Classifying Obstructive Speech in the 1880 British House of Commons

Mitchell Bosley

January 5, 2021

Abstract

When do legislators decide to obstruct the agenda of the legislative majority? Under what conditions do a majority of legislators decide to change the rules to limit the ability of legislators to obstruct? To answer these questions, we first need to be able to *measure* legislative obstruction. In contrast to measures of obstruction that rely on counts of the number of amendments proposed, I argue that the measurement of obstruction is better accomplished by classifying individual speeches as obstructive or not. However, we need to know the context in which a speech was made in order to effectively classify a speech as obstructive or not. Towards this goal, I design a mixture model that incorporates both word use and contextual features. I estimate the model parameters using the EM algorithm, and embed it within an active learning framework, a technique that alternates between human-labeling and model estimation. Preliminary results indicate that the model performs better with metadata features than without.

1 Introduction

Minority obstruction is a ubiquitous feature of legislative politics. In order to frustrate the ability of the legislative majority to implement their preferred policies, minority groups exploit procedural loopholes to drag out the legislative process.

Recently, Goet (2019) and Sieberer et al. (2020) have argued that minority obstruction drives legislative majorities to change the rules of the game in order to limit said obstruction. Analyzing two-hundred years of procedural reform in the British House of Commons, Goet finds inconsistent support for the hypothesis that obstruction leads to anti-minority rule change. However, Goet's measure of the frequency of obstruction in a given legislative session is simply the count of amendments in that session. Sieberer et al. show a positive correlation between obstruction and anti-minority rule change, but their measure for obstruction is simply the distance between the governing and opposition parties' ideal points in a given legislative session, derived from the Comparative Manifesto Project.[1]

While counting the number of amendments is surely a way of detecting one type of obstruction, it should not be the **only** way we have of measuring the amount of obstructive activity that occurs in a given legislative session. And while it sounds reasonable that an increase in ideological distance between governing and minority parties would encourage minority obstruction, it is not justifiable to assume this relationship to be true *a priori*.

To properly estimate the relationship between obstruction and rule changes, we need a measure of obstruction that taps directly into the forum where most obstruction actually occurs: legislative speeches. In the language of machine learning, measuring obstruction at the level of individual

[1] See `https://manifesto-project.wzb.eu/`.

speeches amounts to a classification problem. In the most straightforward approach, given a set of speeches that are labeled as obstructive or not, and a set of features associated with each speech (typically word counts), an algorithm 'learns' the set of weight for the features that minimizes misclassification error. These 'learned' weights are then applied to speeches for which the labels are not provided. Model performance is evaluated by measuring how well the model predicts the 'true' labels of speeches that it has not been trained on.

There are two problems with applying this supervised approach to classification of speeches as obstructive or not. First, this approach to machine learning works well when provided a large number of speeches to learn from. However, to my knowledge there is not an existing repository of legislative speeches labeled as obstructive or not, and manually coding a sufficiently large number of speeches would be time consuming. Second, whether a speech is obstructive or not is not always evident from a context-free inspection of the speech itself. That is, a speech that is clearly obstructive in the context of the overall debate may be difficult to distinguish from a routine speech in the absence of context.

To address the first problem, I employ an active learning approach (see Settles (2011), and also Miller et al. (2020) for a introduction to the method for a political science audience). Rather than randomly labeling speeches as obstructive or not, the active learning process guides the labeller to the speeches that are more likely to be informative to the model. After labelling the new speeches, the model re-estimates the weights for the features. This process continues until the researcher is satisfied with the predictive performance of the model on a held out set of speeches.

The second problem is more pernicious. After all, if we cannot reliably label speeches as obstructive or not simply by looking at them, a speech-based measure of legislative obstruction is not viable. A general solution to this problem is to introduce contextual features into the learning process: if we knew that a legislator making an amendment was doing so in the context of a debate where other members of their party had been speaking for hours, then we might be more inclined to correctly label the speech in question as obstructive.

But how do we best introduce context? One area of cutting edge research in computer science and artificial intelligence is the classification of the intent of each speaker in a conversation, taking the context of the entire conversation into account (see e.g. Wolf et al. (2019), Vedula et al. (2019), and Senese et al. (2020)). These approaches rely on recurrent neural networks (RNN). At its most basic level, a neural network is a universal function approximator: given a set of features and labels, the neural network learns the non-linear function that transforms the features to labels most efficiently. A recurrent neural network basically takes the sequential relationship between observations as feature that it learns from.

While the idea to use RNN models to take conversational context into account is appealing, training RNNs is computationally expensive and requires massive amounts of labeled data to learn from. An alternative approach is to introduce the contextual features of each speech as metadata that, alongside the text features, a model can use to learn the associations between obstructive speech and both text *and* metadata features.

To this end, and building on work from Bosley et al. (2020), I design a Bayesian mixture model where the probability that a speech is obstructive or not as a latent variable, and where each mixture component is a classification category. The probability that a speech is obstructive is a function of both the speech data, modeled as drawn from a multinomial distribution, and the speech-level metadata, modeled as drawn from Bernoulli or Gaussian distributions, depending on the variable type. By the Naive Bayes assumption, I assume all features to be independent. I use the Expectation Maximization (EM) algorithm to find the parameter values that maximize the complete log likelihood of the classes given the speeches and metadata. This model is then wrapped in an active loop, where I iterate between fitting the model and labeling speeches that the model is uncertain about.

This paper proceeds as follows. In Section 2, I describe the structure of the parliamentary speech data that I use in my analysis. I analyze a known period of sustained obstruction in the British House of Commons, and discuss the criteria that I use to determine whether a speech is obstructive or not, noting the difficulty of context-free classification. I then describe the speech-level metadata I use to augment the speech data, as well as my text pre-processing decisions.

In Section 3, I construct the mixture model and derive the EM algorithm that I use to classify speeches. I embed this model within an active learning framework, which I describe. I then outline and justify model specifications.

In Section 4, I discuss my validation strategy, and implement the active learning algorithm. In Section 5 I discuss the results, and consider future steps that I can take to improve model performance.

2 Data and Validation Strategy

The data for this paper comes from Eggers and Spirling (2014).[2] The complete dataset amounts to several hundred thousand speeches from the British House of Commons between 1832 and 1918, and also includes records of electoral returns, offices held by MPs, and party affiliation.

The speech-level data is broken down by parliament, which is divided into sessions, which are further divided into debates. Debates summarize discussion between MPs regarding policy proposals, procedural motions, business of the day, and so on. In this paper, I use the data from the 22nd Parliament, which lasted from 1880 to 1884.

2.1 An Episode of Obstruction

Both Cox (1987) and Dion (1997) identify a period of high obstruction by the Irish secessionist Home Rule party in the 1870s and 1880s. During this period, members of the Home Rule party would make frequent lengthy speeches unrelated to the topic at hand in an effort to slow the ability of the government to implement its legislative agenda. Obstruction in the 22nd Parliament by the Home Rule party was so effective that it directly precipitated the 1882 adoption of the Closure rule in the House of Commons, a rule that allowed a simple majority of Members of Parliament to vote to end debate at any time (Goet, 2019).

During an 1880 debate over the appropriate response of the United Kingdom's House of Commons to hardships that Irish citizens were enduring as a result of a famine the previous year, the Irish-secessionist Home Rule party, led by Charles Parnell, obstructed by repeatedly introducing and debating amendments. First, Parnell or one of his co-partisans would propose an amendment to the government's bill. Then, other members of the Home Rule party would give their own speeches in support of the amendment. After debate on the amendment was exhausted, the proposer of the amendment would allow it to go to a vote or retract the amendment, after which a party member would proposed another amendment, leading to a new round of obstruction. In this fashion, Parnell and the Home Rule party were able to continue the debate for twelve straight hours, eventually forcing the Gladstone-led Liberal government to withdraw key clauses of their proposed bill.[3]

[2] The data can be downloaded at `http://andy.egge.rs/data.html`.

[3] In a later debate, a conservative MP described Parnell's conduct as follows: "[Parnell] accused hon. Members on the Opposition Benches of obstructing this Bill. But what was most properly to be called obstruction: the searching criticism applied to this Bill, or the long, drawling speeches by [Parnell] and his Friends delayed for 12 hours the progress of another Bill which was designed to relieve distress in Ireland, but which had the misfortune not to be drawn up in exact accordance with their ideas?"

Looking to the data, we can verify that the Home Rule party spoke a *lot* during the debate over the Liberal government's version of the relief bill. Table 1 shows the breakdown: members of the Home Rule party uttered more words during the debate than the Conservative and Liberal members combined, despite holding only 63 seats in the 672-person parliament.

There is a pattern to the Home Rule's obstructionary tactics: (1) a member of the members of the party (frequently Parnell) proposes to amend the Government's bill in a lengthy speech; (2) other members of the party make their own speeches lauding the proposed amendment; (3) the Government pushes for the amendment to be formally lodged, hoping to vote it down and move on; and (4) the Home Rule party can formally move the amendment, but more often will withdraw the motion in favour of a new amendment, continuing the cycle of obstruction.

Party	Number of MPs	Words Used in Debate	Number of Speeches
Conservative	237	50784	62
Home Rule	63	176680	160
Liberal	352	77127	95
No Party		1731	3

Table 1: Number of Words Spoken During Debate Over Liberal Government's Ireland Relief Bill, by Party

2.2 Classification Criteria and Metadata

Given the overt nature of the obstruction, one might think that it would be straightforward to identify which speeches during the debate were obstructive and which weren't by simple inspection. However, obstructive speeches are frequently difficult to classify without the context of the entire debate. Consider the following speech by William Parnell:[4]

> [Parnell] moved, after Clause 8, to insert the following Clauses: "The Commissioners of Public Works may, from time to time, out of the said sum of one million five hundred thousand pounds, payable to them by the Commissioners of Church Temporalities, apply any sums not exceeding in all the sum of two hundred and fifty thousand pounds for the purposes of the Fishery Piers Act, to be expended in the manner therein mentioned, but subject to the conditions and exceptions hereinafter mentioned..."

In the naive reading, Parnell is simply proposing an amendment to a bill–a routine occurrence, to be sure. Looking at this speech in the context of the entire debate, however, can help us determine that Parnell's speech is in fact obstructive. We know that Parnell is a member of the Home Rule party, that the debate was lengthy, that Parnell's Home Rule party made an inordinate number of speeches, and that many of those speeches were long. We also know that much of the obstruction involved debating at length on topics only tenuously related to the government's proposal.

By combining information about the speech itself (whether it is amendatory and/or critical of the government, etc.) with contextual information (party of the speaker, topicality of the speech, nature of the debate, etc.) we can correctly identify the speech as obstructive. In the language of machine learning, we can combine information from both speech features and with speech-level metadata.

I define my metadata feature as follows. Party membership is encoded as a series of binary variables, and party size is the proportion of seats that the speaker's party holds. I also include the

[4] The speech is relayed in second person, from the perspective of the speech-recorder in the House of Commons.

logged number of MPs in each party. In addition to the logged number of words in each speech, I include the logged number of the total number words spoken by each MP in each debate, as well as the logged number of all words spoken by each speaker's party during the debate. For the topicality of the speech, I use a variant of the log-odds model from Spirling et al. (2018) that computes the probability that a given speech was made by a member of the majority party. I perform Z-score normalization on this value.

Summary statistics for each these metadata features is shown in Table 2. Figure 1 shows the distribution of the continuous metadata features by party across all debates, while Figure 2 shows the distribution by party for the debate over Irish famine relief referenced above. We can see that the continuous metadata features are for the most part distributed normally. The normality of these features informs my decision in Section 3 to fit Gaussians to the continuously distributed features.

Table 2: Metadata Features

Statistic	N	Min	Max	Mean	St. Dev.
Liberal Party	53,204	0	1	0.438	0.496
Conservative Party	53,204	0	1	0.277	0.448
Home Rule Party	53,204	0	1	0.224	0.417
Num. MPs in Party (Log)	49,989	4.143	5.864	5.336	0.689
Individual Speech Length (Log)	53,204	1.099	11.086	6.184	1.522
Total Words by Party in Debate (Log)	53,204	2.197	12.786	9.987	1.605
Total Words by Speaker in Debate (Log)	53,204	1.609	11.465	8.055	1.501
On-Topic Probability (Z-Score Normalized)	53,204	−1.377	1.487	−0.000	1.000

I combine these metadata features with word-count data from the speech itself. I use a bag-of-words approach with a variety of preprocessing steps (summarized in Table 3) to form a document-feature matrix (DFM).

Table 3: Text Pre-Processing

Parameter	Value
N-grams	1
Stemmer	Snowball (Eng)
Minimum Characters in Word	4
Minimum Speeches for Word	2
Remove Stop Words	Yes
Word Sparsity Lower Bound	0.0001

3 Model

In this section, building on work from Bosley et al. (2020), I outline the structure of a mixture model that combines information from both speeches and speech-level metadata. In 3.1 I describe the parameters of the model, and in 3.2 the classification task. In 3.3 and 3.4 I derive the likelihood function and EM algorithm, respectively. Finally, in 3.5 I describe the active learning loop that the EM algorithm is embedded within.

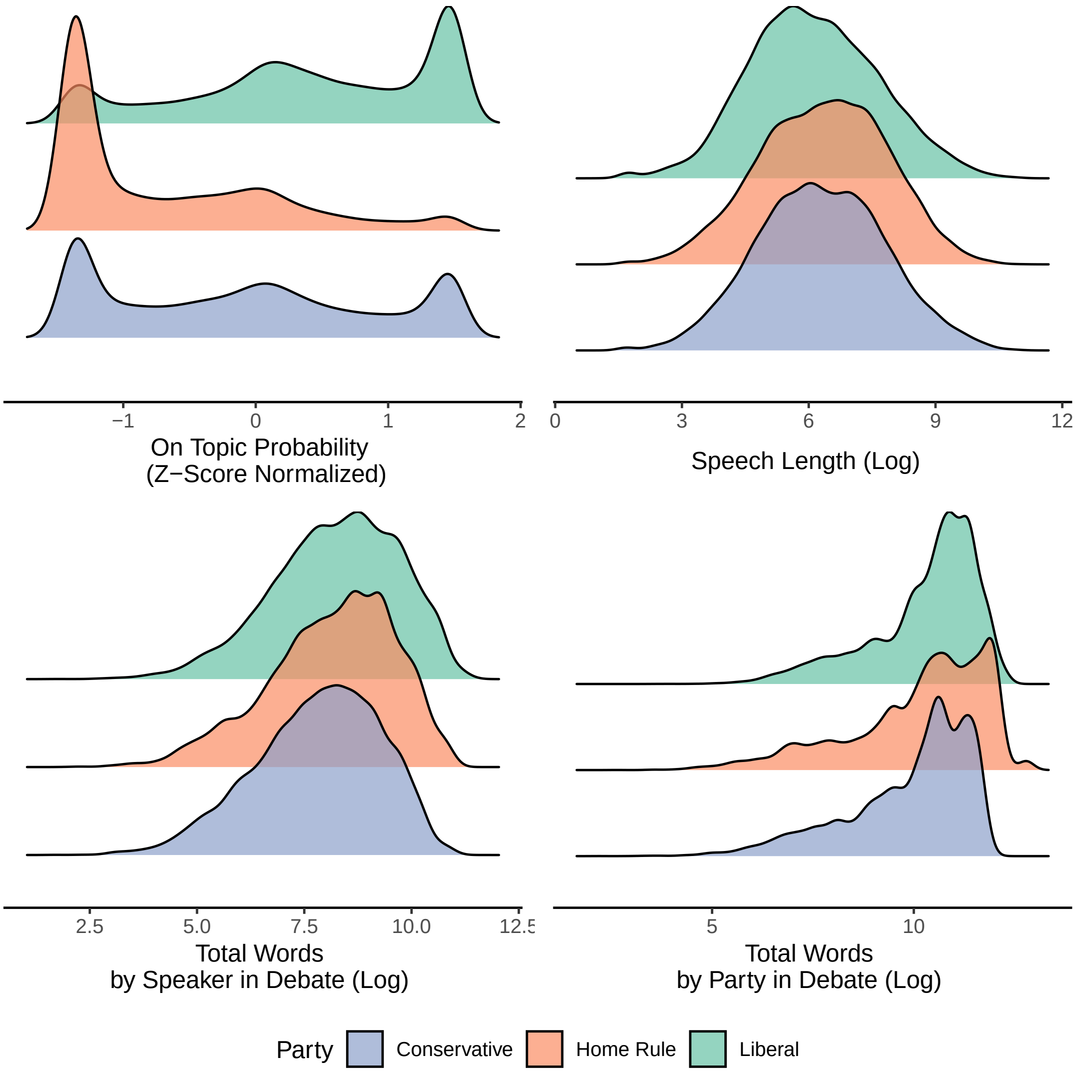

Figure 1: Distribution of Continuous Metadata Variables Across all Debates, by Party

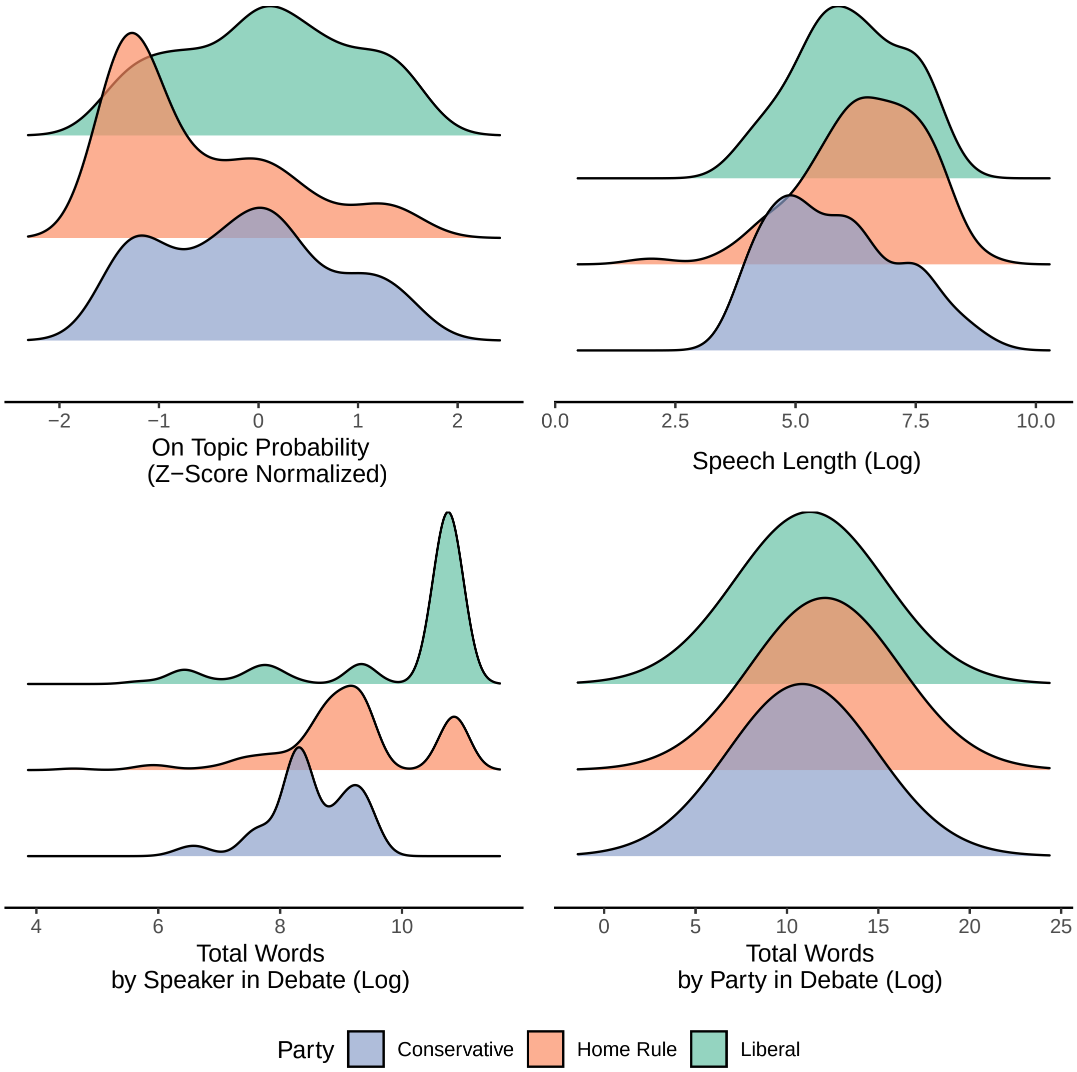

Figure 2: Distribution of Continuous Metadata Variables for Irish Famine Relief Debate, by Party

3.1 Parameters

3.1.1 Class

$$Z_{ij} \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\pi_j) \tag{1}$$

$$\pi_j \sim \text{Beta}(\alpha = 2, \beta = 2) \tag{2}$$

3.1.2 Document

$$D_i \overset{\text{i.i.d.}}{\sim} \text{Multinomial}(n_j) \tag{3}$$

$$\eta_j \overset{\text{i.i.d.}}{\sim} \text{Dirichlet}(\alpha = [2, \ldots, 2]^{\text{T}}) \tag{4}$$

3.1.3 Metadata

1. Binary Variable

$$x_{b_i}|Z_{ij} \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(\psi_j) \tag{5}$$

$$\psi_j \sim \text{Beta}(\alpha = 2, \beta = 2) \tag{6}$$

2. Continuous Variable

$$x_{c_i}|Z_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(x_{c_i}|\mu_j, \sigma_j^2) \tag{7}$$

$$\mu_j \sim \mathcal{N}(\mu_j|\mu_0 = 0, \sigma_0^2 = 1) \tag{8}$$

$$\sigma_j^2 \sim IG(\sigma_j^2|\alpha_0 = 1, \beta_0 = 1) \tag{9}$$

3.2 Classification Task

The classification task is to estimate

$$p(Z_i|\text{Data}_i, \theta) = p(Z_i|D_i, x_{b_i}, x_{c_i}, \theta) \tag{10}$$

where Z_i is a binary variable indicating class, Data_i is the complete set of data associated with a given speech, including D_i, the vector of words making up the speech i, x_{b_i} and x_{c_i} are the values of the binary and continuous metadata variables, respectively, associated with the speech, and θ is the set of parameters that we need to estimate.

In order to compute this probability, we need an estimate of θ. We use the maximum a-posteriori (MAP) estimator such that:

$$\hat{\theta}_{\text{MAP}} = \arg\max_{\theta} p(\theta|\text{Data}) \tag{11}$$

We find this estimate using Bayes' theorem as follows:

$$\begin{aligned} p(\theta|\text{Data}) &= p(\theta|Z^l, d, x_b, x_c) \\ &\propto p(\theta)p(D^l, Z^l, x_b^l, x_c^l|\theta)p(D^u, x_b^u, x_c^u|\theta), \end{aligned} \tag{12}$$

where the first probability function is the prior on the parameters, and the second and third functions are the likelihood for the labeled and unlabeled documents, respectively.

3.3 Deriving the Likelihood

3.3.1 Likelihood of the Labeled Data

Let Z_{ij} be an indicator variable that takes the value of one if the speech i has label j, and the value of zero otherwise. Let $|D^l|$ represent the number of labeled speeches, let V be the vocabulary (the vector of every word found in the corpus) and $|V|$ the length of the number of words in the vocabulary. Finally, let N_{it} represent the number of times word t is used in speech i. Using the Naive Bayes assumption, we can derive the likelihood for the labeled data as follows

$$\begin{aligned}
p(D^l, Z^l, x_b^l, x_c^l|\theta) &= \prod_{i=1}^{|D^l|} p(D_i, Z_i, x_{b_i}, x_{c_i}|\theta) \\
&= \prod_{i=1}^{|D^l|} \prod_{j=0}^{1} p(D_i, Z_{ij} = 1, x_{b_i}, x_{c_i}|\theta)^{Z_{ij}} \\
&= \prod_{i=1}^{|D^l|} \prod_{j=0}^{1} p(D_i, Z_{ij} = 1, x_{b_i}, x_{c_i}|\eta_j, \pi_j, \psi_j, \mu_j, \sigma_j^2)^{Z_{ij}} \\
&= \prod_{i=1}^{|D^l|} \prod_{j=0}^{1} \Big\{ p(D_i|Z_{ij} = 1, n_j) p(x_{b_i}|\psi_j, Z_{ij} = 1) p(x_{c_i}|\mu_j, \sigma_j^2, Z_{ij} = 1) p(Z_{ij} = 1|\pi_j) \Big\}^{Z_{ij}} \\
&\propto \prod_{i=1}^{|D^l|} \prod_{j=0}^{1} \Big\{ \prod_{t=1}^{|V|} \eta_{tj}^{N_{it}} \pi_j \psi_j^{x_{b_i}} (1 - \psi_j)^{1 - x_{b_i}} (\sigma_j^2)^{-1/2} e^{-(1/2\sigma_j^2)(x_{c_i} - \mu_j)^2} \Big\}^{Z_{ij}},
\end{aligned} \tag{13}$$

Now, taking the log yields

$$\begin{aligned}
\log p(D^l, Z^l, x_b^l, x_c^l|\theta) = \sum_{i=1}^{|D^l|} \sum_{j=0}^{1} Z_{ij} \Big\{ & \sum_{t=1}^{|V|} N_{it} \log \eta_{tj} + \log \pi_j \\
& + x_i \log \psi_j + (1 - x_i) \log(1 - \psi_j) - \frac{1}{2} \log \sigma_j^2 - \frac{(x_{c_i} - \mu_j)^2}{2\sigma_j^2} \Big\}
\end{aligned} \tag{14}$$

3.3.2 Likelihood of the Unlabeled Data

We perform a similar derivation for the unlabeled data.

$$\begin{aligned}
p(\text{Data}^u|\theta) &= \prod_{i=1}^{|D^u|} p(D_i, x_{b_i}, x_{c_i}|\theta) \\
&= \prod_{i=1}^{|D^u|} \sum_{j=0}^{1} p(D_i, Z_{ij}, x_{b_i}, x_{c_i}|\theta) \\
&= \prod_{i=1}^{|D^u|} \sum_{j=0}^{1} \Big\{ p(D_i|Z_{ij}, n_j) p(x_{b_i}|Z_{ij}, \psi_j) p(x_{c_i}|Z_{ij}, \mu_j, \sigma_j^2) p(Z_{ij}|\pi_j) \Big\}
\end{aligned} \tag{15}$$

Because of the sum over the parameters of the mixture model, we cannot cleanly take the log of the joint probability of the unlabeled data. However, if we had the labels for the unlabeled

Algorithm 1: EM algorithm to classify text

```
Result: Maximize $p(\{\pi, \eta, \psi, \mu, \sigma^2 \in \theta\} \mid \{D, Z^l, x_c, x_b \in \text{Data}\})$
if In the first iteration of active learning then
    Initialize $\theta^{(0)}$ by Naive Bayes;
else
    Inherit $\theta^{(0)}$ from the previous iteration of active learning;
end
while $p(\theta^{(t)} \mid Data)$ does not converge do
    (1) E step: obtain the probability of the class for unlabeled documents;
      $p(Z^u \mid \text{Data}^u, \theta^{(t)}) \leftarrow \text{E step}(\text{Data}^u, \theta^{(t)})$;
    (2) Combine the estimated and known classes ;
      $p(Z \mid \text{Data}, \theta^{(u)}) \leftarrow \text{combine}(Z^l, p(Z^u \mid \text{Data}^u, \theta^{(t)}))$;
    (3) M step: Maximize $Q \equiv \mathbb{E}[p(Z^u, \theta \mid \text{Data})]$ w.r.t each parameter
      and given $\lambda$ weight to obtain $\theta^{(t+1)}$;
    (4) Check convergence: Obtain the value of $p(\theta^{(t+1)} \mid \text{Data})$;
end
```

documents, computing the complete likelihood would be simple. This leads us to using the EM algorithm, where we alternate between calculating the expected value of the labels for the unlabeled data given the model parameter values (the E step), and calculating the values of the parameters that maximize the probability of the document labels (the M step).

3.4 Deriving the EM Algorithm

Deriving the EM algorithm is straightforward. First we define the Q function, the expectation over the product of likelihoods and priors. Then, we take the derivative of the Q function with respect to each parameter to derive the updating equation for each parameter. Because of the conjugate nature of the priors used in the model, we can derive each updating equation in closed form.

Once we have the Q function and the updating equations, we can simply alternate between performing the E step given the previously updated model parameters, and updating the model parameters given the results of the previous E step, until the complete log likelihood of the parameters stops increasing. This process is summarized in Algorithm 1.

3.4.1 Deriving the Q Function

1. *Joint Posterior Density.* To derive the EM algorithm, we first need to obtain the Q function, the expectation of the complete log joint posterior. Noting that

$$\log p(\theta|\text{Data}) = \log p(\theta, Z^u|\text{Data}) - \log p(Z^u|\theta, \text{Data}), \tag{16}$$

we can define the Q function as

$$\begin{aligned} Q \equiv \mathbb{E}_{Z^u|\hat{\theta}^{\text{old}}}[\log p(\theta|\text{Data})] &= \mathbb{E}_{Z^u|\hat{\theta}^{\text{old}}}[\log p(\theta, Z^u|\text{Data}) - \log p(Z^u|\theta, \text{Data}))] \\ &= \mathbb{E}_{Z^u|\hat{\theta}^{\text{old}}}[\log p(\theta, Z^u|\text{Data})] - \mathbb{E}_{Z^u|\hat{\theta}^{\text{old}}}[\log p(Z^u|\theta, \text{Data}))] \\ &= \mathbb{E}_{Z^u|\hat{\theta}^{\text{old}}}[\log p(\theta, Z^u|\text{Data})] - 0 \\ &= \mathbb{E}_{Z^u|\hat{\theta}^{\text{old}}}[\log p(\theta, Z^u|D, x_b, x_c)], \end{aligned} \tag{17}$$

where

$$\begin{aligned}
p(\theta, Z^u|\text{Data}) &\propto p(\text{Data}|\theta, Z^u)p(\theta, Z^u) \\
&\propto p(D^l, D^u, Z^l, x_b^l, x_c^l, x_b^u, x_c^u|\theta, Z^u)p(\theta, Z^u) \\
&\propto p(D^l, Z^l, x_b^l, x_c^l|\theta) \\
&\times p(D^u|Z^u, \eta)p(\eta) \\
&\times p(x_b^u|Z^u, \psi)p(\psi) \\
&\times p(x_c^u|Z^u, \mu, \sigma^2)p(\mu, \sigma^2) \\
&\times p(Z^u|\pi)p(\pi),
\end{aligned} \tag{18}$$

and by extension

$$\begin{aligned}
\log p(\theta, Z^u|\text{Data}) &\propto \log p(\text{Data}|\theta, Z^u)p(\theta, Z^u) \\
&\propto \log p(D^l, Z^l, x_b^l, x_c^l|\theta) \\
&+ \log p(D^u|Z^u, \eta)p(\eta) \\
&+ \log p(x_b^u|Z^u, \psi)p(\psi) \\
&+ \log p(x_c^u|Z^u, \mu, \sigma^2)p(\mu, \sigma^2) \\
&+ \log p(Z^u|\pi)p(\pi),
\end{aligned} \tag{19}$$

We know the value of $\log p(D^l, Z^l, x_b^l, x_c^l|\theta)$ from 14, and we derive the other expressions as follows.

2. *Probability of the Unlabeled Documents and Document Probability.*

$$\begin{aligned}
p(D^u|Z^u, \eta)p(\eta) &= p(\eta) \prod_{i=1}^{|D^u|} p(D_i|Z_i, \eta) \\
&= \prod_{j=0}^{1} p(\eta_j) \prod_{i=1}^{|D^u|} p(D_i|Z_i, \eta) \\
&= \prod_{j=0}^{1} p(\eta_j) \prod_{i=1}^{|D^u|} p(D_i|Z_i = 1, \eta)^{Z_{ij}} \\
&= \prod_{j=0}^{1} \left\{ \prod_{t=1}^{|V|} \eta_{tj} \prod_{i=1}^{\|D^u\|} \prod_{t=1}^{|V|} \eta_{tj}^{Z_{ij}N_{ti}} \right\}
\end{aligned} \tag{20}$$

Taking the log yields

$$\log p(D^u|Z^u, \eta)p(\eta) = \sum_{j=0}^{1} \left\{ \sum_{t=1}^{|V|} \log \eta_{tj} + \sum_{i=1}^{|D^u|} \sum_{t=1}^{|V|} Z_{ij} N_{ti} \log \eta_{tj} \right\}. \tag{21}$$

3. *Class Probability and Unlabeled Document Probability.*

$$\begin{aligned}
p(Z^u|\pi)p(\pi) &= p(\pi) \prod_{i=1}^{|D^u|} p(Z_i|\pi) \\
&= \prod_{j=0}^{1} p(\pi_j) \prod_{i=1}^{|D^u|} p(Z_{ij}=1|\pi_j)^{Z_{ij}} \\
&= \prod_{j=0}^{1} \pi_j \prod_{i=1}^{|D^u|} \pi_j^{Z_{ij}} = \prod_{j=0}^{1} \pi_j \pi_j^{\sum_{i=1}^{|D^u|} Z_{ij}} \\
&= \prod_{j=0}^{1} \pi_j^{1+\sum_{i=1}^{|D^u|} Z_{ij}}
\end{aligned} \tag{22}$$

Taking the log yields

$$\log p(Z^u|\pi)p(\pi) = \sum_{j=0}^{1} \left(1 + \sum_{i=1}^{|D^u|} Z_{ij}\right) \log \pi_j. \tag{23}$$

4. *Probability of Binary Metadata.*

Because $\psi_j \sim \mathrm{Beta}(2,2)$,

$$p(x_b^u|Z^u,\psi)p(\psi) = \prod_{j=0}^{1} \psi_j(1-\psi_j) \prod_{i=1}^{|D^u|} (\psi_j^{x_{b_i}}(1-\psi_j)^{1-x_{b_i}})^{Z_{ij}} \tag{24}$$

Taking the log yields

$$\log p(x_b^u|Z^u,\psi)p(\psi) = \sum_{j=0}^{1} \log \psi_j + \log(1-\psi_j) + \sum_{i=1}^{\|D^u\|} Z_{ij}\big\{(x_{b_i}) \log \psi_j + (1-x_{b_i}) \log(1-\psi_j)\big\} \tag{25}$$

5. *Probability of Continuous Metadata.*

$$\begin{aligned}
p(x_c^u|Z^u,\mu,\sigma^2)p(\mu,\sigma^2) &= \prod_{j=0}^{1} p(\mu_j,\sigma_j^2) \prod_{i=1}^{|D^u|} p(x_{c_i}^u|Z_{ij}=1,\mu_j)^{Z_{ij}} \\
&\propto \prod_{j=0}^{1} (\sigma^2)^{-5/2} e^{(-\mu_j^2-1)/2\sigma_j^2} \prod_{i=1}^{|D^u|} ((\sigma_j^2)^{-1/2} e^{-(1/2\sigma_j^2)(x_{c_i}-\mu_j)^2})^{Z_{ij}} \\
&\propto \prod_{j=0}^{1} (\sigma_j^2)^{-\frac{\sum_{i=1}^{|D^u|} Z_{ij}}{2} - 5/2} e^{\frac{-\mu_j^2 - 1 - \sum_{i=1}^{|D^u|} Z_{ij}(x_{c_i}-\mu_j)^2}{2\sigma^2}} \\
\log p(x_c^u|Z^u,\mu,\sigma^2)p(\mu,\sigma^2) &\propto \sum_{j=0}^{1} \left(-\frac{\sum_{i=1}^{|D^u|} Z_{ij}}{2} - 5/2 \right) \log \sigma_j^2 + \frac{-\mu_j^2 - 1 - \sum_{i=1}^{|D^u|} Z_{ij}(x_{c_i}-\mu_j)^2}{2\sigma_j^2}
\end{aligned} \tag{26}$$

3.4.2 E Step

We can now reform the Q function as follows:

$$
\begin{aligned}
Q &\equiv \mathbb{E}_{Z^u|\hat{\theta}^{\text{old}}}[\log p(\theta, Z^u|D, x_b, x_c)] \\
&= \mathbb{E}_{Z^u|\hat{\theta}^{\text{old}}}\Bigg[\sum_{i=1}^{|D^l|}\sum_{j=0}^{1} Z_{ij}\Bigg\{\sum_{t=1}^{|V|} N_{it}\log\eta_{tj} + \log\pi_j \\
&+ x_{b_i}\log\psi_j + (1-x_{b_i})\log(1-\psi_j) - \frac{1}{2}\log\sigma_j^2 - \frac{(x_{c_i}-\mu_j)^2}{2\sigma_j^2}\Bigg\} \\
&+ \sum_{j=0}^{1}\Bigg\{\sum_{t=1}^{|V|}\log\eta_{tj} + \sum_{i=1}^{|D^u|}\sum_{t=1}^{|V|} Z_{ij}N_{ti}\log\eta_{tj}\Bigg\} + \sum_{j=0}^{1}\Bigg(1+\sum_{i=1}^{|D^u|} Z_{ij}\Bigg)\log\pi_j \\
&+ \sum_{j=0}^{1}\log\psi_j + \log(1-\psi_j) + \sum_{i=1}^{|D^u|} Z_{ij}\big\{(x_i)\log\psi_j + (1-x_i)\log(1-\psi_j)\big\} \\
&+ \sum_{j=0}^{1}\Bigg(-\frac{\sum_{i=1}^{|D^u|} Z_{ij}}{2} - 5/2\Bigg)\log\sigma_j^2 + \frac{-\mu_j^2 - 1 - \sum_{i=1}^{|D^u|} Z_{ij}(x_{c_i}-\mu_j)^2}{2\sigma_j^2}\Bigg]
\end{aligned}
\tag{27}
$$

3.4.3 M Step

We can now find the parameter update functions by maximizing Q with respect to π_j, η_{tj},ψ_j, μ_j, and σ_j^2. Additionally, define $\Lambda(i)$ to be a weighting factor λ when d_i is in the unlabeled set of documents:

$$
\Lambda(i)\begin{cases}\lambda & \text{if } D_i \in D^u \\ 1 & \text{if } D_i \in D^l\end{cases}
\tag{28}
$$

1. *Maximizing w.r.t.* π_j. First, form a Lagrange as follows

$$
L_{\pi_j} = Q - \lambda_{\pi_j}\Bigg(\sum_{j=0}^{1}\pi_j - 1\Bigg).
\tag{29}
$$

Then, taking the derivative and setting it to zero

$$
\begin{aligned}
\frac{\partial L_{\pi_j}}{\partial \pi_j} &= \frac{\partial Q}{\partial \pi_j} - \lambda_{\pi_j} \\
&= \frac{\sum_{i=1}^{|D^l|} E[Z_{ij}]}{\pi_j} + \frac{1+\sum_{i=1}^{|D^u|} E[Z_{ij}]}{\pi_j} - \lambda_{\pi_j} \\
&= \frac{1+\sum_{i=1}^{|D|} E[Z_{ij}]}{\pi_j} - \lambda_{\pi_j} = 0
\end{aligned}
\tag{30}
$$

Then, noting that $\frac{\partial L_{\pi_j}}{\partial \pi_j} = \sum_{j=0}^{1}\pi_j - 1 = 0$, we can solve for π_j such that

$$
\begin{aligned}
\hat{\pi}_j &= \frac{1+\sum_{i=1}^{|D|} E[Z_{ij}]}{\sum_{j=0}^{1}(1+\sum_{i=1}^{|D|} E[Z_{ij}])} \\
&= \frac{1+\sum_{i=1}^{|D|} E[Z_{ij}]}{2+|D|}
\end{aligned}
\tag{31}
$$

Adding the λ weighting yields

$$\hat{\pi}_j = \frac{1 + \sum_{i=1}^{|D|} \Lambda(i) E[Z_{ij}]}{2 + |D^l| + \lambda |D^u|} \tag{32}$$

2. *Maximizing w.r.t.* η_{tj}. Noting that $\sum_{t=1}^{|V|} \eta_{tj} = 1$, we can form a Lagrange such that

$$L_{\eta_{tj}} = Q - \lambda_{\eta_{tj}} \big(\sum_{t=1}^{\|V\|} \eta_{tj} - 1 \big) \tag{33}$$

We can now take the derivatives with respect to λ and η and set to 0.

$$\begin{aligned} \frac{\partial L_{\eta_{tj}}}{\partial \eta_{tj}} &= \frac{\partial Q}{\partial \eta_{tj}} - \lambda_{\eta_{tj}} \\ &= \frac{\sum_{i=1}^{|D^l|} E[Z_{ij}] N_{it}}{\eta_{tj}} + \frac{1}{\eta_{tj}} + \frac{\sum_{i=1}^{|D^u|} E[Z_{ij}] N_{it}}{\eta_{tj}} - \lambda_{\eta_{tj}} \\ &= \frac{1 + \sum_{i=1}^{|D|} E[Z_{ij}] N_{it}}{\eta_{tj}} = 0 \\ \frac{\partial L_{\eta_{tj}}}{\partial \lambda_{\eta_{tj}}} &= 0 \end{aligned} \tag{34}$$

Solving for η_{tj} yields

$$\begin{aligned} \hat{\eta}_{tj} &= \frac{1 + \sum_{i=1}^{|D|} E[Z_{ij}] N_{it}}{\sum_{t=1}^{|V|} (1 + \sum_{i=1}^{|D|} E[Z_{ij}] N_{it})} \\ &= \frac{1 + \sum_{i=1}^{|D|} E[Z_{ij}] N_{it}}{|V| + \sum_{t=1}^{|V|} \sum_{i=1}^{|D|} E[Z_{ij}] N_{it}} \end{aligned} \tag{35}$$

Adding the λ weights yields:

$$\hat{\eta}_{tj} = \frac{1 + \sum_{i=1}^{|D|} \Lambda(i) E[Z_{ij}] N_{it}}{|V| + \sum_{t=1}^{|V|} \sum_{i=1}^{|D|} \Lambda(i) E[Z_{ij}] N_{it}} \tag{36}$$

3. *Maximizing w.r.t.* ψ_j.

$$\begin{aligned} \frac{\partial Q}{\partial \psi_j} &= \frac{\sum_{i=1}^{|D^l|} E[Z_{ij}] x_{b_i}}{\psi_j} - \frac{\sum_{i=1}^{|D^l|} E[Z_{ij}](1 - x_{b_i})}{1 - \psi_j} \\ &\quad + \frac{1}{\psi_j} + \frac{\sum_{i=1}^{|D^u|} E[Z_{ij}] x_{b_i}}{\psi_j} - \frac{1}{1 - \psi_j} - \frac{\sum_{i=1}^{|D^u|} E[Z_{ij}](1 - x_{b_i})}{1 - \psi_j} \\ &= \frac{1 + \sum_{i=1}^{|D|} E[Z_{ij}] x_{b_i}}{\psi_j} - \frac{1 + \sum_{i=1}^{|D|} E[Z_{ij}](1 - x_{b_i})}{1 - \psi_j} = 0 \end{aligned} \tag{37}$$

Solving for ψ_j yields

$$\hat{\psi}_j = \frac{1 + \sum_{i=1}^{|D|} E[Z_{ij}] x_{b_i}}{2 + \sum_{i=1}^{|D|} E[Z_{ij}]} \tag{38}$$

and adding the λ weighting yields

$$\hat{\psi}_j = \frac{1 + \sum_{i=1}^{|D|} \Lambda(i) E[Z_{ij}] x_{b_i}}{2 + \sum_{i=1}^{|D|} \Lambda(i) E[Z_{ij}]} \tag{39}$$

4. *Maximizing w.r.t.* μ_j.

$$\begin{aligned}
\frac{\partial Q}{\partial \mu_j} &= \frac{\partial}{\partial \mu_j}\frac{-\sum_{i=1}^{|D^l|} E[Z_{ij}](x_{c_i}-\mu_j)^2 - \mu_j^2 - 1 - \sum_{i=1}^{|D^u|} E[Z_{ij}](x_{c_i}-\mu_j)^2}{2\sigma_j^2} \\
&= \frac{\partial}{\partial \mu_j}\frac{-\mu_j^2 - 1 - \sum_{i=1}^{|D|} E[Z_{ij}](x_{c_i}-\mu_j)^2}{2\sigma_j^2} \\
&= -\mu_j + \sum_{i=1}^{|D|} E[Z_{ij}]x_{c_i} - \sum_{i=1}^{|D|} E[Z_{ij}]\mu_j = 0
\end{aligned} \tag{40}$$

Solving for μ_j yields

$$\hat{\mu}_j = \frac{\sum_{i=1}^{|D|} E[Z_{ij}]x_{c_i}}{1 + \sum_{i=1}^{|D|} E[Z_{ij}]} \tag{41}$$

and adding the λ weighting yields

$$\hat{\mu}_j = \frac{\sum_{i=1}^{|D|} \Lambda(i)E[Z_{ij}]x_{c_i}}{1 + \sum_{i=1}^{|D|} \Lambda(i)E[Z_{ij}]} \tag{42}$$

5. *Maximizing w.r.t.* σ_j^2

$$\begin{aligned}
\frac{\partial Q}{\partial \sigma_j^2} &= \frac{\partial}{\partial \sigma_j^2}\frac{-\sum_{i=1}^{|D^l|} E[Z_{ij}](x_{c_i}-\mu_j)^2 - \mu_j^2 - 1 - \sum_{i=1}^{|D^u|} E[Z_{ij}](x_{c_i}-\mu_j)^2}{2\sigma_j^2} \\
&+ \frac{\partial}{\partial \sigma_j^2}\frac{-\sum_{i}^{|D^l|} E[Z_{ij}] - \sum_{i}^{|D^u|} E[Z_{ij}] - 5}{2}\log\sigma_j^2 \\
&= \frac{\partial}{\partial \mu_j}\frac{-\mu_j^2 - 1 - \sum_{i=1}^{|D|} E[Z_{ij}](x_{c_i}-\mu_j)^2}{2\sigma_j^2} + \frac{-\sum_{i}^{|D|} E[Z_{ij}] - 5}{2}\log\sigma_j^2 \\
&= \frac{1}{2\sigma^2}\left(\frac{1}{\sigma^2}\big(\sum_{i=1}^{|D|} E[Z_{ij}](x_{c_i}-\mu_j)^2 - \mu_j^2 - 1\big) - \big(\sum_{i=1}^{|D|} E[Z_{ij}] - 5\big)\right) = 0
\end{aligned} \tag{43}$$

Solving for σ_j^2 yields

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{|D|} E[Z_{ij}](x_{c_i}-\mu_j)^2 + \mu_j^2 + 1}{\sum_{i=1}^{|D|} E[Z_{ij}] + 5} \tag{44}$$

and adding the λ weighting yields

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{|D|} \Lambda(i)E[Z_{ij}](x_{c_i}-\mu_j)^2 + \mu_j^2 + 1}{\sum_{i=1}^{|D|} \Lambda(i)E[Z_{ij}] + 5} \tag{45}$$

3.5 Active Learning

The active learning algorithm can be split up into the following distinct steps outlined in Algorithm 2: initialization with a small number of randomly labeled documents, estimation of the probability that each unlabeled document belongs to the positive class, selection of the unlabeled documents that the model is most uncertain about, and labeling of the selected documents by a human coder. The algorithm then iterates until it is reaches one of a pre-specified set of thresholds.

Algorithm 2: Active learning with EM algorithm to classify text

```
Result: Obtain the predicted classes for all documents up to a chosen level of certainty.
Initialize $D^l_{old}$ by sampling some documents randomly, and have humans label them ;
Initialize $D^u \leftarrow D \setminus D^l_{old}$;
while Not all documents are classified with some certainty yet do
    (1) Predict labels for each document in $D^u$ using Algorithm 1;
    (2) Sample $k$ most uncertain documents in $D^u$ and have humans labels them;
        $D^l_{new} \leftarrow k$ most uncertain documents in $D^u$;
    (3) Update labeled and unlabeled documents;
        $D^l_{old} \leftarrow D^l_{old} \cup D^l_{new}$;
        $D^u \leftarrow D \setminus D^l_{old}$;
end
```

4 Results

4.1 Model Specifications and Validation Strategy

Table 4 summarizes the model specifications. The small amount of data used is due to the difficulty I had in labeling speeches as obstructive or not from the roughly 50,000 speeches in the 22nd Parliament. As a result, for this preliminary analysis I decide to stick with the Irish famine debate. I hand-labeled 107 speeches for the held-out testing set, and used the remaining 210 to train the model. I initialized the model with 10 randomly chosen documents, and then performed 10 active learning steps, labeling 5 speeches with the highest cross-entropy at each step. I set the lambda weighting value to 0.001 following advice from Bosley et al. (2020). This has the effect of introducing a small amount of information from both the word and metadata features associated with the unlabeled speeches in the training set.

Table 4: Model Specification

Parameter	Value
Training Speeches	210
Testing Speeches	107
# of Init. Speeches	10
Speeches Labeled per Iter	5
Total Labeled Speeches	60
Active Sampling Type	cross-entropy
Lambda Value	0.001

4.2 Model Performance

To assess classification performance, I fit the learned model to the held out testing dataset compute accuracy, precision, recall, and F1 scores.[5] Figure 3 compares each of these statistics across the active learning iterations for the model that combines speech-level features with the binary and continuously distributed metadata features. Despite an early dip in the recall score, all statistics are trending in the right direction.

[5] The F1 score is defined as the harmonic mean of the precision and recall scores.

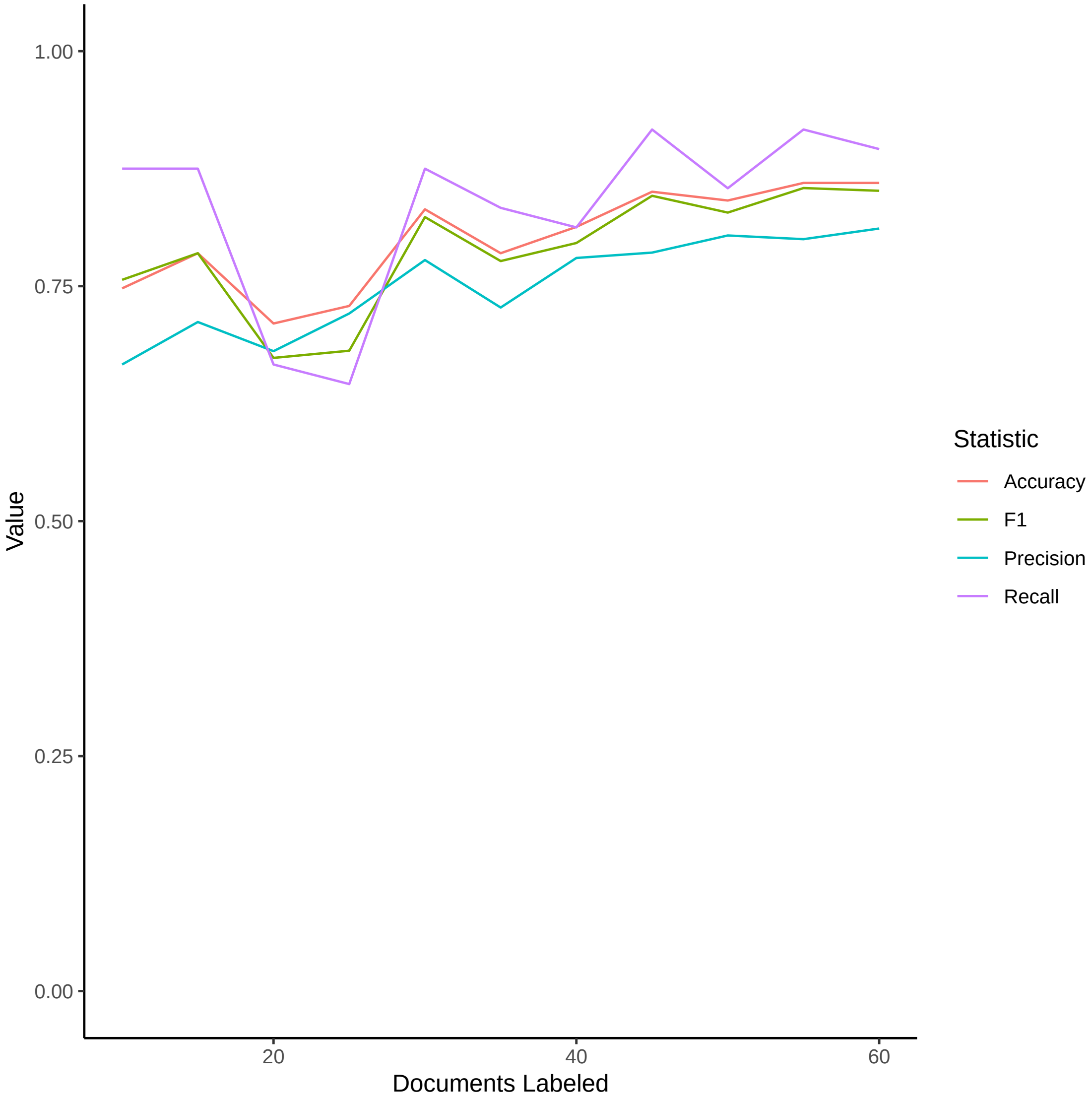

Figure 3: Model Performance Across Active Learning Iterations: All Metadata

Table 5 compares model performance at 60 documents labeled depending on the type of feature included. The first row describes the model that does not include any metadata features (that is, it includes only the word features); the second includes both word features and *binary* metadata features; the third includes both word features and *continuous* metadata features; and the last row includes word features and both types of metadata features. This table shows that performance across all metrics improves as both binary and continuous metadata is added to the model.

Table 5: Comparing Model Performance With and Without Metadata

Model	Accuracy	Precision	Recall	F1
No Metadata	0.607	0.571	0.500	0.533
Binary Metadata Only	0.822	0.809	0.792	0.800
Cont. Metadata Only	0.785	0.821	0.667	0.736
All Metadata	0.860	0.824	0.875	0.848

Tables 6 and 7 shows the parameter values for the learned model from Figure 3. Each cell in Table 6 is the value of ψ_j from the mixture model, the probability that a speech belongs to one of the two classes given the feature in question is equal to one. We can see that being in the Home Rule party is associated with a high probability of belonging to the obstructive class, while the opposite is true for membership in the Liberal or Conservative parties. I also included a feature that tracks whether the word 'amend' appeared at least once in a given speech, and it skews slightly toward the obstructive class.

Table 6: Binary Metadata Feature Parameters

Feature	Non-Obstructive	Obstructive
Liberal	0.645	0.010
Conservative	0.318	0.019
Home Rule	0.045	0.981
Amendment	0.045	0.067

Table 7 is structured similarly, except each cell provides the mean (with variance in parentheses) of a gaussian distribution (μ_{j} and $\sigma^2{}_{\mathrm{j}}$ from the mixture model). The continuous variable with the biggest difference between classes is the standardized on-topic score: speeches that are less likely to be on topic are more likely to be obstructive.

Table 7: Continuous Metadata Feature Parameters

Feature	Non-Obstructive	Obstructive
Speech Length (Log)	6.231 (24.080)	6.626 (29.374)
MPs in Party (Log)	5.623 (30.369)	4.116 (62.679)
On Topic Prob (Std.)	-0.026 (123.889)	-0.679 (160.987)
Num. Words By Speaker in Debate	9.488 (2.825)	9.032 (9.164)
Num. Words by Party in Debate	11.050 (0.089)	11.953 (0.038)

5 Discussion and Conclusion

In this paper, I have introduced a metadata-augmented speech classification algorithm, wrapped in an active learning process, with the goal of developing a speech-based classifier of legislative obstruction. While the results I have shown provide preliminary support for the effectiveness of this model, It is important to remember that I have implemented this model in the most granular of settings: a single debate of about 300 speeches. To truly demonstrate the use of this method, I need to expand the analysis, first to multiple debates within the 22nd parliament, then to the entire parliament, and then to multiple parliamentary sessions.

The difficulty I encountered in labeling randomly chosen documents from the dataset, even with the help of contextual metadata features, is remains a problem. One solution may be to farm out the initial labeling of the speeches to British parliamentary history experts. Another (potentially complementary) approach could be to expand the classification options in order to identify a variety of types of legislative speech, some of which could be considered obstructive.

It would also be worthwhile to expand the metadata-augmented classification approach. One intriguing possibility would be to encode hierarchical features of the data into the mixture model, although this would likely involve at least the partial abandonment of the Naive Bayes assumption that makes the model tractable.

Given the wealth of legislative speech data that has become available over the last decade, it is clear that applying machine learning approaches to analyze this data is a profitable path for future research.

References

Bosley, M., Kuzushima, S., Enamorado, T., and Shiraito, Y. (2020). Improving Probabilistic Models in Text Classification via Active Learning. In *American Political Science Association 2020*.

Cox, G. W. (1987). *The Efficient Secret: The Cabinet and the Development of Political Parties in Victorian England*. Political Economy of Institutions and Decisions. Cambridge University Press, Cambridge ; New York.

Dion, G. (1997). *Turning the Legislative Thumbscrew: Minority Rights and Procedural Change in Legislative Politics*. University of Michigan Press, Ann Arbor, MI.

Eggers, A. C. and Spirling, A. (2014). Electoral Security as a Determinant of Legislator Activity, 1832-1918: New Data and Methods for Analyzing British Political Development: Electoral Security. *Legislative Studies Quarterly*, 39(4):593–620.

Goet, N. D. (2019). The Politics of Procedural Choice: Regulating Legislative Debate in the UK House of Commons, 1811–2015. *British Journal of Political Science*, pages 1–19.

Miller, B., Linder, F., and Mebane, W. R. (2020). Active Learning Approaches for Labeling Text: Review and Assessment of the Performance of Active Learning Approaches. *Political Analysis*, pages 1–20.

Senese, M. A., Rizzo, G., Dragoni, M., and Morisio, M. (2020). MTSI-BERT: A Session-aware Knowledge-based Conversational Agent. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 717–725, Marseille, France. European Language Resources Association.

Settles, B. (2011). *Synthesis Lectures on Artificial Intelligence and Machine Learning : Active Learning*. Morgan & Claypool Publishers, San Rafael.

Sieberer, U., Dutkowski, J. F., MEIßNER, P., and Müller, W. C. (2020). 'Going institutional' to overcome obstruction: Explaining the suppression of minority rights in Western European parliaments, 1945-2010. *European Journal of Political Research*, pages 1475–6765.12376.

Spirling, A., Huang, L., and Patrick, P. (2018). Boring in a New Way: Estimation and Inference for Political Style at Westminster, 1935–2018. *SSRN Electronic Journal*.

Vedula, N., Lipka, N., Maneriker, P., and Parthasarathy, S. (2019). Towards Open Intent Discovery for Conversational Text. *arXiv:1904.08524 [cs]*.

Wolf, T., Sanh, V., Chaumond, J., and Delangue, C. (2019). TransferTransfo: A Transfer Learning Approach for Neural Network Based Conversational Agents. *arXiv:1901.08149 [cs]*.