Chapter 7: Prediction[*]

Justin Grimmer[†] Margaret E. Roberts [‡] Brandon M. Stewart [§]

March 19, 2018

1 Introduction

In this chapter, we consider the task of prediction. Understanding how to use text to measure *causation*, like we did in the last chapter, can help us build social science theories and make policy decisions. Prediction; however, is a distinct task of building a crystal ball. Predicting what will happen to a person, an economy, an institution, or a country not only can inform policy decisions because it can reveal what the circumstances of the future might look like, it also is the focus of many individuals and businesses trying to make money in our economy or build tools that provide very day utility for people. Indeed, one of the primary businesses on the Internet is predicting which advertisements and links web surfers will click on in order to optimize the placement of links on webpages. Predicting stock market or business success is the primary task of investors and economic forecasters. Predictions spelling errors or the next word you will type in your text message are now default software on most phones, computers, and search engines.

Text densely stores information about people, businesses, and governments and therefore can be extremely useful for prediction. For example, your e-mail client uses the text of your

[*] Preliminary and incomplete draft, please do not circulate.

[†] Associate Professor, Department of Political Science, University of Chicago

[‡] Assistant Professor, Department of Political Science, University of California at San Diego

[§] Assistant Professor, Department of Sociology, Princeton University

e-mail to decide whether or not the e-mail is important to you and whether or not you would classify it as spam. Search engines use the text of yours' and others' previous searches to suggest the ending to your search even as you type. Text is also useful in social forecasting – researchers are using the text of legal proceedings to predict decisions (Aletras et al., 2016), the text of tweets to predict box office returns on movies (Asur and Huberman, 2010), and the text of newspaper articles to predict the onset of conflict (Mueller and Rauh, 2017).

While machine learning algorithms in computer science have been developed with almost an exclusive eye toward prediction, social scientists have been more hesitant to make prediction the ultimate goal of their research and instead often focus on making descriptive or causal inferences, like we described in the previous chapters. However, as social science has encountered a deluge of new types of data that are often now available to them in realtime, prediction may begin to take on increasing importance Predictive models are not only useful for practitioners, they also may be help clarify the usefulness of particular types of social science theories and contextualize effect sizes.

We introduce the basic task of prediction to start the chapter. Then we clarify how prediction differs from causal inference. We then explain some general misconceptions about prediction that the reader should keep in mind when using text for prediction. Then, we will turn to different examples of prediction using text data – first, *source prediction* or prediction that detects the identify or attributes of the source of a text, second, *linguistic prediction* or prediction that generates text, third, *societal prediction* or using text to predict economic, political, or sociological events in the future and last, *nowcasting* or using text to predict ongoing events that cannot be measured quickly enough through other means.

2 The Basic Task of Prediction

To start, we introduce a basic task of prediction that we will return to throughout the chapter. Our goal is to predict an outcome y_i where i indicates that the prediction is made for a particular unit (such as an individual, business or country).[1] We want to forecast y_i, that is guess a value for y_i that we will call $\hat{y}_i$, before we actually have the chance to observe y_i.

We can use any data that we have available to us to predict y_i. For each individual observation, we will call non-text predictors of y_i $\boldsymbol{x}_i$, or a collection of P variables that we have observed for unit i at the time we make the prediction that we will use to predict y_i. We will refer to text predictors of y_i $\boldsymbol{w}_i$, or the collection of J features about the text that we will use to predict y_i.

We will need three essential components to build a model to predict y_i. First, we need some data that is *labeled*, meaning that we have both the predictors $\boldsymbol{x}_i$ and $\boldsymbol{w}_i$ as well as the resulting y_i for these observations. This will allow us to create a model to predict future y_i's. Second, we to split the labeled data into *training* and *test* data – some data that we can use to create the model and some data that we can use to estimate the model's prediction error. Prediction error is often known as the *generalization error* because it is the estimated error when the model is fit to new data.

The third component we need is the model. We can think of selecting the model (as we did in Chapter 5) as selecting a *loss function* $L(\cdot)$, or how we quantify the difference between the predicted outcome and the actual outcome within the labeled data. The model will try to find the parameters that minimize $L(\cdot)$. To take the simplest example, we might use a linear regression to predict a 1xN vector of y's $\boldsymbol{y}$ from a NxP matrix of their corresponding

[1] Often this y_i will include additional indices which specify other aspects of the prediction. For example, we might want to predict y_i at time $t+1$ or in geography g. In this case, the task could be to predict $y_{i,g,t+1}$. For now, we will keep things general to application and denote our quantity we are interested in predicting as y_i.

$\boldsymbol{x}_i$'s, $\boldsymbol{X}$. In this case, we would minimize the sum of squared errors, such that

$$\texttt{arg min}_\beta \ \ \textstyle\sum_{i=1}^N (y_i - x_i\beta)^2$$

to obtain coefficients $\hat{\boldsymbol{\beta}}$. We would then be able to use the estimated $\hat{\boldsymbol{\beta}}$ in the future when we don't know y_i to obtain a prediction $\hat{y}_i$ based on computing $\boldsymbol{x}_i\beta$ using the $\boldsymbol{x}_i$ that we observed at the time. Of course, linear regression is a very simple predictive model and we are likely to be able to do a lot better with text by including a more complex functional form or including regularization, for example. Any of the models described in the measurement chapter – from kernel methods to trees to ensembles – can also be used in prediction.

How will we know which model is best for a particular predictive task? To evaluate which model is best for prediction in our application, we want to estimate what the generalization error or prediction error would be for each model when we predict the next $\hat{y}_i$. To do this, instead of using the in-sample error from the training set I to measure the error rate, we would use the out-of-sample error from the held out sample O to measure the error rate by taking the expectation of the loss function L in the out-of-sample data:

$$Error_{out} = E[L(\boldsymbol{Y}_{i \in O}, f(\hat{\beta}, \boldsymbol{x}_{i \in O}))|I]$$

If the out-of-sample data is representative of data that we want to use in the predictive set, then the error that we are interested in is the expectation of the out of sample error rate, as the average error rate across many different representative held-out samples would approximate the true error rate.

$$Error = E[E[L(\boldsymbol{Y}_{i \in O}, f(\hat{\beta}, \boldsymbol{x}_{i \in O}))|I]]$$

To estimate this error rate in practice, analysts use cross-validation, where they randomly leave a subset of posts out, estimate the error rate, then repeat K times within the same dataset. In general, when the held out data is a random sample of the dataset of interest, cross-validation can be a good and accurate measure of the expected error rate and can be an appropriate way to select the model to use for prediction.

2.1 Similarities and Differences Between Prediction, Measurement, and Causal Inference

You might notice that up until now, we have described a problem almost identical to the supervised learning problem described in the measurement chapter, where y_i is the category of interest and $\boldsymbol{w}_i$ are the words within the document that we use to predict that category. It is also very close to the set up of the problem in causal inference, where y_i is the outcome of interest, t_i is the treatment of interest, and $\boldsymbol{x}_i$ are potential confounders. In all cases, we could use a variety of models to estimate the relationship between y_i and the predictors. In each case, we also advocated for some type of split sample approach for model validation.

While they seem similar on the outside, prediction is closer to the supervised learning problem described in the measurement chapter than it is to causal inference. You can think of the supervised learning described in the measurement chapter as one specific example of prediction – you are trying to predict how hand coders would label a document given the words within the document $\boldsymbol{W}$ and their covariates $\boldsymbol{X}$. We describe estimating misclassification error in the same way that we describe prediction error here and suggest similarly that analysts use held out samples to characterize this error. The only difference in what this chapter will tackle in comparison to the supervised learning problem discussed in measurement is that we are interested in predicting a range of y_i's much broader than just the label of a particular document, including the outcome of Supreme Court decisions, the future of the stock market, or using text to predict subsequent text. This comes with more considerations than the more focused task of labeling documents.

Because of their similarities, many of the same models we described in the measurement chapter can also be used for prediction. Ridge and lasso regression, support vector machines, neural networks, tree models, and ensembles are all common methods commonly used to predict outcomes y_i based on inputs $\boldsymbol{w}_i$ and $\boldsymbol{x}_i$. We talked through each of these in the supervised learning section of the measurement chapter. For brevity purposes, we won't review all of these models in this chapter, but highlight a few examples of how they can be used in a broader predictive context.

Unlike the comparison between prediction and supervised learning, the task of causal inference is quite different from the task of prediction and the distinctions between these two tasks are paramount (See Shmueli (2010) for a good overview of the differences between explanation and prediction). The fundamental difference between prediction and causal inference is that in the former we are trying to predict a future outcome or data point, and in the later we are trying to estimate one or a set of coefficients within the model without bias. In prediction, we are trying to estimate as precisely as possible the outcome – this is a predictive estimand. We do not necessarily care how we arrived at that prediction or what we used to get there, as long as the forecast of the future is accurate.

In causal inference, we are interested in what *would* have happened or would have been different had one or a set of independent variables changed. The difference between what *would* have happened – the counterfactual – and what did happen – the outcome – is the causal estimand or coefficient of interest, which we want to estimate without bias. Unlike prediction, we often do not care as much about whether or not our model accurately predicts the outcome because we observe the outcome; instead, we want to predict the *potential outcome*, or our best guess of what *would* have happened had one or a set of the independent variables been different.

The differences between the tasks of causal inference and prediction have important implications for what we do with the data in each case. Indeed, we believe that much confusion about causal inference and prediction stems from their conflation with each other. In prediction, as we discuss below, we are much less interested in what goes *into* the model, as long as the predicted outcome of the model is accurate. In prediction, we are comfortable introducing bias to the coefficients in the model, as long as it improves our predictive power. Not so in causal inference, where we have to be very careful about the role in the causal process for each variable that enters the model so to make sure we are estimating what would have happened, importantly by avoiding post-treatment bias and accounting for potential confounding.

Similarly, while we have suggested a split sample approach for validation in both causal inference and prediction, the split sample is used differently in each case. In prediction, we use the split sample to estimate the error rate and tune the model. That is, we revisit the model to retune it until we have minimized the generalization error in the test data. Not so in causal inference, where the split sample is used for identification and to ensure that we are correctly characterizing the uncertainty around the treatment effect or the coefficient of interest. Indeed, returning to the model in causal inference might cause overfitting or an Analyst Induced SUTVA violation, as we described in the last chapter. Because we think the distinction between prediction and causal inference is so important, we will return to this theme throughout the chapter, starting with the common misconceptions in the next section.

3 Common Prediction Misconceptions

In this section, we cover some common misconceptions about prediction, highlighting the differences between prediction, causal inference, and measurement to help clarify how the task of prediction defines how the analyst should go about the research process.

Misconception 1: Predictive features have to cause the outcome. Unlike causal inference, a prediction task does not require that predictors cause the outcome. In prediction, how we arrived at the prediction matters less than the accuracy of the prediction. For example, say that we want to predict swimsuit sales. We find that social media mentions of ice cream are a good predictor of swimsuit sales. Of course, social media mentions of ice cream do not cause swimsuit sales, but the two might be correlated because they are both caused by a combination of hotter weather and summer vacation. However, social media mentions of ice cream could and should still be used in a predictive model of swimsuit sales if the data improve overall predictive power.

Of course, causal factors may be useful for prediction and including them in a model may increase the accuracy of the predictions. Including information on the weather, weekends, or vacation – all of which likely cause swimsuit sales – may indeed help in a model predicting swimsuit sales. But in some cases even causal factors may not be very predictive of the outcome – if the causal effect is minimal in magnitude, even if it is significantly different from zero, it will not necessarily improve the accuracy of the prediction (Lo et al., 2015).

Not only do the predictors not have to cause the outcome, the predictors in a predictive analysis can even be the result, or post-treatment of, the outcome. In an example that we'll cover in more detail at the end of this chapter, nowcasting often uses data that is the *result* of y_i to predict y_i. For example, the incidence of flu in any given area is difficult to observe and data from flu cases often lags several weeks because it first has to be collected by health officials. However, one result of an increased incidence of the flu is that people in the area search for flu remedies on Google. Researchers have used this search data to predict flu incidence data, even though flu incidence itself causes the search, not the other way around (Ginsberg et al., 2009).

As we discussed in the last chapter, when we are making causal inferences, we have to be very careful about where variables are in the causal chain and whether to include. If you include post-treatment variables in a causal analysis, you induce post-treatment bias, undermining the entire analysis. Even worse, if the outcome causes the predictors instead of the predictors causing the outcome, such as including flu searches as a causal factor for flu incidence, we would reach the wrong conclusion that flu searches cause the incidence of flu and not the other way around. In prediction, however, we can ignore these considerations and simply focus on whether the model obtains better predictions of y_i.

Misconception 2: Cross validation is always a good measure of predictive power. Earlier in this chapter, we described how we can evaluate a model's predictive performance by holding out a subset of the data and estimating the error rate on the held-out data. We noted that if the test data is a random sample of the dataset of interest, then estimating the generalization error on the test data would be a good estimate of its performance of the ultimate task. However, often when we want to make predictions, we want to predict points of data that have yet to occur. If the data we train the model on is historical data, it is not typically a random sample of the points we are hoping to predict – future data. If the held out historical data differs systematically from the future data that we are interested in using for prediction, then in general, $Error \neq E[E[L(\boldsymbol{Y}_{i \in O}, f(\hat{\beta}, \boldsymbol{x}_{i \in O}))|I]$ and cross validation will not accurately estimate the error rate.

Let's go back to the example of social media mentions of ice cream predicting swimsuit sales. We might use data from the past two summers to estimate when swimsuit sales will peak based on social media mentions of ice cream. And we might have sold our model to a swimsuit store, who is using it to predict demand this summer. But perhaps in the meantime, the social media platform we are using changed their algorithm to downweight ice cream in the news feed in order to promote the health of their users. Because users aren't reading about ice cream, they might not be sharing as much about ice cream. This would increase our error of predicting swimsuit sales because the relationship between social media mentions of ice cream and swimsuit sales has changed.

The best case for using cross validation to estimate the error rate for a prediction problem will be when the future is like a random sample of past data, in other words when we have information about the future in past data. Forecasting and predictive models that are based on historical data rely on the similarity between the past and the future. Of course, the problem is the future can change quickly – the relationship between a good predictor and an outcome can be broken in the future time period. Take a trivial example – mentions of "Obama" on social media in 2008 might have predicted whether or not the Democratic presidential candidate won the election, but this relationship is unlikely to be as powerful in the presidential elections of 2020 when Obama will not be a presidential candidate. Such cases of drift over time in the relationship between the predictors and outcomes will complicate our efforts to tune accurate models.

Interestingly, the act of doing a predictive task itself might create drift in the relationship between the predictors and outcomes, undermining an analysts' predictions by making these predictions. Say that it becomes well known that the words within financial reports of companies predict their future stock market value. If the prediction becomes widely known, either the companies or the investors may change their behavior. Companies might omit or find words that substitute for those that predict negative stock changes. Investors may change the way they sell and buy these stocks. The change in behavior might undermine the prediction, as the relationship between financial reports and the stock market will have changed *because* of the predictions. In these cases, cross validation in the historical data is unlikely to be a good measure of the error rate.

While cross validation might be an accurate and low cost way in many cases to measure the predictive power of a prediction, in the end what really matters is how well the prediction holds up in the real world. Making predictions and then subsequently checking them based on what actually came about is the best measure of accuracy. In some predictive cases such as spam or clicks, when data on outcomes comes quickly, measuring accuracy in real-time by making predictions. In other cases, such as predicting war or elections, an analyst making real-time predictions must wait a long time between observation of outcomes, and it therefore might be infeasible to tune a model based only on real-time predictions.

Misconception 3: It's always better to be more accurate on average. As we stated in the beginning of this chapter, we want to make our predictions as accurate as possible. But what does accuracy actually mean? Take a simple example where we are hoping to predict whether an e-mail is spam, $y_i \in \{1, 0\}$. One way we could approach the problem is to try to optimize average accuracy, or the proportion of time our prediction was correct, $(y_i = \hat{y}_i)/N$.

But accuracy is not always the best metric to use when evaluating a predictive model. To see why, imagine that we are trying to predict a rare event like a genetic disease or the outbreak of World War. Say we had two positive cases of this rare event and 998 negative cases. Then the model would do quite well if we predicted that there would never be war or a genetic disease, we would have accuracy $998/1000 = .998$, or be right 99.8% of the time. But in doing so, we would never predict the outbreak of World War or a genetic disease – the point of the model in the first place. This is what is often known as the *accuracy paradox* – higher accuracy does not always mean a better model.

One way to mitigate the accuracy paradox is to use what is called an F1 score. The F1 score takes into account both *recall* – how many of the actual class were predicted to be of that class – and *precision* – how many of the predicted class were of the actual class. To see this more clearly, consider the confusion matrix below.

Actual Class			
		Positive	Negative
Predicted Class	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Across the columns is the actual class, across the rows is the predicted class. Recall is the number of True Positives divided by the number of True Positives and False Negatives, in other words how many of the actual class positives were predicted to be positive. Precision is the number of True Positives divided by the number of True Positives and False Positives. In other words, how many of the predicted positives were actually positive.

The F1 score trades off precision and recall by multiplying them together. In other words, if either recall and precision get too low, the F1 score will also get low, since it reflects the product rather than the sum of recall and precision.

$$F1 = 2 * \frac{Recall * Precision}{Recall + Precision}$$

But still, F1 might not accurately reflect the costs and benefits of precision and recall to the analyst. For example, we might think that it is much more costly to fail to diagnose a serious disease rather than diagnose a serious disease and turn out to be wrong. In other words, there many be times when we care more about the *recall* of the classifier (how many of the actual diseases we diagnosed), then the *precision* of the classifier (how many of the predicted diseases were actually diseases). There may be other times when we care more about precision than about recall. If we can quantify this cost, by specifying C the number of times recall is more important than precision, then we can update the F1 score to reflect this cost, using:

$$F_C = (1 + C^2) * \frac{Recall * Precision}{Recall + C^2 * Precision}$$

Analysts should also take into consideration the types of units for which they tend to make accurate predictions versus those where predictions tend to be inaccurate. This has come up frequently in the literature on predicting recidivism rates in courts, where some have alleged that prediction algorithms are more accurate for the white majority than for minorities.[2] Many argue that these prediction algorithms are unethical as they may unfairly burden particular groups. In these cases, the distributional costs of prediction should be incorporated in the loss function to make more accurate predictions.

Misconception 4: There is no practical value in interpreting predictions. While in misconception 1 we noted that how you arrive at a prediction is much less important than in causal inference, we present a caveat here that there can be practical advantages of being able to explain predictions. In areas where predictions have consequences, for example, in foreign policy or predictions in medicine, decision makers may view with skepticism algorithms that take in a set of inputs and seemingly magically produce a forecast. It may behoove the analyst to find ways to explain the reasoning behind the algorithm in order to build trust in the algorithm and identify potential pitfalls of the algorithm in conversations with experts.

We note that the point we are making – that there are *practical* advantages to making predictions interpretable – is a very different point than those made in the debate around algorithmic transparency, which deals with whether or not it is *ethical* to make consequential decisions based on opaque algorithmic functioning. While we think this debate is an

[2] Jeff Larson, Surya Mattu, Lauren Kirchner and Julia Angwin. "How We Analyzed the COMPAS Recidivism Algorithm," *ProPublica.* https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

important one to have, our only point here is that for analysts making predictions, explanations of algorithms are of practical value, even if not required by ethics, as decision makers and experts often view predictions with skepticism until they understand their essential components.

4 Using Text as Data for Prediction: Examples

In this section, we review examples of using text for prediction to give the reader a sense of the ways in which text can be used for prediction in the field. Since we have already covered many of the models used for prediction in the measurement chapter, here we focus on using these models to illustrate how they can be used in a prediction context and how this might differ from measurement. We separate out prediction using text as data into four non-exhaustive categories: Source Prediction, or predicting the person or the characteristics of the person who wrote the text, Linguistic Prediction, or forecasting what a person would say or will say next, Social Forecasting, or using text as an input to predict societal phenomena, such as war, the passage of laws, or what the market will do next, and Nowcasting, or using text to measure social phenomena occurring now, such as measuring the incidence of flu, unemployment, or consumer confidence.

4.1 Source Prediction

Our social world is full of text – news, e-mails, letters, books, social media, and articles written by a wide variety of people and many with unknown source or origin. One of the primary ways in which text is used for prediction is to understand the source or the characteristics of the source of some text. While this does not involve prediction into the future, it is a prediction problem if the analyst is hoping to estimate information about the source of the text with the information in and surrounding the text.

Source prediction has already been brought up in several different contexts in this book. One example that we discussed in Chapter 3 is Mosteller and Wallace (1963), uses the text of the Federalist Papers to predict who wrote the Federalist Papers which were not attributed to any particular source. Using filler words, the authors predicted that the papers with unknown authorship were written by Hamilton.

Source prediction is another way to think about spam detection, where the algorithm tries to predict whether an e-mail was sent to a large number of people with the intent of deceiving or making money. Source prediction is often used to detect bad actors on the internet – from malicious websites (Ma et al., 2009), to bots (Ferrara et al., 2016), fake reviews (Li et al., 2014) to online propagandists (King, Pan and Roberts, 2017).

Source prediction is often also used to predict the underlying nature of the text or writer of the post. Tausczik and Pennebaker (2010); Pennebaker and Graybeal (2001) have used texts to predict the author's personality, whether the author is telling the truth (Mihalcea and Strapparava, 2009), or predict the popularity of the author's text (Yano, Cohen and Smith, 2009). Text has more recently been used to try to predict the author's underlying level of depression (De Choudhury et al., 2013; Resnik, Garron and Resnik, 2013). Text of medical records is being used to predict attributes of the person the doctor is writing about – for example, predicting whether or not a patient has a disease or whether or not they are likely to need readmission to the hospital (Rumshisky et al., 2016).

We illustrate source prediction in this chapter drawing inspiration from a recent blog-post written by David Robinson.[3] Robinson's post, which was written during the Trump campaign of 2016, shows that data obtained from Twitter about then candidate Donald Trump's tweets could distinguish between whether Donald Trump himself was tweeting or Donald Trump's staff was tweeting. The secret was that Donald Trump tended to tweet from an Android device, while his staff tweeted from a iPhone device. Tweets from the

[3] http://varianceexplained.org/r/trump-tweets/

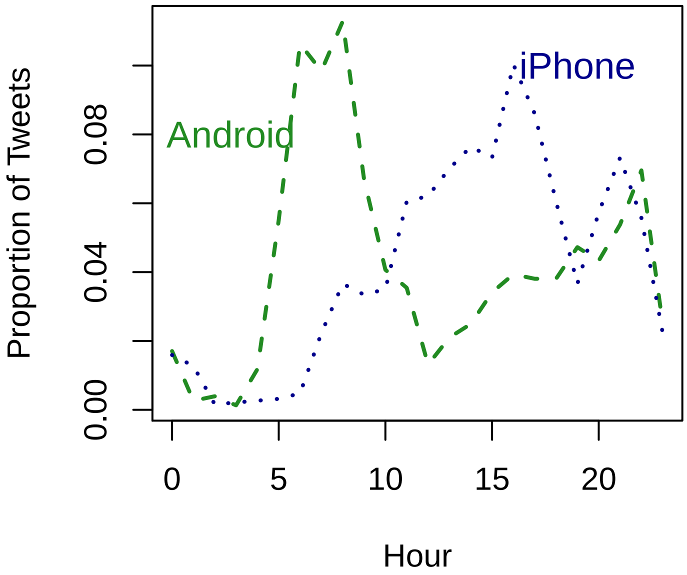

Figure 1: Proportion of Trump's Tweet's from Android vs. iPhone by Hour, Recreated from `http://varianceexplained.org/r/trump-tweets/`

Android device tend to come in the early hours of the morning or late at night, consistent with Trump's schedule, while tweets from the iPhone device tended to come in the middle of the day come from the iPhone, consistent with his staff's schedule, see Figure 1. Robinson's post goes on to show systematic differences between the text of the tweets on the iPhone versus the text of the tweets on the Android, including that those from the Android tend to be more negative in tone.

Sometime in 2017, the now President Trump began tweeting from an iPhone, instead of from an Android device. While previously Trump and his staff could be distinguished, now they can not be. But, we can use what we learned from the campaign Twitter to try to predict which tweets come from Trump and which tweets come from his staff based on the content of the message.[4] This is a classic case of source prediction, as we can use information about known source from before to predict source in the future.

To do this, we took a dataset of 1,390 tweets from Trump's campaign provided by Robinson to train a model that predicts tweets from Android versus tweets from iPhone using only the words within the tweet. Many different classifiers could be fit, for illustration, we fit a Naive Bayes classifer using the word within the tweets of the text as features. We use 10-fold cross validation to choose the words to include within the model.[5] With the final model, we achieve a precision of 0.93 and recall of .80, indicating strong model performance on the historical data. In Table 1 we show the top 20 words most predictive of an Android tweet in the historical data and an iPhone tweet in the historical data. Trump's tweets are associated with words like "win", his own name, and references to CNN and Fox News. Staff tweets are associated with links, thank you's, and hashtags like #makeamericagreatagain and #trump2016.

We then collected 471 new tweets from the most recent Trump feed.[6] Based on the model we created, we predicted which of the 471 tweets came from Trump versus his staff. Of course, even though we achieve very high precision and recall in the historical data, we will not necessarily achieve high precision and recall in current data. The topics that both Trump and his staff tweet about have changed, both because there are constant changes to current events and due to the changing the strategy of the Trump team as they have shifted from campaigning to governance. While we have no way of knowing for certain whether or not our model is indeed good at predicting which tweets are written by Trump and which by his staff, here we do some quick validation to see if the model seems to be doing well based

[4] Others have also done this exercise, see `https://www.theatlantic.com/politics/archive/2017/03/a-bot-that-detects-when-donald-trump-is-tweeting/521127/`, `https://www.wired.com/story/tell-when-someone-else-tweets-from-realdonaldtrump/` and the Twitter handle TrumporNot.

[5] For example, removing punctuation decreased our overall F1 score, while removing stopwords increased it, we therefore included punctuation and removed stopwords.

[6] Data collected on February 16, 2018.

	iPhone	Android
1	/	win
2	https	@realdonaldtrump
3	t.co	”
4	#makeamericagreatagain	hillary
5	#trump2016	@cnn
6	money	last
7	like	trump
8	thank	(
9	jobs	)
10	american	job
11	&	want
12	amp	@foxnews
13	;	even
14	made	rubio
15	back	today
16	soon	much
17	support	’
18	record	good
19	campaign	ever
20	new	tonight

Table 1: Words most predictive of iPhone and Android tweets in historical data.

on what we know about Trump and his Twitter behavior.

First, we look at tweets that are predicted to be highly likely to be written by Trump's staff versus Trump himself. Table 2 shows the five tweets most likely to come from Trump's staff versus most likely to have come from Trump. The algorithm predicts that the staff authored announcements about events or new reports, whereas Trump authored Trump authored tweets about ISIS, votes in the House, and news coverage. Reading these, they conform to our expectations about the types of Tweets the Android used to Tweet in the past versus the types of tweets the iPhone tweeted.

Second, we look at the hourly timing of tweets predicted to be written by Trump's staff versus tweets predicted to be written by Trump. Figure 2 shows the proportion of tweets sent each hour of those predicted to be written by Trump versus predicted to be written

Predicted Staff Tweets	Predicted Trump Tweets
Presidential Proclamation Honoring the Victims of the Tragedy in Parkland, Florida: https://t.co/RTQWAKiSnR https://t.co/Nhs32bm5zB	...Based on that, the Military has hit ISIS "much harder" over the last two days. They will pay a big price for every attack on us!
Join me this Friday in Pensacola, Florida at the Pensacola Bay Center! Tickets: https://t.co/zA7SGgWqBE https://t.co/QzFdxQdgAL	ISIS just claimed the Degenerate Animal who killed, and so badly wounded, the wonderful people on the West Side, was "their soldier." .....
Thank you @SenOrrinHatch. Let's continue MAKING AMERICA GREAT AGAIN! https://t.co/PIv9OAVZcf https://t.co/6egRvuwj1l	Big win today in the House for GOP Tax Cuts and Reform, 227-205. Zero Dems, they want to raise taxes much higher, but not for our military!
Join me live at the 2018 World Economic Forum in Davos, Switzerland! #WEF18 https://t.co/fdHejCmv73 https://t.co/DuKJlVTWB8	Jeff Flake, with an 18% approval rating in Arizona, said "a lot of my colleagues have spoken out." Really, they just gave me a standing O!
New report from DOJ &amp; DHS shows that nearly 3 in 4 individuals convicted of terrorism-related charges are foreign-b? https://t.co/9F2yHphpOi	"90% of Trump 2017 news coverage was negative" -and much of it contrived!@foxandfriends

Table 2: Tweets predicted to be authored by Trump's staff versus Trump.

by his staff. Overall, we see a very similar pattern to that in the historical data shown in Figure 1 – predicted Trump tweets tend to be sent in the very early hours of the morning or the very late hours of the night, and predicted tweets by Trump's staff tend to be sent during the afternoon. However, there are some tweets that are predicted to be written by staff that are sent before 10AM, diverging from the expectation in the historical data. Is this because Trump's staff schedule has changed, or is the model predicting wrong based off of the text? Without a validation set, we might not ever know. However, the model gives us an overall guess as to the continued social media behavior of the President, even after his tweets cannot be separately identified.

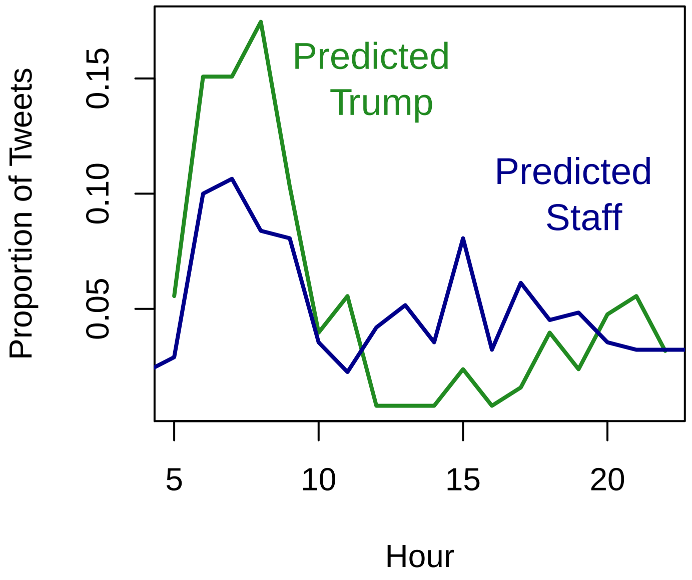

Figure 2: Proportion of Tweet's from Predicted to be Trump versus Predicted to be Staff by Hour, Recent Data

5 Linguistic Prediction

The second category of text prediction we consider is linguistic prediction, or using text data to predict subsequent text data. Can we predict what someone will say next or what someone meant to say? This category is more difficult than source prediction because the predicted outcome is of higher dimension – we do not just want to predict whether Trump or his staff wrote a tweet, we'd like to predict the next tweet, the next word within a tweet, or the way that a particular tweet would be translated.

Linguistic prediction appears in many practical applications that people use every day. Autocorrect is one form of prediction – predicting which words you mis-typed and which you would like to change. Autocomplete in search and text messaging predict what you plan to write even before you write it. Each of these forms of linguistic prediction can be customized to the individual by using what the individual has written or searched for before to optimally predict what they intend to say subsequently.

Question answering is one type of linguistic prediction where the algorithm guesses what will be the most useful answer to a question posed by the user. Usually, instead of generating text from scratch, the computer searches a database of possible answers to find the one or list of results it predicts to be most relevant, then returns these answers to the user. This type of question answering is one example of a larger class of information retrieval tasks where the algorithm tries retrieve information it predicts to be relevant to the user. Famous question answering programs include IBM's Watson and Apple's Siri.

Machine translation is another type of linguistic prediction where the algorithm predicts is how the same text would be expressed in a different language. The most straight forward way to do machine translation is to take each word within a sentence and translate it directly to the opposite language by looking each word up in a dictionary and replacing it with the translated word, a process called *direct translation* (Jursfsky and Martin, 2009, Chapter 25).

However, since the structures of languages vary extensively, direct word by word translation typically lacks *fluency* and can be completely uninterpretable.

While there are deterministic algorithms specific to two languages that can make direct translation more fluent, statistical approaches to machine translation have proven more generally successful at machine translation. The basic idea of many statistical approaches to machine translation is that there is a tradeoff between the fidelity of the translation and the fluency of the translation. Thus, machine translations often jointly optimize a metric of translation quality between languages and the level of fluency of the final result (Jursfsky and Martin, 2009, Chapter 25).

Machine learning is increasingly the approach of choice for doing machine translation. These algorithms take texts each with several expert translations and learn models that most closely produce the expert translation from the source text. Researchers have begun using neural networks as the primary approach to achieving this machine translation; for example, Google now uses neural networks for its tool Google Translate.[7]

[Summary of material to be added: We plan to create a chatbot using a Seq2Seq model to illustrate how one might go about creating a model for generating text based on a transcripts.]

6 Social forecasting

So far we've covered predicting the source of text and automatically generating predictive text. But analysts using natural language processing have also aspired to more lofty goals – such as predicting what will happen in society using the information encoded in text. In this section, we discuss just a few examples of this type of prediction – which we call *social forecasting*.

One of the potentially financially lucrative areas to use text for prediction is using lan-

[7] `https://research.googleblog.com/2016/09/a-neural-network-for-machine.html`

guage to predict markets. Features extracted from business news, message boards, Twitter, and company websites have been used to predict how well the stock market – from the S&P 500 to the Taiwan stock exchange – will fare (For a review, see Nassirtoussi et al. (2014)). Others have set out to more specifically predict company success by using 10-K reports from companies (Kogan et al., 2009) or news stories that specifically mention a particular company (Tetlock, Saar-Tsechansky and Macskassy, 2008). Relatedly, researchers have used social media to predict sales performance (Liu et al., 2007), box office returns (Asur and Huberman, 2010), and consumer confidence (O'Connor et al., 2010).

In the political realm, researchers have used text to predict political outcomes. Aletras et al. (2016) uses the facts described in European human rights court cases to predict the decisions of these cases. Reilly, Richey and Taylor (2012) identifies trends in Google searches for the name and topic of ballot measures to determine which ballot measures in local elections are likely to be subject to "roll-off," or be measures where people abstain from voting on them even though they voted for other measures or candidates on the ballot. Nay (2017) forecasts the probability that bills proposed to Congress will eventually become law, leveraging the text of the bill as data for the prediction.

Using text for prediction also has significant potential for providing forecasts that augment the public good. Tkachenko, Jarvis and Procter (2017) use Flickr tags related to nature to predict where floods are likely to happen in the future. Kang et al. (2013) use features from Yelp reviews to flag restaurants that are likely to have lower hygiene scores, which they suggest could be used to more efficiently audit restaurants to ensure public safety.

We now take a closer look at social forecasting by reviewing in detail Mueller and Rauh (2017) which uses the text of newspaper articles to predict the timing of civil conflict. Mueller and Rauh (2017) use the content of English languages newspaper articles that mention particular countries to forecast whether or not that country will experience civil conflict in the next one to two years. While some variables associated with increased risk of conflict –

like GDP or institutions – are very good at predicting *which* countries are likely to experience civil conflict, the authors note that of similar importance is to predict *when* one particular country will experience civil conflict. Because newspapers provide more granular data about the changes within particular countries, the authors conjecture that newspapers might be better suited to predict the timing of such events.

To make prediction of the timing and location of civil conflict, the authors download 633,835 articles from the *New York Times*, *Economist* and *Washington Post* from 1975-2010 from Lexis Nexis using the name of each of country as a mechanism for selecting articles. The authors run a Latent Dirichlet Allocation 15 topic topic model to estimate the topics in each of these articles. The authors then take the average of the topic vector by country-year to aggregate the topics within each of 185 countries for each year. They then use this estimated share of each topic in each country-year to predict the onset of violence in the 1-2 years after that year using a simple linear fixed effects model:

$$y_{it} = \beta_i + \theta_{it}\beta^{topics} + \epsilon_{it}$$

where y_{it} is the onset of civil conflict in country i in the one to two years following time t, θ_{it} is the vector of aggregated topic proportions for country i at time t. The authors train the model on sets of data between 1975 and time T, varying T from 1995 to 2010, then predict the onset of conflict based on the trained model in time $T+1$ and $T+2$ to evaluate their model out of sample. They find that the newspapers provide additional information about the timing of conflict outside of other variables typically used to predict conflict in countries.

Mueller and Rauh (2017) use a relatively simple model in order to compare their predictions to cross-country regressions typically used in the Political Science literature. However, Mueller and Rauh (2017) could likely get more predictive power out of their model by using their outcome variable to drive feature selection, like we showed in the Chapter 5. [Summary of material to be added: If we can get access to the data, here we plan to do a re-analysis of Mueller and Rauh (2017) to show that more predictive power could be extracted from the model with better feature selection.]

7 Nowcasting

Projecting social forecasts long into the future can be difficult because social phenomena are often influenced by a variety of interacting factors that change quickly with time. Typically, the more that time elapses between the data used for the prediction and the predicted behavior, the more likely predictions are to be inaccurate. In some cases – like predicting civil conflict – predictions are likely to be noisy because they are made one or two years in advance and the causes of such phenomena are complicated and dynamic. In other cases – like predicting roll-off ballots or stock volatility – predictions are made one to two days before an event, allowing analysts to use information closer to the event to make the prediction.

Nowcasting is one specific type of social forecasting that takes the time between the prediction and predicted event to zero, it "predicts" events that are happening *now*, or at the same time as the prediction is being made. While at first glance this may seem to have little practical value, events that are happening in the present are often difficult to measure and official data on such events are unlikely to appear for days or even months after the event actually occurs. Nowcasting is the process of using alternate data to predict measures of the present that will only be released in the future.

Text data are extremely useful for nowcasting because text from social media and news provide a constant stream of data about the world that often reflect underlying phenomena. Social media data have been used to nowcast consumer confidence and unemployment

(O'Connor et al., 2010; Askitas and Zimmermann, 2009), as they often reflect individuals' perceptions of their own livelihood and of the economy. Nowcasting also frequently employs Google search data – the content and frequency of user searches on Google – to predict the level of economic activity such as sales of merchandise, houses, and cars (Choi and Varian, 2012).

Nowcasting is often used when information is prescient, or when major events are unfolding that require a quick response. Nowcasting has been used to monitor and gather data on events such as hurricanes (Preis et al., 2013). It has also been used to predict food poisoning and other food safety outbreaks (Nsoesie, Kluberg and Brownstein, 2014) as well as the fall-out of migrations after natural or manmade disasters (Bengtsson et al., 2011).

Perhaps most famously, nowcasting has been used as a real-time measure of flu activity. Ginsberg et al. (2009) created "Google Flu Trends" – a collaboration between Google and the Center for Disease Control – which used user search data to predict the number of flu-related hospital visits across regions of the United States. The authors used historical user flu data to find 45 searches that were most highly correlated with flu-related doctor visits. They then used metrics of the prevalence of these 45 searches to predict future flu-related doctor visits.

While the initial Google Flu Trends algorithm was very predictive of flu-related doctor's visits and quickly adopted throughout the U.S. and in other countries, the efficacy of the algorithm decreased substantially with time. In 2009, researchers discovered that GFT did not do well at predicting the outbreak of swine flu (Cook et al., 2011). In 2013, GFT's estimates of flu levels were nearly double of actual flu rates (Butler, 2013). Other researchers showed that the utility of GFT's was weak – using past flu data to predict current flu data performed almost equally as well (Goel et al., 2010).

Why was GFT not more successful in leveraging search results to predict the flu? One of the main reasons is that a model trained to nowcast in 2008 needs to be constantly updated to also be accurate in 2009, or 2013. Media coverage of the outbreak of swine flu in 2009 created a very different type of public interest in flu which could have changed the patterns of search results (Cook et al., 2011; Butler, 2013). Further, Google is constantly updating its search algorithms, which impacts user search behavior and search results (Lazer et al., 2014). With so much data to use for prediction – GFT started with 50 million words and phrases that people search on – and so little data to leverage as the outcome of the prediction, in later years GFT may also have picked up on search terms coincidentally related to the timing of the flu season (Lazer et al., 2014). The drift of the relationship between online searches and flu may overwhelm the ability to predict flu without constant supervision and tweaking to the algorithm.[8]

8 How Well Does this Really Work?

[Summary of material to be added: In this section, we plan to test some of the models discussed in previous sections by using these models for realtime prediction. We plan to highlight how drift can make cross-validation a sub-optimal measure for accuracy.]

9 Conclusion

Text data provides us with new opportunities to predict source characteristics, future text, social outcomes, and current measurements. Because the world is producing text in near real-time, leveraging this information can provide us with a more accurate picture of what tomorrow will look like.

Still, prediction is a difficult problem. In linguistic prediction, while it may be relatively straight forward to predict what someone meant to write with a spell checker or predict a

[8] Some have argued that perhaps other types of real-time text data such as data from Twitter would be better able to predict flu trends than search data [*cites to Broniatowski, Paul, and Dredze]. Still, these data are likely to suffer from the same drift problems as search trends [Lazar, Kennedy, King, and Vespinani].

list of endings to a search string, predicting a conversation or other more complex textual interactions is still very difficult. In societal forecasting, prediction of an outcome that will occur days, months, or years after the prediction inputs is likely to be less accurate than predictions that occur more proximate to the data. And in nowcasting, the constant drift between the predictors and outcome can render previously quite accurate models inaccurate. We emphasize in the last part of the chapter that real-time prediction will be the most accurate measure of a model's accuracy, more accurate than cross validation.

In this chapter, we also highlighted the ways in which prediction differs from causal inference and measurement. We show that measurement can be seen as one specific form of prediction – predicting the category in which a text will fall into. However, causal inference and prediction have important differences. Unlike causal inference, where the inferential problem is to predict what *would* have happened, with prediction we focus on what *will* happen. This lends us more flexibility in the types of information we can include within the model. In causal inference we use the split sample as the last step in estimation after the model is already set, to simulate sequential experiments. In prediction, we use the split sample in the process of cross-validation to build our final model. Understanding how the task of prediction differs from other social scientific tasks is essential to understanding how to best leverage text data for each particular problem.

References

Aletras, Nikolaos, Dimitrios Tsarapatsanis, Daniel Preoţiuc-Pietro and Vasileios Lampos. 2016. "Predicting judicial decisions of the European Court of Human Rights: a Natural Language Processing perspective." *PeerJ Computer Science* 2:e93.

Askitas, Nikolaos and Klaus F Zimmermann. 2009. "Google econometrics and unemployment forecasting." *Applied Economics Quarterly* 55(2):107–120.

Asur, Sitaram and Bernardo A Huberman. 2010. Predicting the future with social media. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on.* Vol. 1 IEEE pp. 492–499.

Bengtsson, Linus, Xin Lu, Anna Thorson, Richard Garfield and Johan Von Schreeb. 2011. "Improved response to disasters and outbreaks by tracking population movements with mobile phone network data: a post-earthquake geospatial study in Haiti." *PLoS medicine* 8(8):e1001083.

Butler, Declan. 2013. "When Google got flu wrong." *Nature* 494(7436):155.

Choi, Hyunyoung and Hal Varian. 2012. "Predicting the present with Google Trends." *Economic Record* 88(s1):2–9.

Cook, Samantha, Corrie Conrad, Ashley L Fowlkes and Matthew H Mohebbi. 2011. "Assessing Google flu trends performance in the United States during the 2009 influenza virus A (H1N1) pandemic." *PloS one* 6(8):e23610.

De Choudhury, Munmun, Michael Gamon, Scott Counts and Eric Horvitz. 2013. "Predicting depression via social media." *ICWSM* 13:1–10.

Ferrara, Emilio, Onur Varol, Clayton Davis, Filippo Menczer and Alessandro Flammini. 2016. "The rise of social bots." *Communications of the ACM* 59(7):96–104.

Ginsberg, Jeremy, Matthew H Mohebbi, Rajan S Patel, Lynnette Brammer, Mark S Smolinski and Larry Brilliant. 2009. "Detecting influenza epidemics using search engine query data." *Nature* 457(7232):1012.

Goel, Sharad, Jake M Hofman, Sébastien Lahaie, David M Pennock and Duncan J Watts. 2010. "Predicting consumer behavior with Web search." *Proceedings of the National academy of sciences* 107(41):17486–17490.

Jursfsky, Dan and James Martin. 2009. *Speech and natural language processing: An introduction to natural language processing, computational linguistics, and speech recognition.* Upper Saddle River: Prentice Hall.

Kang, Jun Seok, Polina Kuznetsova, Michael Luca and Yejin Choi. 2013. Where not to eat? Improving public policy by predicting hygiene inspections using online reviews. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.* pp. 1443–1448.

King, Gary, Jennifer Pan and Margaret E Roberts. 2017. "How the Chinese government fabricates social media posts for strategic distraction, not engaged argument." *American Political Science Review* 111(3):484–501.

Kogan, Shimon, Dimitry Levin, Bryan R Routledge, Jacob S Sagi and Noah A Smith. 2009. Predicting risk from financial reports with regression. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics.* Association for Computational Linguistics pp. 272–280.

Lazer, David, Ryan Kennedy, Gary King and Alessandro Vespignani. 2014. "The parable of Google Flu: traps in big data analysis." *Science* 343(6176):1203–1205.

Li, Huayi, Zhiyuan Chen, Bing Liu, Xiaokai Wei and Jidong Shao. 2014. Spotting fake reviews via collective positive-unlabeled learning. In *Data Mining (ICDM), 2014 IEEE International Conference on.* IEEE pp. 899–904.

Liu, Yang, Xiangji Huang, Aijun An and Xiaohui Yu. 2007. ARSA: a sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval.* ACM pp. 607–614.

Lo, Adeline, Herman Chernoff, Tian Zheng and Shaw-Hwa Lo. 2015. "Why significant variables aren't automatically good predictors." *Proceedings of the National Academy of Sciences* 112(45):13892–13897.

Ma, Justin, Lawrence K Saul, Stefan Savage and Geoffrey M Voelker. 2009. Beyond blacklists: learning to detect malicious web sites from suspicious URLs. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM pp. 1245–1254.

Mihalcea, Rada and Carlo Strapparava. 2009. The lie detector: Explorations in the automatic recognition of deceptive language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers.* Association for Computational Linguistics pp. 309–312.

Mosteller, Frederick and David L Wallace. 1963. "Inference in an authorship problem: A comparative study of discrimination methods applied to the authorship of the disputed Federalist Papers." *Journal of the American Statistical Association* 58(302):275–309.

Mueller, Hannes and Christopher Rauh. 2017. "Reading between the lines: Prediction of political violence using newspaper text." *American Political Science Review* pp. 1–18.

Nassirtoussi, Arman Khadjeh, Saeed Aghabozorgi, Teh Ying Wah and David Chek Ling Ngo. 2014. "Text mining for market prediction: A systematic review." *Expert Systems with Applications* 41(16):7653–7670.

Nay, John J. 2017. "Predicting and understanding law-making with word vectors and an ensemble model." *PloS one* 12(5):e0176999.

Nsoesie, Elaine O, Sheryl A Kluberg and John S Brownstein. 2014. "Online reports of foodborne illness capture foods implicated in official foodborne outbreak reports." *Preventive medicine* 67:264–269.

O'Connor, Brendan, Ramnath Balasubramanyan, Bryan R Routledge and Noah A Smith. 2010. "From tweets to polls: Linking text sentiment to public opinion time series." *ICWSM* 11(122-129):1–2.

Pennebaker, James W and Anna Graybeal. 2001. "Patterns of natural language use: Disclosure, personality, and social integration." *Current Directions in Psychological Science* 10(3):90–93.

Preis, Tobias, Helen Susannah Moat, Steven R Bishop, Philip Treleaven and H Eugene Stanley. 2013. "Quantifying the digital traces of Hurricane Sandy on Flickr." *Scientific reports* 3:3141.

Reilly, Shauna, Sean Richey and J Benjamin Taylor. 2012. "Using google search data for state politics research: an empirical validity test using roll-off data." *State Politics & Policy Quarterly* 12(2):146–159.

Resnik, Philip, Anderson Garron and Rebecca Resnik. 2013. Using topic modeling to improve prediction of neuroticism and depression in college students. In *Proceedings of the 2013 conference on empirical methods in natural language processing*. pp. 1348–1353.

Rumshisky, A, M Ghassemi, T Naumann, P Szolovits, VM Castro, TH McCoy and RH Perlis. 2016. "Predicting early psychiatric readmission with natural language processing of narrative discharge summaries." *Translational psychiatry* 6(10):e921.

Shmueli, Galit. 2010. "To explain or to predict?" *Statistical science* pp. 289–310.

Tausczik, Yla R and James W Pennebaker. 2010. "The psychological meaning of words: LIWC and computerized text analysis methods." *Journal of language and social psychology* 29(1):24–54.

Tetlock, Paul C, Maytal Saar-Tsechansky and Sofus Macskassy. 2008. "More than words: Quantifying language to measure firms' fundamentals." *The Journal of Finance* 63(3):1437–1467.

Tkachenko, Nataliya, Stephen Jarvis and Rob Procter. 2017. "Predicting floods with Flickr tags." *PloS one* 12(2):e0172870.

Yano, Tae, William W Cohen and Noah A Smith. 2009. Predicting response to political blog posts with topic models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics pp. 477–485.