Thank You, Senator: Predicting Politicians' Gender from Responses to their Facebook Posts

Emily Rapport
University of California, Berkeley
School of Information
emilyrapport@berkeley.edu

Lorrel Plimier
University of California, Berkeley
School of Information
lorrel.plimier@berkeley.edu

Abstract

While there is a corpus of research surrounding language and gender, little NLP research focuses on how a speaker's gender influences the responses he or she receives. To explore this area, we use classifiers to predict a politician's gender based on the *responses* to his or her Facebook posts, with a goal of generalizing well to new politicians. We start by including sentiment and responders' gender as features in a logistic regression model. We then try various convolutional neural network (CNN) architectures using pretrained word vectors. We analyze the errors of the CNN and discuss the shortcomings of our model as it is applied to this specific predictive task.

1 Introduction

As the United States has seen an increase in women running for (and winning) elected office over the previous few decades, there has been increased interest surrounding gender bias in politics. More recently, online social media has become increasingly influential in elections. We seek to examine the way that gender affects social media interactions in the political sphere. The RtGender data set created by Voigt et al. (2018) provides a mechanism for examining gender dynamics in online speech by compiling Facebook posts from US Congress members and responses to those posts from other Facebook users. As this is a relatively new data set (and a sub-component of a larger set), there is no existing baseline for the task of using a *response* to a politician to predict the politician's gender. In this project, we aim to create a baseline for this prediction task using a linear model, then improve upon that baseline with more complex models. We also hope to use our modeling process to better understand the challenges and opportunities in this problem space.

2 Background

Much existing work explores the relationship between language and gender. While most of this work centers around the differences in language *used* by people of different genders, there has been some exploration of the differences in language used in *response* to people of different genders. For example, Fu (2016) analyzed post-game interview questions asked to professional tennis players and used language modeling to assess differences in the relevance of the questions the players were asked based on their genders. Adding gender to the Enron corpus of emails, Prabhakaran (2014) explored the differences in language used between pairs of correspondents in email threads. They used the gender of the correspondent and the overall gender environment as additional features to improve the results of an SVM model that predicted the hierarchical power relationship for the pair of email correspondents.

Recognizing a relative lack of study of how speech differs in response to speakers of different genders, Voigt et al. (2018) created RtGender as a cross-domain, cross-platform dataset for the study of how online comments differ based on the gender of their target. The authors gathered 25 million response-post combinations from five different domain/platform combinations, ranging from politicians' Facebook posts to fitness-oriented subreddits, with the intention of promoting further research on gender and response speech.

In framing our project as a gender classification task, we look to many examples of the use of neural networks to improve the accuracy of NLP classification. In particular, Kim (2014) found then-state-of-the-art results could be achieved for NLP classification tasks using a neural network with a single convolutional layer and pre-trained

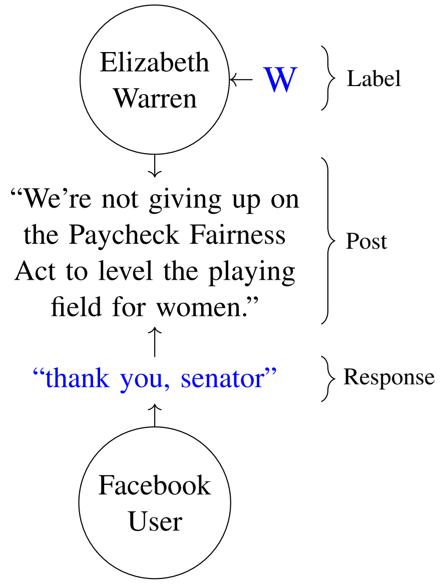
Elizabeth Warren
W
Label
"We're not giving up on the Paycheck Fairness Act to level the playing field for women."
Post
"thank you, senator"
Response
Facebook User

Figure 1: Problem Setup. We aim to predict labels (W) from Responses ("thank you, senator").

word vectors. Many subsequent research projects have found continued success applying convolutional networks to classification problems, including Johnson and Zhang (2017), who designed a deep CNN architecture with layers of gradually decreasing size that outperformed previous best models on benchmark NLP datasets.

3 Methods

3.1 Data

We focused on a subset of the RtGender dataset that consists of Facebook posts from 412 politicians (306 men and 96 women) and first level responses (comments directly on the original post, rather than nested comments), for a total of 399,037 posts and 13,866,507 Responses. Figure 1 depicts the relationship between the input text and the gender labels for our data. We use "Response(s)" to denote our model input and "W" or "M" to describe our class labels, which represent the gender of the politicians. In addition to the labeled Responses, we used 3,872 annotated post-response pairs that include crowd-sourced labels for the sentiment of the Response.

Since there is a relatively small set of politicians in scope, we recognized the risk that the model could effectively memorize which Responses were to which individual politicians and predict the correct gender by overfitting to the politicians in the training set. We thus opted to split our data set into train, dev, and test sets based on the politician, such that Responses to a given politician's posts would be in only one of the three sets. By evaluating the model on Responses to politicians who were unseen during training, we felt more confident that our metrics reflected the model's ability to learn patterns that apply generally to Responses.

The imbalanced proportion of the two classes adds a challenge to the gender prediction task. There are more male than female politicians in the overall data set, reflecting the overall gender balance of the US Congress. This imbalance translates indirectly to the number of training rows given that the number of Responses per politician varies significantly (though there does not appear to be a correlation between number of Responses and poster gender). The 3,962,284 rows in the reduced training set contain 73.2% M. The dev and test sets are even more imbalanced with the dev set containing 85.0% M and the test set containing 84.6% M.

3.2 Metrics

Knowing our data was imbalanced, we strove to find evaluation metrics other than accuracy, since the naive baseline of guessing M every time scores relatively well on this data set. We tracked receiver operating characteristics (ROC) curves and precision-recall curves (PRC) and used the area under these curves (AUROC and AUPRC) as summary metrics for our models. Because the AUPRC does not rely on true negatives (in our case a male politician predicted as M), the metric will not be overly influenced by the predominance of the M class in our data (Saito and Rehmsmeier, 2015). We thus use AUPRC as our primary evaluation metric in conjunction with AUROC and validation accuracy.

3.3 Models

We began our modeling process with a simple logistic regression model, then supplemented this model by adding the responder's gender and a sentiment analysis rating as features. We followed this baseline with a series of CNN models where we experimented with data preprocessing, overall model architecture, and hyperparameter tuning.

Baseline Model and Sentiment Analysis: For our baseline models, we used Scikit-learn's LogisticRegression class and its HashingVectorizer, which allowed us to transform chunks of data at a time and bypass memory issues related to our large data size. This initial model used 1,048,576 unigram features, and got 82.4% accuracy on the dev set, with an AUPRC of 0.236. The model tended to underpredict the W label (6.7% of predictions on the dev set).

To explore our theory that sentiment might vary significantly in the Responses to male and female politicians, we conducted a basic sentiment analysis with the goal of generating features that would improve the accuracy of our baseline model. We implemented three common algorithms using Hu and Liu opinion lexicon[1], TextBlob[2] and VADER[3]. We then tested the accuracy of our sentiment ratings against the sentiment-annotated subset of our data. While the accuracy of all three algorithms varied by only 9%, the VADER algorithm, which was developed specifically for social media content (Hutto and Gilbert, 2014), performed the best with an accuracy of 57.4%, an improvement over a naive all-positive prediction of 41.4%.

Based on our initial hypotheses and EDA, we suspected that adding both the Vader sentiment and the responder's gender (inferred from first name) could provide additional signal to our baseline logistic regression and potentially improve its accuracy at gender prediction. When these features gave us no additional benefit, however, we decided to pivot to neural network methods since they had the potential to provide us with a more nuanced representation of the input language.

Convolutional Neural Networks: We used Keras to implement CNNs with architectures inspired by two key papers, as well as some models with custom architectures. Kim (2014) uses an input layer composed of the input text represented through pre-trained word vectors, a single convolutional layer with differing filter sizes (3, 4, and 5) and rectified linear unit (ReLU) activation, and a final classification layer with dropout applied. After experimenting with the Kim model, we were interested in trying a deeper CNN while remaining mindful of the overall number of parameters in the model (due to our desire to keep training time reasonable). Johnson and Zhang (2017) increase model depth while mitigating parameter growth by stacking blocks of convolutional layers followed by max pooling layers with size 3 and stride 2; the stride of 2 ensures that only every other internal representation is sent to the next block, decreasing the size of the representation by half with each block. Johnson's final model uses an input layer composed of the input text represented through pre-trained word vectors, with dropout applied; seven blocks, each with two convolutional layers followed by a max pooling layer; and a final sigmoid classification layer. The application of the Johnson model that we adopted is laid out in Figure 2. We also experimented with custom models that followed neither architecture, which allowed us to experiment more freely with number of layers, layer sizes, and filter sizes.

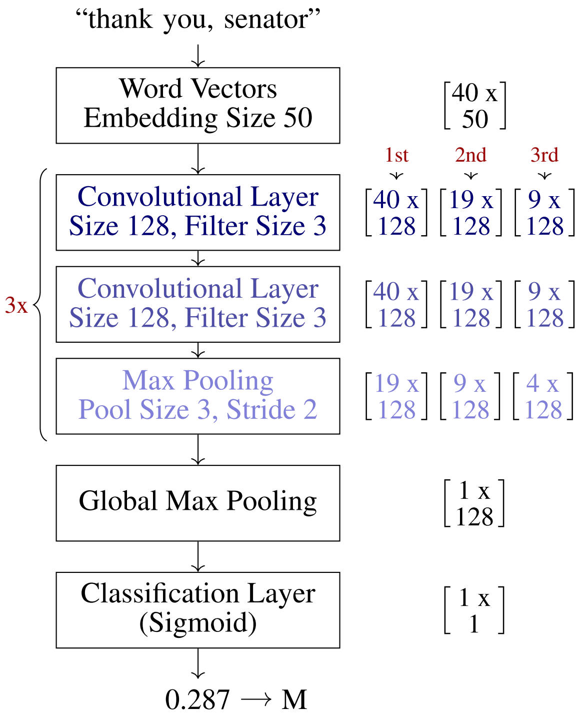

Figure 2: Model Architecture (output dimensions labeled by layer)

We used 50-dimensional GloVe[4] word embeddings in all versions of our CNNs. To keep our size of trainable parameters smaller than our number of examples, we did not allow the word embeddings to be updated during the training process.

3.4 Modeling Strategies

At the outset of modeling, we identified key challenges related to our data set and experimented with strategies for overcoming them.

[1] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html#lexicon

[2] https://textblob.readthedocs.io/en/dev/

[3] https://www.nltk.org/_modules/nltk/sentiment/vader.html

[4] https://nlp.stanford.edu/projects/glove/

Identifying Unknown Words: We found that about 95% of the unique words in the training set were unknown to GloVe. The unknown words that appeared most often in our data were contractions, so we used a pre-made dictionary[5] to convert contractions into their expanded versions (ex: "should've" → "should have"). While this removed only a small number of the unknown words, it impacted many examples. The remaining unknown words had more disparate patterns (emojis, misspellings, hashtags), and we chose not to invest in further preprocessing, recognizing that even the most common unknown words were only seen a handful of times and that focusing on modeling choices would be more impactful.

Selecting Training Rows: The distribution of Responses per post across the entire data set has a strong right skew, with a mean of 34, a median of 6, and a maximum of 157,366. The posts with a large number of Responses are generally from well-known members of congress, and their content touches on controversial issues. When we initially saw the model overfitting, we suspected it might be because the high-response posts were over-represented in the training data but were not representative of the gender prediction task at large. By randomly keeping only 50 Responses per post, we reduced our number of training rows from over 9,000,000 to 3,962,284, and our validation accuracy remained more stable over the training epochs. We did not apply this reduction to the dev or test data, as we want the trained model to perform well on new examples whether or not they are from high-response posts.

Combating Class Imbalance: We experimented with using Keras class weighting to put double or triple weight on the W class during the training process in hopes that it would help the model better learn to recognize W labeled Responses. We found that as we increased the weight put on W labeled Responses, the number of W labels predicted went up, as did recall, indicating that the model got more true positives; however, its precision went down, as many of the new W predictions were false positives. Since both AUPRC and validation accuracy also decreased when we augmented the W class weights, the models in the Results section do not reflect use of this strategy. We also experimented with dropout rates as a method for combatting class imbalance and helping the model to generalize more effectively. (Dabare et al., 2018)

3.5 Hyperparameter Tuning

To tune model hyperparameters, we ran many combinations of each of the three architectures we considered (Kim, Johnson, and custom), systematically varying 1-2 hyperparameters at a time to see how the results changed. We experimented with the following hyperparameters (whose application varied slightly across the model architectures): maximum sequence length; number of convolutional layers (Johnson and custom only); convolution layer sizes; filter sizes; and dropout rate. Overall, we trained over 75 distinct gender prediction models. During the hyperparameter tuning process, we used only the dev set for validation and other metrics, switching to the test set only for final error analysis.

Manual Coding Exercise: Noticing minimal improvements in validation accuracy while hyperparameter tuning, we questioned whether there was any signal in our data. Per James' suggestion, we pulled 1000 random Responses from our dev set and manually coded them for gender of the politician of the original post. We compared our own scores to that of one of our early CNNs. The results are shown in Table 1. While the model was more accurate than we were overall, we were able to correctly predict W more often, which is why we get significantly higher AUPRC than the model. That said, we found this task very challenging, and had relatively low confidence in our guesses.

Coder	**No. Correct**		**AUPRC**	**Accuracy**
	M	**W**		
Lorrel	664	82	0.480	74.6%
Emily	716	64	0.422	78.0%
Model	777	14	0.231	79.1%

Table 1: Manually Coded 1000 Responses

In coding these Responses ourselves, we made observations about our data that we applied to our modeling. First, we discovered that there was very little signal in one or two-word Responses such as, "Thank you!" (with two exceptions: "me too" and "go Rubio"). Second, we noticed that for longer Responses, the words most likely to indicate gender were at the beginning and end of the Response.

[5] https://mlwhiz.com/blog/2019/01/17/deeplearning_nlp_preprocess/

We used these insights to inform our preprocessing choices.

4 Results

After our initial rounds of experimentation using ten epochs, we chose our top ten best-performing models and retrained them for 30 epochs. The best six of these models included two that use the Kim (2014) architecture, two that use the Johnson (2017) architecture, and two that use a custom model architecture.

4.1 Top Performing Models

The architecture and results of the best six models as well as our baseline logistic regression model can be seen in table 2. As discussed above, while we used AUPRC as our main evaluation metric, we also tracked AUROC and validation accuracy.

Though our model with the best AUPRC score was one of the Johnson models, we also had high-scoring Kim and custom models. Our results don't point to the superiority of one particular architecture, suggesting instead that multiple architectures and hyperparameter combinations are reasonably suited to the task, and that further tuning of the various architectures might still be needed.

4.2 Effects of Data Preprocessing

As we had surmised upon completion of our manual coding, removing shorter Responses and including the beginning and ending words for longer Responses improved our classification results. Table 3 shows the AUPRC and AUROC steadily increasing as we add these two preprocessing steps to our first custom model. We used these tactics in all subsequent models.

5 Analysis of Results

During the training process, our gender models typically reached their lowest validation loss within a couple of epochs, then flatlined as the training loss continued dropping. This suggests overfitting. However, our own struggle when we attempted the manual coding task, combined with our analysis of the types of errors the model makes, cause us to question whether there are many nuanced, generalizable rules to learn in this problem at large.

To understand patterns of model successes and errors, we looked at subsets of Responses that contained a particular word or phrase and compared the predictions for those subsets to the predictions for the entire test set, looking at distributions of model probabilities, percentage of W predictions, and error rate. This method allowed us to speculate about feature importance in the model, with this caveat: by looking only at a particular word, we are merely approximating a simple version of the features the CNN may ultimately be learning. It could also be that the words we chose are actually highly correlated with *other* words that the model learned as features (though spot-checking suggests that we weren't missing highly correlated neighbor words). While we clarify that this method is only approximating feature importance and further study would be needed to make more conclusive claims, this analysis nonetheless turns up some interesting results.

5.1 Pronouns, Titles, and Names

Pronouns and titles seem to point the model strongly towards a gender prediction, though these signals aren't always correct.

The model confidently predicted M for many Responses that included the word "sir." Similarly, the words "congressman" and "congresswoman" are strong predictors of original poster gender, as they are often used to directly address the poster:

> "good move, ms. congresswoman."

While the input "thank you" receives a probability of 0.297 (relatively close to the overall prediction mean of 0.243), "thank you congressman" receives 0.005 (confident M prediction) and "thank you congresswoman" receives 0.999 (confident W prediction). This suggests the importance of "congress(wo)man" as a feature.

The word "she" is much more problematic as a gender signal. Responses with "she" were predicted as W at a much higher rate than Responses at large were (39.6% vs 5.3%) . The model, however, gets these predictions incorrect 37.9% of the time, compared to 17.8% of the time overall. The word "she" is often picking up on women who are referenced within the content of the original post, like when (male) Senator Thom Tillis makes a post referencing (female) Senator Kay Hagan and a Response reads: "she sucks..." (model prediction: W). Often, the Responses that the model was less confident about contained some words that indicate female and some words that indicate male, none of which reference the politician:

Model	AUPRC	AUROC	Val-Acc	Max seq length	No. layers	Layer size	Filter size	Dropout rate
Baseline	0.236	0.635	0.824	NA	NA	NA	NA	NA
Custom-1	0.287	0.671	0.849	50	2	128	2	0.5
Custom-2	0.288	0.662	0.849	20	3	32	2	0.2
Kim-1	0.289	0.657	0.850	20	1	192	3,4,5	0.2
Kim-2	0.271	0.645	0.849	20	1	384	3,4,5	0.2
Johnson-1	0.276	0.654	0.834	20	4	256	3	0.2
Johnson-2	0.308	0.680	0.844	40	6	128	3	0.2

Table 2: Model Results

Preprocessing	AUPRC	AUROC
None	0.271	0.652
Remove one & two-word responses	0.285	0.669
Remove one & two-word responses; use beg/end words	0.287	0.671

Table 3: Results with Preprocessing Techniques

> "it has (sic) not limited to bathrooms, if a male student feels like a female today and wants to reside in an all girl dorm on college campus, they have to admit him."

Like with pronouns and titles, first names were sometimes a strong predictor of gender for the model, but other times misleading. The model confidently predicted M for this Response:

> "merry christmas tom!!! keep up the great work in 2014 !!!! :-)"

While Tom Cotton, the addressee of this comment, does not appear in the training set, eight other Toms do, which allowed the model to learn that "Tom" is a male name.

The first name signal backfires for the model when politicians are being referenced rather than addressed. (Female) Senator Kamala Harris's Responses are in the train set, and the model predicts W in 86.3% of test set Responses that contain the word Kamala. However, the "Kamala" rows in the test set are primarily Responses to a post that (male) Senator Cory Booker made about Senator Harris. Most of the Responses mention Kamala by name, and as a result, the model predicts all of these examples incorrectly. This case shows how much the particular train/dev/test split in this problem can impact what the model learns and how it scores, as the model might not have learned that Kamala is a signal for a W label if Senator Harris had been sorted into the dev or test set.

The cases of pronouns, titles, and names suggest one of the fundamental limitations of this task: by attempting to go from response text to poster gender, we are bypassing the context of the original post. When the original post references people other than the poster, the gender context of the referenced individual throws the model off. These are examples that we would expect humans to get wrong as well, where the signal between input and label is weak or nonexistent.

5.2 Evidence of Bias?

In addition to explicitly gendered language, we suspect that the model may have also picked up on other words that are used more often in response to one gender. Table 4 highlights some of these potential biases.

Word	No.	% W		Error
	Resps	Preds	Labels	Rate
beautiful	5298	13.2	24.8	25.4
attractive	123	17.1	34.1	26.8
[full test set]	1.4M	5.3	17.2	17.8

Table 4: Test Set Predictions by Contained Word

The words "beautiful" and "attractive" appear disproportionately in W Responses in both the train and test sets. The model seemed to learn some signal between these words and the W label: test Responses containing "beautiful" or "attractive" were both predicted as W at significantly higher rates than the test set as a whole, as seen in Table 4. Interestingly, the Response "so beautiful" received a prediction of .537 (slight W,) while "beautiful words" and "beautiful post" received predictions of .386 and .372 (slight M,) which suggests that the model could actually be less likely to treat "beautiful" as a signal for W if it's clearly not referring to a person. The correlation between these Response words and W labels could be taken as evidence of gender bias against female politicians, if they are more likely to be objectified based on their appearance compared to their male counterparts, and our model seems to have partially picked up on this bias.

6 Next Steps

While we would have liked to have spent more time on tasks like hyperparameter tuning and data preprocessing, ultimately, we believe that there are some fundamental limitations in the amount of signal between Responses and labels that will prevent the accuracy and AUPRC from improving substantially. The prevalence of Responses that reference gendered subjects other than the politician adds a lot of noise to the task; additionally, though we were able to get higher AUPRC than any of our eventual models in the manual coding exercise, we suspect that many of our correct predictions stemmed from our knowledge of non-language concepts that likely would not come through in the training data. Further work might include reframing the problem by scrubbing congress members' names or gendered pronouns; though the model would likely do worse at the predictive task, perhaps in the absence of these overwhelmingly strong signals, it would be more effective at learning patterns that speak to the presence (or lack thereof) of gender bias in the Responses.

References

Balahur, A. (2013). Sentiment Analysis in Social Media Texts. *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 120–128.

Buda, M., Maki, A. and Mazurowski, M.A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. `https://doi.org/10.1016/j.neunet.2018.07.011`.

Dabare, R., Wong, K.W., Koutsakis, P., and Shiratuddin, M.F. (2018). A Study of the Effect of Dropout on Imbalanced Data Classification using Deep Neural Networks. *Journal of Multidisciplinary Engineering Science and Technology (JMEST)*, Vol. 5 Issue 10, 8905-09 Western Australia, Australia.

Fu, L., Danescu-Niculescu-Mizil, C., and Lee, L. (2016). Tie-breaker: Using language models to quantify gender bias in sports journalism. *Proceedings of the IJCAI Workshop on NLP Meets Journalism.*

Hendrycks, D. and Gimpel, K. (2017). A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. *International Conference on Learning Representations 2017.*

Hutto, C.J. and Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. *International AAAI Conference on Weblogs and Social Media. AAAI*, 216—225.

Johnson, R. and Zhang, T. (2017). Deep Pyramid Convolutional Neural Networks for Text Categorization. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 562—570. Vancouver, Canada.

Kim, Y. (2014). Convolutional Neural Networks for Sentence Classification. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751. Doha, Qatar.

Martinc, M. and Pollak, S. (2018). Reusable workflows for gender prediction. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*, 515–520, Myazaki, Japan.

Mihaltz, M., Varadi, T., Cserto, I., Fulop, E., Polya, T. and Kovago, P. (2015). Beyond Sentiment: Social Psychological Analysis of Political Facebook Comments in Hungary. *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA 2015)*, 127–133, Lisbon, Portugal.

Prabhakaran, V., Reid, E. and Rambow, O. (2014). Gender and Power: How Gender and Gender Environment Affect Manifestations of Power. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1965—1976.

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.doi:10.1371/journal.pone.0118432.*

Voigt, R., Prabhakaran, V., Jurafsky, D. and Tsvetkov, Y. (2018). RtGender: A Corpus for Studying Differential Responses to Gender. *LREC 2018 Proceedings: Eleventh International Conference on Language Resources and Evaluation*, 2814–2820.