Globalization and Border Securitization in International Discourse
Appendices

Appendix A

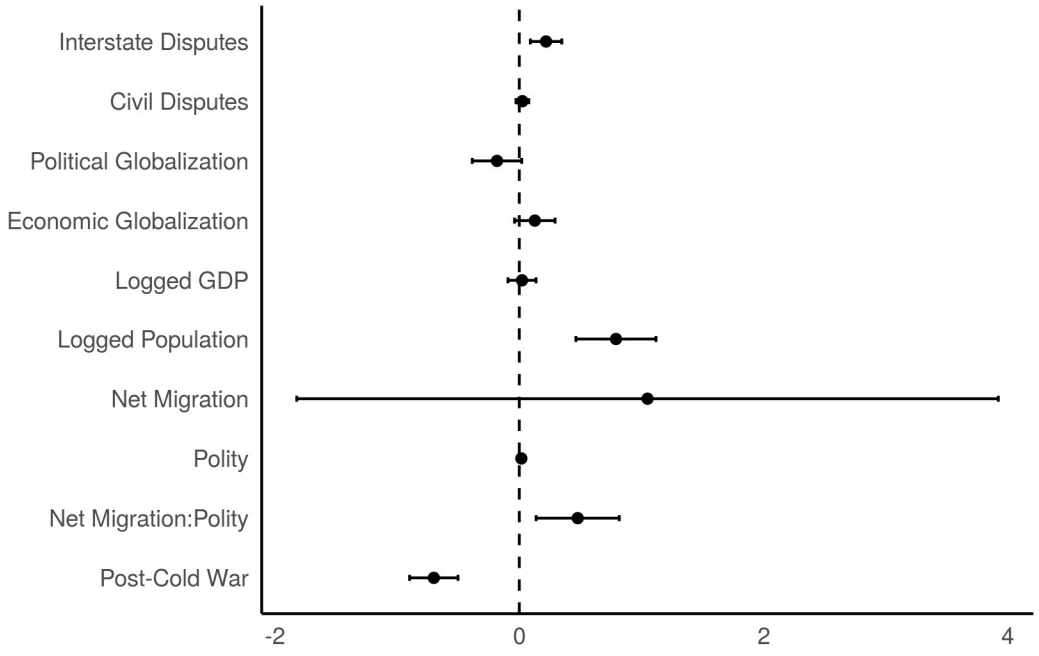

Figure A.1: Correlates of Border Mentions. *The dependent variable is a dichotomous indicator of whether or not a country included a border-relevant mention in their annual address to the UNGA. Explanatory variables are speaker characteristics. Results of a logit model with country fixed effects.*

Appendix B

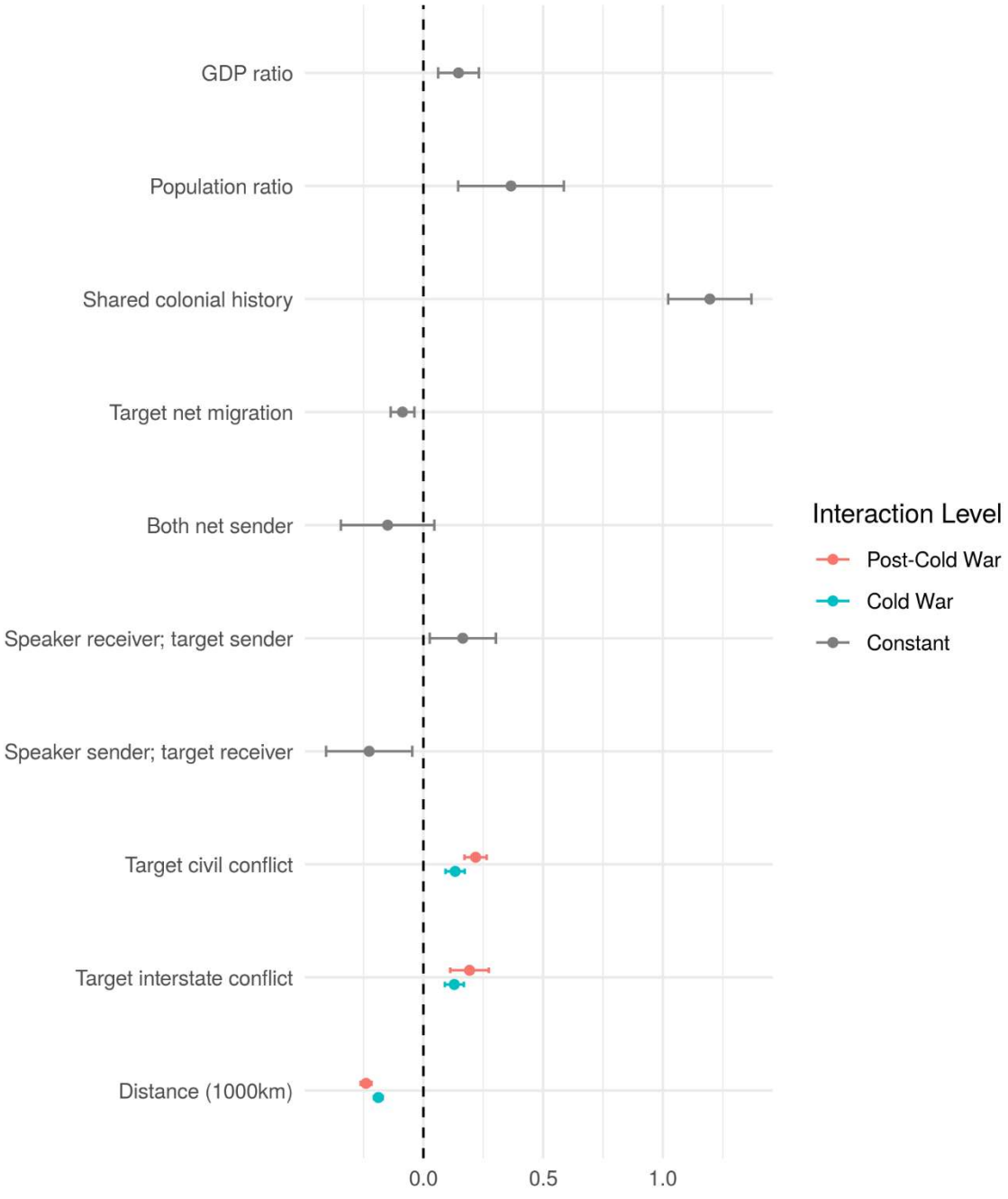

Figure B.1: Coefficient plots, directed dyad-year binary mentions data. *Dots represent point estimates, while lines represent 95% confidence intervals from a conditional logistic regression model. Speaker and target country fixed effects estimated but not shown. Democracy/Democracy is the held-out baseline for the government type variable.*

Appendix C

As described in-text, we use a supervised machine learning approach to predict whether pseudo-paragraphs in UNGA speeches (as identified using Hearst (1997)'s TextTiling algorithm) are relevant to international borders and boundaries. As features for this prediction model, we used per-paragraph topic proportion vectors from a LDA model, fit to the dataset of all paragraphs in all speeches in our dataset (n=135,292). We chose this feature extraction approach for two reasons. First, compared with simpler feature extraction approaches (e.g. TF-IDF-weighted word counts or similar), topic proportion features are parsimonious, which eases the computational and estimation burden on our downstream classification models. Second, since our concept of border-relevance is abstract, high-level features like those produced by a topic model are likely to represent the information in our corpus more efficiently than a word-specific model like a word embedding approach.

To construct our training set, we first pre-processed our corpus. We lower-cased all tokens, and removed punctuation, stopwords, numbers, and tokens shorter than 3 letters. We also removed all tokens that occurred in fewer than 10 pseudo-paragraphs and tokens that occurred in more than 99% of pseudo-paragraphs. Next, we fit a 30-topic LDA model (via MALLET), and extracted topic proportion vectors for all paragraphs that contained one of the border-relevant keywords we describe in-text. Finally, for all keyword-identified topics with human annotations (n=2293), we fit a prediction model, where the dependent variable took on a value of 1 if at least one annotator viewed the paragraph as relevant and 0 otherwise. We then used the outputs of this prediction model to generate relevance predictions for the remaining documents in our keyword-identified dataset (n=3084).

As a prediction model, we opted for a random forest. We selected this model to capture potential non-linear relationships between our input features and our dependent variable. In principle, topic models are designed to capture abstract themes within a text, which may include ideas closely related to our annotators' understanding of the notion of border relevance. However, we cannot be sure what quantity of language is sufficient to for human annotators to code border relevance; plausibly, annotators might code "relevance" based on the presence of a single word, a small phrase, a multi-sentence discussion, or anything in between. Models like random forests are well-suited to capture these kinds of non-linear relationships.

To estimate the model, we used 500 trees and selected the number of candidate features at each split selected via cross-validation. This model achieved an average 10-fold cross-validated predicted accuracy of 0.788, and a cross-validated F1 score of 0.867 (averaged over 20 iterations).

Appendix D: Mechanical Turk Sentiment Model

As described in-text, we use a supervised machine learning approach to predict sentiment values for each document in our dataset. In this Appendix, we provide details on feature engineering, feature selection, and modeling choices used in this task.

Feature extraction

We use two types of features in our models. First, for each document, we extract unsupervised sentiment values using four off-the-shelf methods:

- *Google AI sentiment predictor*. The Google AI API provides a pre-trained sentiment classifier. Given a document, the API returns a "sentiment" score and a "magnitude" score. Roughly, the "sentiment" score corresponds to the average sentiment of a given document, while the "magnitude" score corresponds to the frequency of "emotional" content in the document.
- *Bing, Loughran, and NRC sentiment dictionaries.*. Like most sentiment dictionaries, each of these dictionaries contains an expert-compiled list of words coded as "positive" or "negative" sentiment. For each dictionary, we counted all terms coded by that dictionary as positive or negative. We then combined these scores into a single value by calculating a normalized difference in counts, defined as $diff = \frac{pos-neg}{pos+n}$.

As mentioned in-text, because of the challenges involved in analyzing diplomatic language, we do not expect existing sentiment dictionaries and prediction algorithms to perform especially well in our application. However, we do expect these features to be at least somewhat related to our sentiment conceptualization. As a result, these features offer a reasonable starting place in the feature extraction process.

Second, we created an aggregated word embedding vector for each document. As mentioned in-text, an embedding model is essentially an unsupervised dimensionality reduction tool, which converts a given word into a high-dimensional vector representation designed to represent the semantics of that word. For example, a particular element of an embedding vector might correspond to the "gender" of a word (e.g. "king" vs "queen"), or the "emotional" content of that word (e.g. "enraged" versus "concerned"). To capture a wide variety of semantic dimensions, embedding vectors are usually high-dimensional (50+ elements). However, when aggregated to the document level, high-dimensional word embedding methods are substantially lower-dimensional and denser than word-count matrices or other feature engineering methods designed to represent document content. Since we have a relatively small training set (n = 1124), these latter two features are particularly attractive during model fitting.

To generate an embedding vector for each individual token, we draw on pre-trained 50-dimensional GloVe embeddings, trained using the Wikipedia 2014 and Gigaword 5 corpora. For each word in each document, we extracted an embedding vector, and averaged the vectors pre-trained for each individual word in a given document. This process provided us with a 50-dimensional embedding vector for each document, which we used as an additional set of features in our downstream prediction model.

Model selection

Using these feature sets, we experimented with five modeling approaches: namely, a simple linear regression, a lasso-, ridge-, and elastic net-penalized linear model,[1] and a random forest.[2] For each approach, we trained model versions with dictionary features only, and both dictionary and embedding features, leaving us with a total of ten candidate models. For each modeling variant, we assessed out-of-sample RMSE and correlation via ten-fold cross-validation.

The results of this process are given in Figure A1. All approaches out-perform a baseline generated using the Google AI sentiment score, which was the best-performing of the unsupervised methods we examined. Models trained using both dictionary and embedding features performed best, suggesting that both feature sets indeed added to predictive performance. By contrast, choice of model mattered less, with the three penalized models slightly out-performing the linear model and random forest by correlation and performing equivalently by RMSE.

Based on these findings, for our analyses in-text we opted to use scores generated using the ridge-penalized linear model, with both dictionary and embedding features included. This approach offers a noticeable – though modest - performance gain over the unsupervised baseline, with approximately a 10% reduction in out-of-sample RMSE and an increase from approximately 0.51 to 0.63 out-of-sample correlation moving from the unsupervised Google AI sentiment scores to ridge-penalized model.

[1] With mixing parameter equal to 0.5 and lambda selected by cross-validation.

[2] With 500 trees and the number of candidate variables at each split selected by cross-validation within the training set.

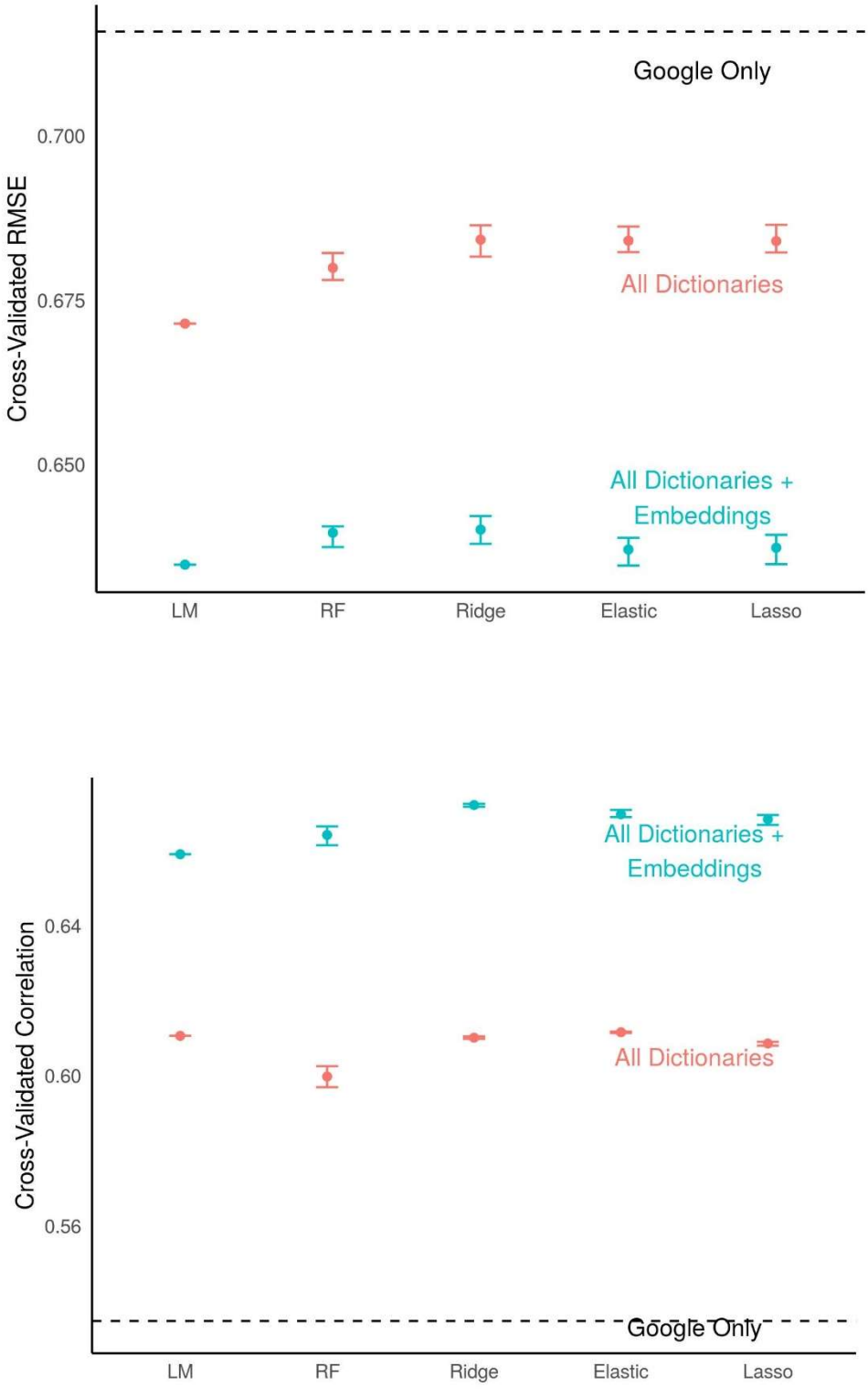

Figure A1: Cross-validated RMSE and correlation. *Dots represent average cross-validated performance over 20 iterations, while lines represent 2.5th and 97.5th percentiles. Baseline is the Google AI sentiment algorithm.*