newspapers articles were analyzed; and, for each country, only *opinion articles* in three newspapers were analyzed in each four week period. To undertake this effort, many coders were necessary—which introduces both the issue of inter-coder reliability as well as the cost of labor.

The other widely adopted approach uses party membership to determine if actors—particularly politicians—are anti-elite. This happens most in the context of studies of populism, particularly in proportional representation systems and/or when a party is led by a charismatic, anti-elite leader (Mudde & Kaltwasser, 2012; Von Beyme, 1988). This approach rarely (systematically) probes how politicians within the party vary in terms of anti-elitism. Further, in two-party systems, this approach can be quite problematic. In the US, for instance, there are—and have long been—anti-elite elements on both sides of the political spectrum, within the two major parties. Using party to classify how anti-elite politicians are in this case almost surely results in too much measurement error to be truly useful for most scholarly inquiries.

I explore whether recent breakthroughs in natural language processing (NLP) and supervised learning can be applied to measure the anti-elite content of text at scale. One of the first efforts to map sentences of more or less arbitrary length into high dimensional space (these need not be sentences in the traditional sense, but can continue on past end marks)—known as sentence embedding—was Google's Universal Sentence Encoder (USE) in 2018 (Cer et al., 2018). Since then there have been several sentence encoders that have seen wide adoption in the academy and industry due to their state-of-the-art performance on language tasks; these sentence embeddings can be combined with supervised learning techniques to perform language tasks downstream from the encoding process. This thesis examines if such a method might help with classifying text as anti-elite at a granular level while demanding fewer resources than relying only on human coding.

2 Measuring anti-elitism with sentence embeddings

A recent development in NLP is sentence embeddings. The term sentence embedding refers to a set of techniques where sentences are converted to vectors of numbers. Sentence embeddings differ from previous NLP techniques in important ways. One approach to analyzing text, known as bag of words, simply looks at the count of words in a text, disregarding grammar and word order. Since sentence embeddings analyze sentences, they preserve word order. A more recent approach, word embeddings, represents words in the form of vectors of numbers that encode the meaning of the words so that similar words in the vector space will have similar meanings. This is more closely related to sentence embeddings, in that words are transformed into high-dimensional vectors. Again, a key difference is that sentence embeddings use the order of words in a sentence to help map the sentence into high-dimensional space—usually into vectors of 512 to 1028 real numbers.

One first attempt to create sentence embeddings to be used on downstream linguistic tasks was Google's USE, as mentioned above (Cer et al., 2018)—there were other early attempts of note (Lample, Conneau, Denoyer, & Ranzato, 2017). USE was found "to obtain surprisingly good task performance with remarkably little task specific training data" (Cer et al., 2018, p. 1)—a quality that could reduce the resource burden in a task like labeling text as anti-elite or not. USE maps any given sentence into a vector of 512 numbers. The transformer version uses attention to compute these embeddings in a way that accounts for all the words in the sentence and their order (Vaswani et al., 2017), which is trained through unsupervised learning on Wikipedia, web news, web question-answer pages, and

discussion forums—and augmented with the Stanford Natural Language Inference (SNLI) corpus (Bowman, Angeli, Potts, & Manning, 2015; Cer et al., 2018).

Perhaps the encoder model to attain the most widespread application is Google's Bidirectional Encoder Representations from Transformers (BERT), which obtained state-of-the-art performance on many language tasks (Devlin, Chang, Lee, & Toutanova, 2018). BERT, like previous encoders (A. M. Dai & Le, 2015), uses recurrent neural networks to predict a masked word in a sentence using other words in the sentence. Unlike previous encoders, BERT uses words from both the left and the right to predict the masked word. There are different sizes of BERT producing different lengths of vectors, which reflect the amount of compute and the size of the neural networks used in pretraining the different models.

RoBERTa—short for "A Robustly Optimized BERT Pretraining Approach"—departs little from BERT, as the name suggests. RoBERTa adopts the same architecture as BERT, but more thoroughly explores the possible values selected for hyperparameters and does considerably more pretraining (Liu et al., 2019). The sentence embeddings from this approach obtained state-of-the-art results on several linguistic tasks (Liu et al., 2019).

Also noteworthy is XLNet (Yang et al., 2019). Like previous models, it relies on a transformer architecture with recurrence. While it follows BERT in using all of the words in a sentence (not just those to the left or right) to produce embeddings, it does not rely on masking. XLNet is an auto-regressive model that calculates conditional probabilities for each word in a sentence given the other words (and their locations) in the sentence. That is, its task—rather than recovering a masked word—is to predict each word in a sentence using any combination of the other words in the sentence. XLNet outperforms other models, like BERT, in some downstream language tasks.

While there are other sentence embeddings, this thesis will focus on these four—USE, BERT, RoBERTa, and XLNet—in attempting to label text as anti-elite. The goal of this inquiry is not to find *the best* possible solution, but rather to find one capable of satisfactorily

labeling text as anti-elite.

2.1 Sentence embeddings for downstream tasks

A key draw of sentence embeddings is that these vectors can then be used for language tasks downstream of the pretraining of the embeddings. This is in part why these models have gained renown. For instance, USE was tested on several tasks (Cer et al., 2018): predicting a movie's rating (out of five stars) given the accompanying review (Pang & Lee, 2005); determining customer sentiment from product reviews (Hu & Liu, 2004); and determining the similarity of pairs of sentences scored by Pearson correlation with human performance (STS-B) (Cer, Diab, Agirre, Lopez-Gazpio, & Specia, 2017).

BERT was tested on the General Language Understanding Evaluation (GLUE), which consists of 9 language datasets used to evaluate language understanding. These include, in addition to the STS-B mentioned above, the following:

- Multi-Genre Natural Language Inference (MNLI), a large-scale, crowd-sourced entailment classification task (Williams, Nangia, & Bowman, 2017). The model must predict the second of two sentences affirms, contradicts, or is neutral with respect to the first.

- Quora Question Pairs (QQP), a binary classification task. The model must determine if two Quora questions are semantically equivalent (Chen, Zhang, Zhang, & Zhao, 2018).

- Question Natural Language Inference (QNLI), a version of the Stanford Question Answering Dataset (Rajpurkar, Zhang, Lopyrev, & Liang, 2016) converted to a binary classification task (Wang et al., 2018). The model must determine if a question-answer pair is positive—that is contains the correct answer—or negative—does not contain the correct answer.

- The Stanford Sentiment Treebank (SST-2), a binary single-sentence classification task containing sentences from movie reviews with human codings of reviewer sentiment (Socher et al., 2013).

- The Corpus of Linguistic Acceptability (CoLA), a single-sentence classification task. The model must predict whether a sentence is "acceptable" or not (Warstadt, Singh, & Bowman, 2019).

- Microsoft Research Paraphrase Corpus (MRPC), sentence pairs extracted from online news. Humans have coded whether the pairs of sentences are semantically equivalent (Dolan & Brockett, 2005).

- Recognizing Textual Entailment (RTE), a binary entailment task. The model must predict the second of two sentences affirms, contradicts, or is neutral with respect to the first, but with less training data than MNLI (Bentivogli, Clark, Dagan, & Giampiccolo, 2009).

- Winograd NLI (WNLI), a small natural language inference dataset (Levesque, Davis, & Morgenstern, 2012).

BERT's average score on GLUE was 82.1. Its closest competitor was OpenAI GPT, with a score of 75.1. BERT was also tested on The Stanford Question Answering Dataset (SQuAD v1.1), a collection of 100k crowdsourced question/answer pairs (Rajpurkar et al., 2016). It achieved close to human performance.

RoBERTa and XLNet both outperform BERT on the GLUE and SQuAD tasks. The difference in performance between the RoBERTa and XLNet is small and hard to distinguish (and seemingly differ slightly depending on which paper one consults, likely due to uneven improvements in the performance of the two models over time). They both are near or have exceeded human baselines. While tests like GLUE and SQuAD are valuable for tracking

model performance, I aim to see if any of these sentence embeddings can offer a relatively easy way to classify text as anti-elite to aid social science inquiry.

2.2 Supervised learning

Using these embeddings for downstream tasks can be done with machine learning techniques. With a sufficient number of examples labeled as anti-elite (or not), it is conceptually straightforward to use the vectors of real numbers—that is, sentence embeddings—to predict the label using supervised learning. I implement and compare multiple machine learning algorithms in this thesis.

Several issues must be kept in mind, however. First, overfitting must be avoided. Cross-validation can help with this. Second, the data that I use is imbalanced—there are far fewer "anti-elite" sentences, as I discuss below—and thus the appropriate metrics must be chosen to evaluate the performance of the machine learning models. Further, data augmentation can sometimes help when modeling imbalanced data, which I explore.

2.2.1 Machine learning algorithims

I use a suite of machine learning algorithms to predict whether a sentence is anti-elite given its sentence embedding:

- *Classification decision tree*: a tree model where the outcome variable takes on discreet values. The leaves in the tree represent these discreet values, while the branches are formed by logical conjunctions of the features that predict the label (Breiman, Friedman, Olshen, & Stone, 2017). A *Bagged tree* model extends a decision tree by bagging (bootstrap aggregating), which can lead to performance improvements (Breiman, 1996).

- *k-nearest neighbors (k-NN)*: a non-parametric classification algorithm that uses a plurality vote of the k closest training examples to predict the label (Altman, 1992; Fix & Hodges Jr, 1951).

- *Support-vector machine (SVM)*: a model that maps training data to points in space to maximize the gap between the two categories (Aizerman, Braverman, & Rozonoer, 1964; Boser, Guyon, & Vapnik, 1992). Testing data can be mapped into the same space to apply labels. One implementation of an SVM—which I'll call a *Gaussian SVM*—implements the kernel trick using the radial basis function (Gaussian) to perform a non-linear mapping into space. Another—which I'll call a *Polynomial SVM*—implements the kernel trick over polynomials of the original variable.

- *Multivariate adaptive regression splines (MARS)*: a non-parametric regression technique that, though similar to linear regression, also models nonlinearities and interactions between variables (Friedman, 1991). A *Bagged MARS* model extends the model by bagging (Breiman, 1996). *Flexible discriminant analysis* transforms the response variable to better enable linear separation while generating the discriminant surface using MARS.

- *Naive Bayes classifier (NBC)*: a probabilistic classifier that uses Bayes theorem. This method assumes independence between features and thus is computationally efficient.

2.2.2 Scoring metric

A key concern is how to choose a metric to evaluate the performance of these models, particularly in the context of imbalanced data. Further, the choice of metric matters even more when considering things like crossvalidation and tuning model parameters—discussed below—as the metric determines which of dozens or even hundreds or thousands of models emerge as the most promising.

The essential requirement of any such metric is to adequately summarize the confusion matrix. When the prediction threshold is not fixed, then metrics that can summarize the confusion matrix across the possible thresholds are ideal. For these reasons, it is common for binary classification methods to be evaluated using the receiver operating characteristic (ROC). However, ROC can often deceive with inflated performance estimations, particularly when data is imbalanced (Cao, Chicco, & Hoffman, 2020; Kleinbaum, Klein, & Pryor, 2002; Saito & Rehmsmeier, 2015). The problem is not limited to ROC, as the precision-recall (PR) curve also has issues (Cao, Chicco, & Hoffman, 2020).

The problem is perhaps most easily illustrated if we think about using accuracy, $\frac{\text{correct predictions}}{\text{total predictions}}$, as a metric. When the data is highly imbalanced, let x be the proportion of the more prevalent class, it is trivial to build a predictor that achieves $x \times 100\%$ accuracy; simply predict the majority class every time. If $x > .9$, this predictor achieves over 90% accuracy, though whether it is a good instance of learning seems dubious (Cao, Chicco, & Hoffman, 2020; Saito & Rehmsmeier, 2015).

Likewise, ROC—and the metric that summarizes it, the area under the ROC (AUC)—can give a misleading picture of performance when using imbalanced data. The ROC compares the true positive rate (TPR) and the false positive rate (FPR) across the range of possible threshold values. A high TPR can be achieved by simply predicting many elements as positive. If the data is imbalanced, it is possible for this to lead to seemingly impressive ROCs when in reality the performance on the minority class may be unsatisfactory (Cao, Chicco, & Hoffman, 2020; Kleinbaum, Klein, & Pryor, 2002; Saito & Rehmsmeier, 2015). Thus, the ROC (and AUC) can give overly optimistic evaluations of predictors (Kleinbaum, Klein, & Pryor, 2002).

2.2.2.1 Average precision

Average precision—an alternate to the area under the PR curve (PR AUC) that avoids ambiguity about the value of precision when recall is 0—may perform better on imbalanced data (Jeni, Cohn, & De La Torre, 2013; Saito & Rehmsmeier, 2015). PR plots the TPR, or "recall," against the "precision"—$\frac{\text{True positives}}{\text{True positives+false positives}}$—across the range of the threshold. The average precision—and the PR AUC—involves integrating PR to find the area under the curve, with a score of 1 being perfect. By evaluating the percentage of true positives among positive predictions, PR—as well as PR AUC and average precision—may provide viewers with an accurate prediction of future classification performance.

2.2.2.2 MCC-F1 curve

Average precision, however, still can have its drawbacks. Looking at it's components (recall $= \frac{\text{True positives}}{\text{True positives+false negatives}}$ and precision $= \frac{\text{True positives}}{\text{True positives+false positives}}$), it's obvious that it falls short of summarizing the entire confusion matrix. Specifically, neither recall nor precision contain information on the true negatives (Cao, Chicco, & Hoffman, 2020). Average precision is contingent on which class is labeled "positive," and the more imbalanced the data the more extreme this contingency (Cao, Chicco, & Hoffman, 2020).

A more recent, less widely used, measure is the MCC-F1 curve and its accompanying single-number summarization (MCC-F1 metric) (Cao, Chicco, & Hoffman, 2020). This curve combines two single threshold metrics that both summarize the confusion matrix, calculating them across the range of possible threshold values to produce a curve. MCC is defined as

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

and F1 is the harmonic mean of precision and recall,

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}}.$$

The MCC-F1 curve plots the unit normalized MCC—$\frac{\text{MCC}+1}{2}$—against the F1 score. The MCC-F1 metric summarizes this curve, with the worst possible performance receiving a score of 0 and the best possible performance a score of 1 (Cao, Chicco, & Hoffman, 2020, p. 7). In addition, the MCC-F1 curve can provide an optimal threshold.

Though the MCC-F1 curve seems to perform better than both the AUC and the PR AUC on imbalanced data (Cao, Chicco, & Hoffman, 2020), it is quite new and not widely adopted. Further, measures like the AUC, accuracy, and balanced accuracy tend to enjoy frequent use and have familiar interpretations. I use both the MCC-F1 metric and the PR AUC for hyperparameter tuning and crossvalidation, running the analyses separately. I still calculate and report other popular performance metrics, using the MCC-F1 optimal threshold when doing so for single threshold metrics, such as accuracy.

2.2.3 Bayesian parameter tuning, crossvallidation, and data augmentation

Machine learning algorithms tend to have hyperparameters that must be tuned. While some have relatively few, other algorithms have several. Grid search, though theoretically feasible, can be computationally taxing, particularly given the length of sentence embeddings. A good alternative is to use Bayesian optimization—specifically a Gaussian process (GP)—to tune these parameters (Snoek, Larochelle, & Adams, 2012). I begin with an initial 15 iterations of the model, which the GP uses to learn how the algorithm's performance changes with the hyperparameter values. Tuning is allowed for a further 30 iterations but stops if there is no improvement after 5 iterations.

Relatedly, to avoid overfitting and ensure out-of-sample performance (Ward & Ahlquist, 2018), I perform 5-fold crossvalidation, repeated 5 times. This crossvalidation is done on the training data. It is done for each iteration of the Bayesian optimization, with the evaluation of performance based on the out-of-sample fold results. The mean of these out-of-sample fold results across iterations is used to evaluate the models—and of course, choose hyperparameters. As noted above, it is either the average precision or the MCC-F1 metric that is used for performance evaluation.

Data augmentation can sometimes help performance on imbalanced data. In addition to running the algorithms with no data augmentation or subsampling, I try several methods of data augmentation to see how they impact the results: ROSE (Menardi & Torelli, 2014); ADASYN (He, Bai, Garcia, & Li, 2008); SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002); Borderline-smote (Han, Wang, & Mao, 2005); down-sampling—randomly removing rows of the prevalent class till the classes are equal; removing majority class instances that are close to one another based on k-NN (Mani & Zhang, 2003); removing majority class instances using tomek links (Tomek & others, 1976); and upsampling from the minority class until the classes are equal.

2.3 Speeches of presidential candidates

The data in this thesis comes from candidate speeches for the 2016 presidential election in the United States (K. A. Hawkins, 2016). This election provides the chance to explore the prevalence of anti-elite sentiment for both Republican and Democrat candidates. Relative to most presidential races in the United States, the primary competitions were drawn out and there were fairly large fields of primary candidates for both parties. Further, notable candidates in both parties—Bernie Sanders and Donald Trump—were labeled as populist and, by extension, anti-elite—in popular discourse. This data thus allows an opportunity to test if this conventional wisdom can be confirmed (and quantified) using natural language

Table 2.1: Details of speeches.

Candidate	Speeches	Number of sentences
Carson	1	58
Clinton	22	2081
Cruz	3	195
Kasich	2	116
Rubio	4	127
Sanders	5	199
Trump	21	2376
All	58	5152

processing and machine learning. The data consists of 58 speeches by various candidates. Table 2.1 shows some of the details of the data. When running analysis and generating predictions, I remove any text that is exactly duplicated; my assumption—based on having read through several speeches—is that this removes transcriptions of audience responses (e.g. "Applause"). For analysis and predictions, the total number of sentences is 4404.

2.3.1 Training data

I coded 1049 sentences by hand—where I used the speech transcribers' paragraph breaks to designate sentences—labeling them as anti-elite or not. Since my goal is to test if NLP and ML offer an *efficient* way to label sentences, choosing the transcribers' paragraph breaks offers an easy way to partition the data automatically. I selected these sentences from speeches by Bernie Sanders, Hilary Clinton, and Donald Trump. I did this in part because they had the most speeches in the data. I also was particularly interested in Donald Trump and Bernie Sanders as both were labeled as populists by parts of the mainstream media and academia. I believed they may offer more anti-elite sentences, thereby hopefully ameliorating, if only somewhat, the issue of class imbalance. When I remove duplicates (again, primarily transcriptions of audience responses like "Applause" or "Laughter"), I'm left with 961 sentences.

Table 2.2: Anti-elite results from hand coding.

Candidate	Anti-elite proportion	Number of sentences
Clinton	0.059	438
Sanders	0.201	199
Trump	0.109	412
All	0.106	1049

Here are examples of the sentences, which though clearly sometimes longer than a sentence, tend to be focused on a single theme—which probably is why the transcribers chose these as paragraphs.

- Example 1: Candidate Sanders said, "Tonight, we served notice to the political and economic establishment of this country that the American people will not continue to accept a corrupt campaign finance system that is undermining American democracy, and we will not accept a rigged economy in which ordinary Americans work longer hours for lower wages, while almost all new income and wealth goes to the top 1%."
- Example 2: Candidate Clinton said, "Instead of an economy built by every American, for every American, we were told that if we let those at the top pay lower taxes and bend the rules, their success would trickle down to everyone else."
- Example 3: Candidate Trump said, "So I've watched the politicians. I've dealt with them all my life. If you can't make a good deal with a politician, then there's something wrong with you. You're certainly not very good. And that's what we have representing us. They will never make America great again. They don't even have a chance. They're controlled fully— they're controlled fully by the lobbyists, by the donors, and by the special interests, fully."

I classified all of these examples as anti-elite. And Table 2.2 shows the proportion of anti-elite sentences for the hand-coded sample. These sentences provide testing and training data, with 20% left for testing.

Table 2.3: Examples of embeddings: first eight of 512 digits.

candidate	X1	X2	X3	X4	X5	X6	X7	X8
Sanders	0.0067	0.0343	-0.0038	0.0530	-0.0723	0.0145	0.0791	0.0309
Clinton	0.0436	0.0513	-0.0427	0.0163	-0.0698	0.0622	0.0819	0.0618
Trump	0.0082	0.0196	-0.0227	-0.0028	-0.0808	0.0101	0.0965	0.0558

I take the sentences that have been hand-coded and get embeddings, tokenizing the sentences as required by the encoder models. As mentioned above, I get sentence embeddings from USE, BERT, RoBERTa, and XLNet. I use the transformers version of USE. For Bert, I use both encodings for the cased (BBC) and uncased (BBU) base models (cased models distinguish between capitalized and lower-case letters). For RoBERTa, I use the base cased model (RBC). XLNet does not automatically provide a single vector of embeddings for a sentence, so I use max-pooling (XLN.x) and mean-pooling (XLN.m)—separately—to get general sentence embeddings appropriate for downstream tasks. To show what sentence embeddings look like, the first eight numbers for the USE embeddings for these examples are shown in Table 2.3.

3 Results

For each set of embeddings, I perform crossvalidation and Bayesian hyperparameter tuning twice for each model (once using the MCC-F1 metric to evaluate performance, then the other using average precision). For XLNet, I do this for both the max-pooling embeddings and the mean-pooling embeddings. Further, I do this with no data augmentation and then once for each of the 8 data augmentation methods listed in Section 2.2.3. This process is iterated

across all 9 of the machine learning algorithms listed in Section 2.2.1. I thus attempt to run 972 models, each with 5 rounds of 5-fold crossvalidation and up to 45 iterations of Bayesian hyperparameter tuning. In reality, some data augmentation methods do not always work with each machine learning algorithm, so the actual number of models that successfully ran is 863; again, since I hope to find a satisfactory model for labeling sentences, not the absolute best model, the failure of some models need not threaten the enterprise. The proof will be in the results of the models that ran.

3.1 Top-performing models

I order the results by the ascending mean log loss for each model achieved across the 5 iterations of the 5-fold crossvalidation for the best performing hyperparameters resulting from the Bayesian tuning. Across all the iterations of crossvalidation—for the optimized hyperparameters—I calculated the mean MCC-F1 metric, the mean average precision, the mean AUC, the mean PR AUC, and the mean log loss. All of these metrics are independent of any choice of threshold, and thus are useful in assessing models without specifying a threshold. I then examine which of these evaluation metrics is most consistent in predicting training data *and* testing data. That is, I calculate the Pearson correlation coefficient between `test performance`—each model's mean score for each of these 5 metrics when fit on the test data—and the `training performance`—each model's score for each of these five metrics on the 5 crossvalidation iterations when hyperparameters have been optimized. The intuition is that a defensible way to choose the best performing model is to look at the one that performed the best on the *training data*—using the metric that is most consistent predictor of performance across *testing and training data*—in order to avoid simply picking the model that happened to perform best—possibly simply "got lucky"—on the testing data. Table 3.1 shows that the log loss metric is the most consistent across testing and training data. Table 3.2 shows the results for the top 35 models, ranked according to the mean cross-

Table 3.1: Pearson correlation coefficents across testing and training data.

Average precision	MCC-F1	AUC	PR AUC	Log loss
0.908	0.836	0.847	0.863	0.97

validation log loss. A few trends stand out. USE and RBC embeddings tend to perform the best. BERT embeddings do not break the top 35. The dominance of Polynomial SVM and Gaussian SVM is quite striking, taking all but 2 of the top 35 spots. Data augmentation does not appear to offer major benefits, with b-smote and tomek both performing about the same as similar models with no data augmentation. It's not clear whether MCC-F1 or average precision is more useful as a metric for Bayesian hyperparameter tuning. Finally, metrics typically reported in similar types of studies (e.g. Çinar, Stokes, & Uribe, 2020; Y. Dai, 2018), AUC and accuracy, look quite good for the testing data The mean AUC for the top 35 is 0.93 and the mean accuracy is 0.94. AUC does not tend to drop much from the crossvalidation on the training data to the testing data. A naive strategy of simply guessing the majority class would result in an accuracy of 0.89, an MCC-F1 of 0.21, an average precision of 0.11, and a balanced accuracy of 0.5 on the entire training sample. The strategy adopted in this thesis has produced several models that do well predicting anti-elite sentiment on held-out data, particularly compared with a naive strategy that achieves fairly high accuracy on unbalanced datasets.

Table 3.2: The 35 top-performing models.

Emb.	ML algo	Data aug.	Eval. met.	Rank	AP	AP-CV	MCCF1	MCCF1-CV	LL	LL-CV	AUC	AUC-CV	ACC	Bal. ACC
RBC	Poly SVM	none	MCCF1	1	0.66	0.78	0.51	0.54	0.20	0.17	0.95	0.95	0.93	0.82
USE	Poly SVM	tomek	AP	2	0.81	0.77	0.53	0.53	0.16	0.18	0.94	0.94	0.96	0.84
USE	Poly SVM	none	MCCF1	3	0.82	0.76	0.53	0.53	0.15	0.18	0.96	0.94	0.95	0.83
USE	Poly SVM	none	AP	4	0.82	0.76	0.53	0.53	0.15	0.18	0.96	0.94	0.95	0.83
USE	Poly SVM	tomek	MCCF1	5	0.81	0.78	0.52	0.53	0.16	0.18	0.95	0.94	0.95	0.83
RBC	Poly SVM	none	AP	6	0.64	0.78	0.51	0.53	0.24	0.18	0.94	0.95	0.91	0.85
RBC	Gaus. SVM	none	MCCF1	7	0.63	0.79	0.51	0.54	0.24	0.18	0.95	0.95	0.91	0.87
RBC	Gaus. SVM	bsmote	MCCF1	8	0.62	0.77	0.52	0.53	0.23	0.18	0.94	0.94	0.90	0.90
RBC	Gaus. SVM	bsmote	MCCF1	9	0.65	0.78	0.51	0.53	0.24	0.18	0.94	0.94	0.90	0.88
RBC	Gaus. SVM	bsmote	AP	10	0.65	0.78	0.51	0.53	0.24	0.18	0.94	0.94	0.93	0.82
USE	Gaus. SVM	none	AP	11	0.82	0.77	0.54	0.53	0.16	0.18	0.95	0.94	0.96	0.84
RBC	Gaus. SVM	none	AP	12	0.64	0.78	0.51	0.53	0.24	0.19	0.94	0.95	0.93	0.82
XLN.m	Gaus. SVM	none	AP	13	0.66	0.77	0.52	0.52	0.24	0.19	0.87	0.91	0.94	0.81
XLN.m	Gaus. SVM	none	MCCF1	14	0.66	0.77	0.52	0.52	0.24	0.19	0.87	0.91	0.94	0.81
RBC	Poly SVM	tomek	AP	15	0.65	0.77	0.52	0.53	0.25	0.19	0.94	0.95	0.93	0.86
XLN.m	Gaus. SVM	tomek	AP	16	0.67	0.77	0.52	0.52	0.25	0.19	0.87	0.92	0.93	0.84
USE	Gaus. SVM	none	MCCF1	17	0.83	0.76	0.54	0.53	0.16	0.19	0.95	0.94	0.96	0.84
XLN.m	Poly SVM	tomek	AP	18	0.69	0.75	0.50	0.52	0.21	0.19	0.90	0.93	0.91	0.83
XLN.m	Poly SVM	tomek	MCCF1	19	0.69	0.75	0.50	0.52	0.21	0.19	0.90	0.93	0.91	0.83
USE	Poly SVM	bsmote	MCCF1	20	0.81	0.77	0.53	0.53	0.18	0.19	0.94	0.95	0.95	0.82
RBC	Poly SVM	bsmote	AP	21	0.66	0.76	0.51	0.53	0.25	0.19	0.94	0.94	0.92	0.84
XLN.m	Poly SVM	none	MCCF1	22	0.69	0.74	0.51	0.52	0.21	0.19	0.91	0.93	0.93	0.82
XLN.m	Poly SVM	none	AP	23	0.70	0.73	0.51	0.51	0.21	0.20	0.91	0.93	0.93	0.80
RBC	Poly SVM	tomek	MCCF1	24	0.67	0.77	0.52	0.53	0.25	0.20	0.94	0.95	0.92	0.87
USE	Poly SVM	bsmote	AP	25	0.82	0.76	0.53	0.53	0.18	0.20	0.94	0.95	0.95	0.82
USE	Gaus. SVM	bsmote	MCCF1	26	0.81	0.76	0.53	0.53	0.19	0.20	0.94	0.95	0.95	0.82
USE	Gaus. SVM	bsmote	AP	27	0.81	0.76	0.53	0.53	0.19	0.20	0.94	0.95	0.95	0.82
RBC	Poly SVM	bsmote	MCCF1	28	0.65	0.75	0.51	0.53	0.25	0.20	0.94	0.94	0.93	0.82
XLN.x	Gaus. SVM	none	MCCF1	29	0.61	0.71	0.50	0.50	0.24	0.20	0.88	0.91	0.94	0.79
XLN.x	Gaus. SVM	none	AP	30	0.61	0.71	0.50	0.50	0.24	0.20	0.88	0.91	0.94	0.79
USE	Gaus. SVM	upsample	AP	31	0.83	0.75	0.54	0.52	0.18	0.20	0.94	0.95	0.96	0.84
USE	k-NN	none	AP	32	0.80	0.74	0.61	0.58	0.18	0.20	0.95	0.95	0.96	0.88
USE	k-NN	tomek	AP	33	0.82	0.76	0.61	0.59	0.18	0.21	0.95	0.95	0.95	0.85
XLN.x	Poly SVM	tomek	AP	34	0.60	0.68	0.50	0.51	0.23	0.21	0.89	0.91	0.93	0.80
USE	Gaus. SVM	smote	AP	35	0.81	0.75	0.54	0.53	0.20	0.22	0.93	0.95	0.96	0.82

[a] XLN.m = XLNet mean pooling; XLN.x = XLNet max pooling; AP = Average pecision; "-CV" = Mean crossvalidation score; LL = Log loss; ACC = Accuracy

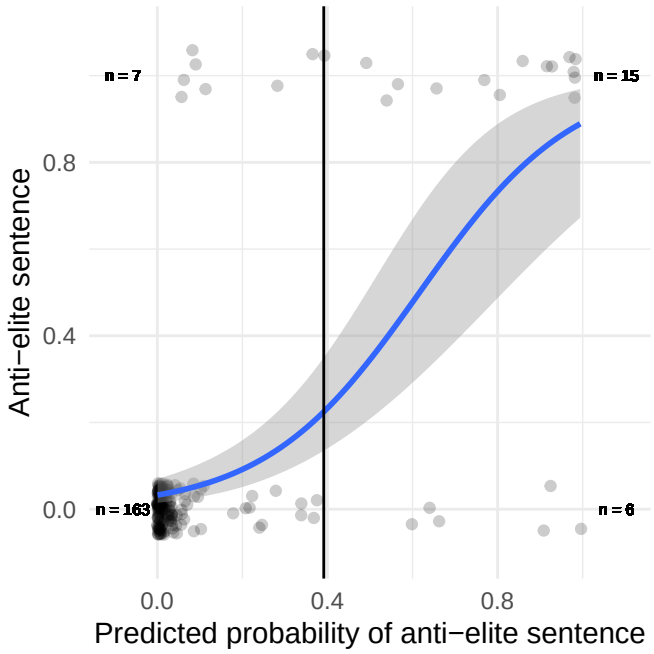

Figure 3.1: **Best model's test-data performance.** Includes fitted line and confidence intervals from logit regression. Vertical line represents the optimal threshold from MCC-F1 curve and the confusion matrix is based on this threshold.

I examine the top model from 3.2 a bit more closely. This polynomial SVM model used the embeddings from RoBERTa with no data augmentation. The Bayesian optimized hyperparameters were a cost of 24.9, a third-degree polynomial, and a scale factor of 1.4e−05. Figure 3.1 displays the performance of this model on the test data. The figure includes a confusion matrix—with the threshold provided by the MCC-F1 curve. The figure also displays the gradations in the model's predictions along the x-axis. A fitted line and confidence intervals from a logit regression show that as the predicted probability of a sentence being anti-elite increases, it is increasingly likely that I coded the sentences as anti-elite.

3.2 Predicting anti-elitism for all data

I take the top model from 3.2 and use it to predict all the data, including the data that doesn't have hand-coded labels. I fit the model to the hand-coded sample, maintaining the hyperparameters and workflow from the crossvalidation and Bayesian hyperparameter

optimization. I then predict the probability a sentence is anti-elite for all the data. Using the optimal threshold from the MCC-F1 score from above, this results in 406 predictions of anti-elite sentences (9.2%).

I randomly select one example of an anti-elite sentence for each candidate that I reproduce below (truncated to 500 characters):

- Candidate Carson: "And I'm saying to people around this nation right now, stop being loyal to a party or to a man and use your brain to think for yourself. That is really the key to us as a nation becoming successful again. Not allowing ourselves to be manipulated by people who think that they are the kingmakers, who think that they are the rulers of thought. They are not the rulers of thought. We, the people are the rulers of thought in this nation. We get to determine what kind of nation we have; other people..."
- Candidate Clinton: "First, let's start with protecting taxpayers and making sure we have more fairness in the system. It is wrong that corporations and the super wealthy play by a different set of rules. A Wall Street money manager should not be able to pay a lower tax rate than a teacher or a nurse."
- Candidate Cruz: "We will win by speaking the truth with a smile. The truth is the American people are fed up with Washington, with a corrupt class that enriches itself and leaves behind the working men and women of this country. The truth is the American people are fed up with a federal government that views the Bill of Rights and the Constitution as a rough draft, as an inconvenience, as something to be set aside, and we are ready to get back to the vision of the framers, of protecting the unalienable rights..."
- Candidate Kasich: "And if we are a neighbor, that means that widow who was married for 50 years who no one calls anymore, you want to change the world? You take her to dinner on Saturday night. She'll wear that dress she hadn't worn in six months. I trust you to do it. You see, what I learned as a boy, what I learned from my mother and

Table 3.3: Mean of anti-elite predictions.

Candidate	Mean probability of anti-elite sentence
Carson	0.176
Clinton	0.069
Cruz	0.065
Kasich	0.057
Rubio	0.059
Sanders	0.240
Trump	0.129

father is that the spirit of America rests in us. It doesn't rest in a big-time politician, the big wigs. Look, you hire us to go do the job, plain and simple, to create an envir. . . "

- Candidate Rubio: "If we remember – if we remember that the family, not the government, is the most important institution in our society. . . "
- Candidate Sanders: "Addressing Wealth and Income Inequality: This campaign is going to send a message to the billionaire class. And that is: you can't have it all. You can't get huge tax breaks while children in this country go hungry. You can't continue sending our jobs to China while millions are looking for work. You can't hide your profits in the Cayman Islands and other tax havens, while there are massive unmet needs on every corner of this nation. Your greed has got to end. You cannot take advantage of all. . . "
- Candidate Trump: "Their financial resources are unlimited. Their political resources are unlimited. Their media resources are unlimited. And, most importantly, the depths of their immorality is unlimited."

For the most part, these sentences seem to contain elements of anti-elitism. The Rubio and especially Kasich examples may not be best classified as anti-elite—though even in those there are some elements of anti-elitism (Rubio downplaying the importance of government, Kasich subtly disparaging the role of "big-time politicians" and "big wigs"). The other examples have quite strong anti-elite meanings, both direct and through connotation.

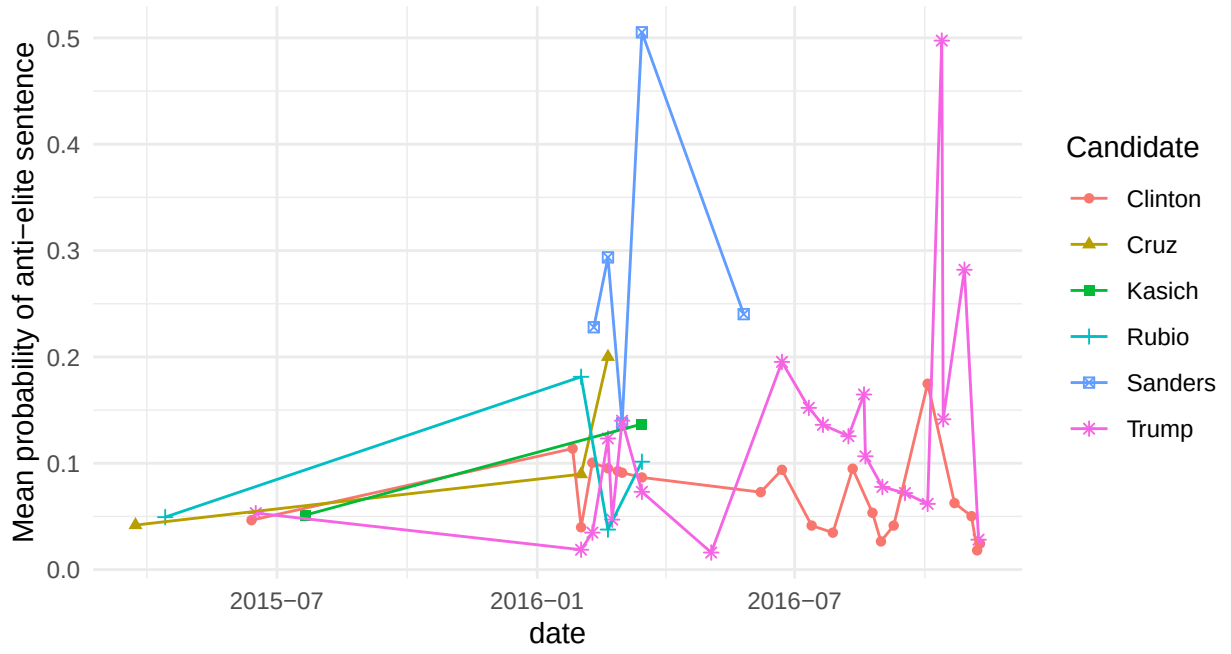

Figure 3.2: **Change in anti-elite sentences over time.** Plots the mean probability sentences spoken by a candidate will be anti-elite over time. Carson is removed since he only had a single speech on April 15, 2015—for which the mean probability of an anti-elite sentence was 0.176.

We can summarize the predictions by candidate as well. Table 3.3 shows the mean predicted probability that sentences are anti-elite. This mostly seems to bolster popular perceptions that Trump—about double the anti-elite probability of most—and Sanders—nearly 4 times the anti-elite probability of most others—were populists. It is somewhat surprising, perhaps, that Carson is so high and that Cruz isn't higher, but it must also be remembered that the sample of speeches is smaller for these candidates.

Lastly, I explore how the best model predicts anti-elite sentences change over time, by candidate. Figure 3.2 shows a good deal of variation. All candidates tend to exhibit changes over time, but it is not uniformly in one direction. Perhaps most surprisingly, Trump does not appear to be any more anti-elite than his Republican primary opponents when running against them, though the observations during this period are sparse. Intriguingly, Trump and Clinton's trend lines appear to mimic one another once the two of them are competing head-to-head. Exploring this variation—and any relationship between the anti-elitism of a

candidate and her most obvious opponent—represents just one possible application of the method I've employed in this thesis.

4 Conclusion

This thesis has explored whether natural language processing and machine learning offer a viable way to measure the anti-elite content of text. I've utilized sentence embeddings on speeches from the 2016 US presidential election, finding that RoBERTa sentence embeddings perform the best—though the Universal Sentence Encoder was also quite good. On widely used metrics the best performing model attained high scores on the held-out test data, 0.95 for the area under the receiver operator curve and 0.93 accuracy. Mindful that imbalanced data—like that used in this paper—can result in overly optimistic estimates for these metrics, I used more appropriate metrics like average precision, the MCC-F1 metric, and log loss to optimize hyperparameters and rank the models. The results are promising and suggest the method developed here can be used to accurately characterize the anti-elite content of text at scale—and at the granular level of the sentence—while minimizing the amount of coding to be done by hand.

5 References

Acemoglu, D., & Robinson, J. A. (2000). Why did the west extend the franchise? Democracy, inequality, and growth in historical perspective. *The Quarterly Journal of Economics*, *115*(4), 1167–1199.

Acemoglu, D., & Robinson, J. A. (2006). *Economic origins of dictatorship and democracy*. Cambridge University Press.

Aizerman, M. A., Braverman, E. M., & Rozonoer, L. I. (1964). Theoretical foundations of potential function method in pattern recognition. *Automation and Remote Control*, *25*(6), 917–936.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, *46*(3), 175–185.

Aslanidis, P. (2016). Is populism an ideology? A refutation and a new perspective. *Political Studies*, *64*(1_suppl), 88–104.

Bentivogli, L., Clark, P., Dagan, I., & Giampiccolo, D. (2009). The fifth PASCAL recognizing textual entailment challenge. In *TAC*.

Bonikowski, B., & Gidron, N. (2015). Populism in legislative discourse: Evidence from the european parliament. *Unpublished Manuscript, Harvard University*.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on computational learning theory* (pp. 144–152).

Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv Preprint arXiv:1508.05326*.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). *Classification and regression trees*. Routledge.

Cao, C., Chicco, D., & Hoffman, M. M. (2020). The MCC-F1 curve: A performance evaluation technique for binary classification. *arXiv Preprint arXiv:2006.11278*.

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., & Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv Preprint arXiv:1708.00055*.

Cer, D., Yang, Y., Kong, S., Hua, N., Limtiaco, N., John, R. S., ... others. (2018). Universal sentence encoder. *arXiv Preprint arXiv:1803.11175*.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chen, Z., Zhang, H., Zhang, X., & Zhao, L. (2018). Quora question pairs. *URL Https://Www. Kaggle. Com/c/Quora-Question-Pairs*.

Çinar, I., Stokes, S., & Uribe, A. (2020). Presidential rhetoric and populism. *Presidential Studies Quarterly*, *50*(2), 240–263.

Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. *Advances in Neural Information Processing Systems*, *28*, 3079–3087.

Dai, Y. (2018). Measuring populism in context: A supervised approach with word embedding models. Pennsylvania State University.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep

bidirectional transformers for language understanding. *arXiv Preprint arXiv:1810.04805*.

Di Tella, T. S. (1965). *Populism and reform in latin america*.

Dolan, W. B., & Brockett, C. (2005). Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the third international workshop on paraphrasing (IWP2005)*.

Fix, E., & Hodges Jr, J. (1951). *Discriminatory analysis-nonparametric discrimination: Consistency properties*. CALIFORNIA UNIV BERKELEY.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, 1–67.

Germani, G. (1978). *Authoritarianism, fascism, and national populism*. Transaction Publishers.

Han, H., Wang, W.-Y., & Mao, B.-H. (2005). Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing* (pp. 878–887). Springer.

Hawkins, K. A. (2016). United states 2016 presidential campaign speeches dataset. *Team Populism*. BYU. Retrieved from https://populism.byu.edu/Pages/Data

Hawkins, K., Read, M., & Pauwels, T. (2017). Populism and its causes. *The Oxford Handbook of Populism*, 267–286.

He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)* (pp. 1322–1328). IEEE.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 168–177).

Jeni, L. A., Cohn, J. F., & De La Torre, F. (2013). Facing imbalanced data–recommendations for the use of performance metrics. In *2013 humaine association conference on affective computing and intelligent interaction* (pp. 245–251). IEEE.

Kleinbaum, D. G., Klein, M., & Pryor, E. R. (2002). Logistic regression: A self-learning text.

Lample, G., Conneau, A., Denoyer, L., & Ranzato, M. (2017). Unsupervised machine translation using monolingual corpora only. *arXiv Preprint arXiv:1711.00043*.

Levesque, H., Davis, E., & Morgenstern, L. (2012). The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Lindh, A., & McCall, L. (2020). Class position and political opinion in rich democracies. *Annual Review of Sociology*, *46*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., ... Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv Preprint arXiv:1907.11692*.

Mani, I., & Zhang, I. (2003). kNN approach to unbalanced data distributions: A case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets* (Vol. 126). ICML United States.

Menardi, G., & Torelli, N. (2014). Training and assessing classification rules with imbalanced data. *Data Mining and Knowledge Discovery*, *28*(1), 92–122.

Moffitt, B. (2020). *The global rise of populism*. Stanford University Press.

Mudde, C. (2004). The populist zeitgeist. *Government and Opposition*, *39*(4), 541–563.

Mudde, C., & Kaltwasser, C. R. (2012). *Populism in europe and the americas: Threat or corrective for democracy?* Cambridge 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.

Müller, J.-W. (2016). *What is populism?* University of Pennsylvania Press.

Müller, J.-W. (2017). Populism and constitutionalism. In *The oxford handbook of populism*.

Pang, B., & Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. *arXiv Preprint Cs/0506075*.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). Squad: 100,000+ questions for machine comprehension of text. *arXiv Preprint arXiv:1606.05250*.

Rodrik, D. (2020). *Why does globalization fuel populism? Economics, culture, and the rise of right-wing populism*. National Bureau of Economic Research.

Rooduijn, M. (2014). The mesmerising message: The diffusion of populism in public debates in western european media. *Political Studies*, *62*(4), 726–744.

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS One*, *10*(3), e0118432.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, *25*.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing* (pp. 1631–1642).

Stavrakakis, Y., & Katsambekis, G. (2014). Left-wing populism in the european periphery: The case of SYRIZA. *Journal of Political Ideologies*, *19*(2), 119–142.

Tomek, I., & others. (1976). Two modifications of CNN.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *CoRR*, *abs/1706.03762*. Retrieved from http://arxiv.org/abs/1706.03762

Von Beyme, K. (1988). Right-wing extremism in post-war europe. *West European Politics*, *11*(2), 1–18.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv Preprint arXiv:1804.07461*.

Ward, M. D., & Ahlquist, J. S. (2018). *Maximum likelihood for social science: Strategies for analysis*. Cambridge University Press.

Warstadt, A., Singh, A., & Bowman, S. R. (2019). Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, *7*, 625–641.

Williams, A., Nangia, N., & Bowman, S. R. (2017). A broad-coverage challenge corpus for sentence understanding through inference. *arXiv Preprint arXiv:1704.05426*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, *32*.