Estimating the Number of Unique words used in BBC Hindi and Aaj Tak

Paul Hunt

Febuary 15, 2022

Executive Summary

Note: I am currently working on some code to match Hindi text to a dictionary of Arabic words. If I manage to successfully implement that, then I will be using Arabic loanwords in the analysis instead of all unique words, but I don't think the methods should be affected too much.

This paper employs words from articles scraped from the front pages of the Hindi language news websites BBC Hindi and Aaj Tak to estimate the total number of unique words that might be used on each website and answer the question "Which website has a larger lexicon?"

The analysis begins with simple linear models of the logged word counts against their frequencies, hoping to extrapolate the additional number of unused words from the intercept, but the models suffered from poor fit and violations of the assumption of linearity rendering them wholly inappropriate for inference. Then, a models are fit based upon the assumption of a multinomial sampling distribution. These models seem to reflect the observed patterns in the data much better, and inference proceeds from there.

We find that the proportion of unobserved words (relative to the number of unique words in the sample) is much greater for Aaj Tak than for the BBC, but this finding was always likely since the sample size of individual words from Aaj Tak was much smaller. Based on the posterior predictive distribution from our multinomial model we find that the total number of words used by the BBC is almost certainly greater than the number used by Aaj Tak.

These results may suggest a first step in informing some critical media studies of the differences in language and style between international news sources and domestic Indian sources, but the conclusions in this paper are too weak to be extrapolated beyond the sample.

Full Analysis

Definitions

For the purposes of this analysis, we are interested in the frequencies of unique word frequencies, which may lead to some confusion, so I open this description with some definitions as employed in this paper:

- *Individual words* refers to observations of particular instances of the words, including repetitions of previously observed words.
- *Unique words* refers to the words observed without counting repetitions, such that we may sample multiple individual observations of a unique word.
- *Single words* refers to the words that are observed individually only once.
- *Frequency* will hereafter denote the number of individual observations of a unique word, and

Table 1: Ten most frequent words in the BBC and AT samples

BBC			Aaj Tak		
Word	Frequency	Definition	Word	Frequency	Definition
	2589	Gen. particle (pl.)		1073	Gen. particle (pl.)
	1762	in		895	in
	1723	is		694	is
	1457	Gen. particle (f.)		689	Gen. particle (f.)
	1165	and		541	to
	1004	to		481	from
	996	from		394	and
	851	are		389	Past tense particle
	793	Past tense particle		372	Gen. particle (m.)
	753	on		347	are

- *Frequency counts* will denote the number of unique words observed at a given frequency.

Data Set

My data set consists of the words in news articles scraped from the front pages of the BBC Hindi and Aaj Tak websites over the five days from December 5, 2021 to December 9. I have collected $n_{BBC} = 67,702$ individual words from the BBC, and $n_{AT} = 30,956$ individual words from Aaj Tak, with $k_{BBC} = 5,359$ unique words from the BBC and $k_{AT} = 4,099$ unique words from Aaj Tak.Both of these websites cover a wide variety of topics from Indian domestic politics and world events to sports and entertainment news, ensuring a sample of words specific to many domains.

To ensure that the samples of words from the two sources are reflective of our expectations, Table 1 gives the ten most frequent unique words in the sample for each website. As expected, the observations of high frequency words are very sparse with large gaps between the frequencies and no repeated frequencies. Additionally, the high frequency words are all grammatical particles, postpositions (the Hindi equivalent of English prepositions), forms of the copula *hona* ("to be"), and conjunctions.

Note on table 1: I have not been able to get unicode to render in the table... I'm still working on that.

After observing the highest frequency words, we want to check the distribution of lower frequency words and get a sense of the overall distribution of frequency counts in the dataset. Figure 1 plots the number of unique words with frequencies between 1 and 100, showing that the counts decrease at a rate approximating exponential decay. With most frequencies after 25 containing counts of either one or zero, and the gaps between ones growing larger at higher frequencies.

After this cursory examination of the data, we turn to methods of modeling the count of unique words at frequency zero, which represents the number of words in the lexicon of each source not captured by our sample.

Linear Regression

Given the apparent exponential decay in in the data, a linear model of logged values for frequency counts (plus one to account for zeros) versus the the logged frequencies may be a reasonable initial model. In fact, a scatterplot of the logged values (Figure 2) seems to show a very strong linear relationship in the first half of the distribution, although this breaks down considerably as the counts greater than zero become more sparse. A well-fitting linear model would enjoy the benefit of including a parameter β_0 for the intercept, which we could interpret as an estimate for the number of words at frequency zero.

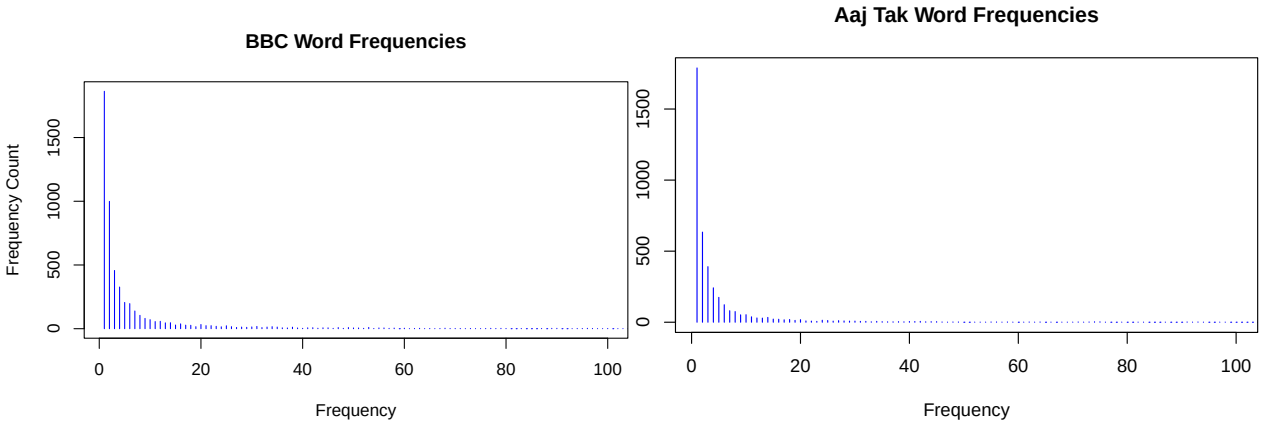


Figure 1: Frequency counts of the first 100 frequencies. This plot has been truncated at 100 to illustrate the the observed pattern in frequency counts, but the frequencies extend beyond 1,000 for both samples.

Table 2: Posterior coefficient estimates

	Mean Coef.	S.D.	2.5%	97.5%
BBC				
Intercept	2.532	0.053	2.427	2.636
Slope	-0.354	0.008	-0.369	-0.339
AT				
Intercept	3.224	0.086	3.054	3.392
Slope	-0.510	0.014	-0.538	-0.482

I have elected to implement my regression with Zellner's g-prior (code given in the appendix), with $g = \max(\text{Frequency} > 0)$ (i.e. the number of observations in the model), $\sigma_0^2 = 2$, and $\nu_0 = 2$. Posterior estimates are given in Table 2.

The slopes are negative, as anticipated, but a quick glance at figure 2 shows that the model underestimates the intercepts by far. This is most likely due to the high concentrations of zero-counts at the higher frequencies shrinking both the slope and intercept estimates. by exponentiating these intercepts, we get 95% credible estimates for $Y_{0,\text{BBC}} = (11.33, 13.96)$, and $Y_{0,\text{AT}} = (21.21, 29.72)$, which are not at all reasonable.

The posterior distributions of the slopes and intercepts show strong correlation, which is to be expected in a two-parameter linear model, and the MCMC diagnostic plots (in Appendix A) show good convergence of the Markov chain with autocorrelation not extending beyond the first lagged value.

Next, consider re-parameterizing our model to include a changepoint for a discontinuous regression model.

Table 3: Intercept Posterior Quantiles and Estimate of N Unique Words from Log-Linear Chagepoint Model

	Intercept		Estimated N	
	BBC	AT	BBC	AT
2.5%	66.20	20.33	5425.20	4119.33
50%	2327.62	1809.74	7686.62	5908.74
97.5%	73395.13	2206.91	78754.13	6305.91

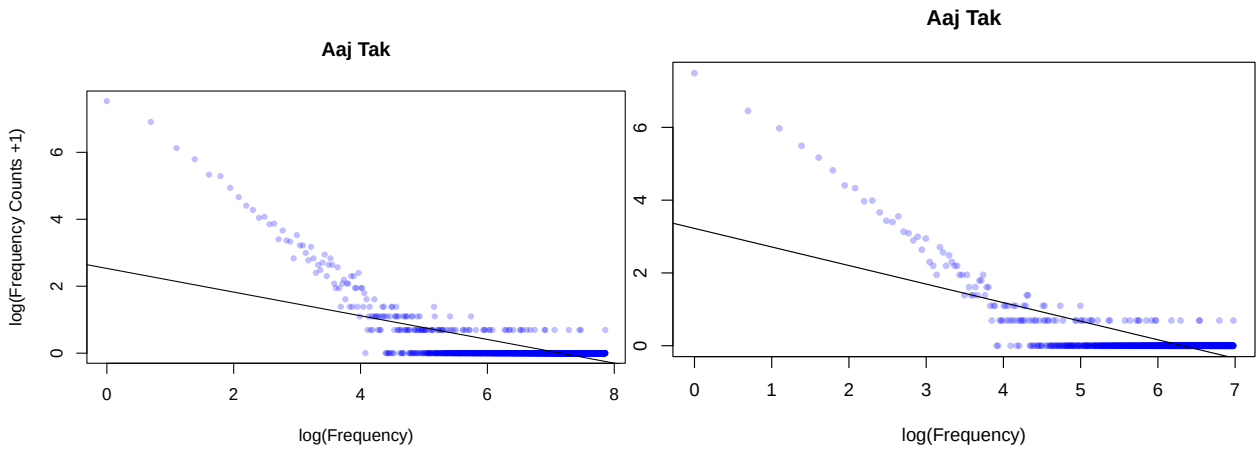


Figure 2: Scatter plots of log(frequency counts) vs. log(frequency) with mean lines of best fit of each source.

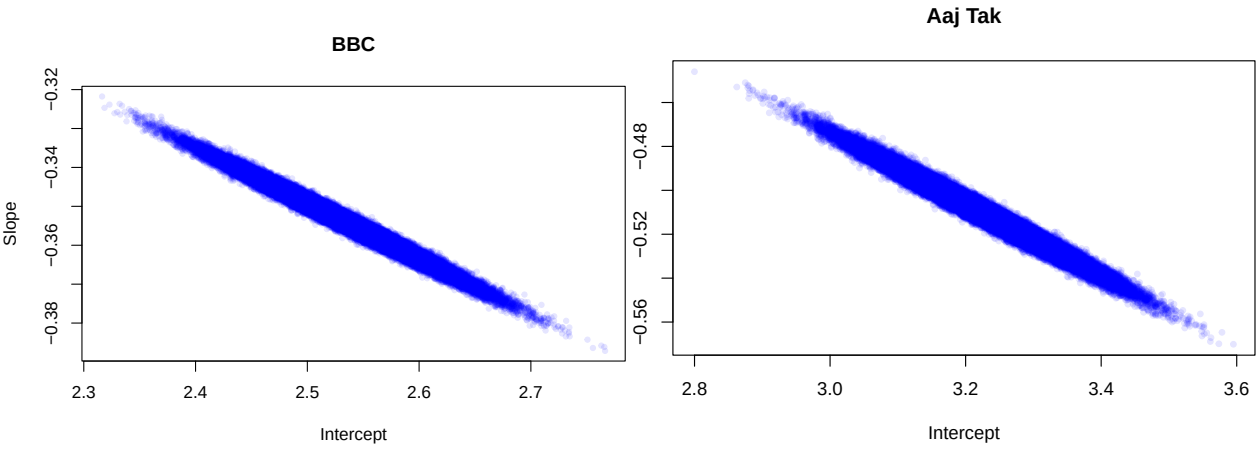


Figure 3: Posterior distributions of slope and intercepts for BBC and AT.

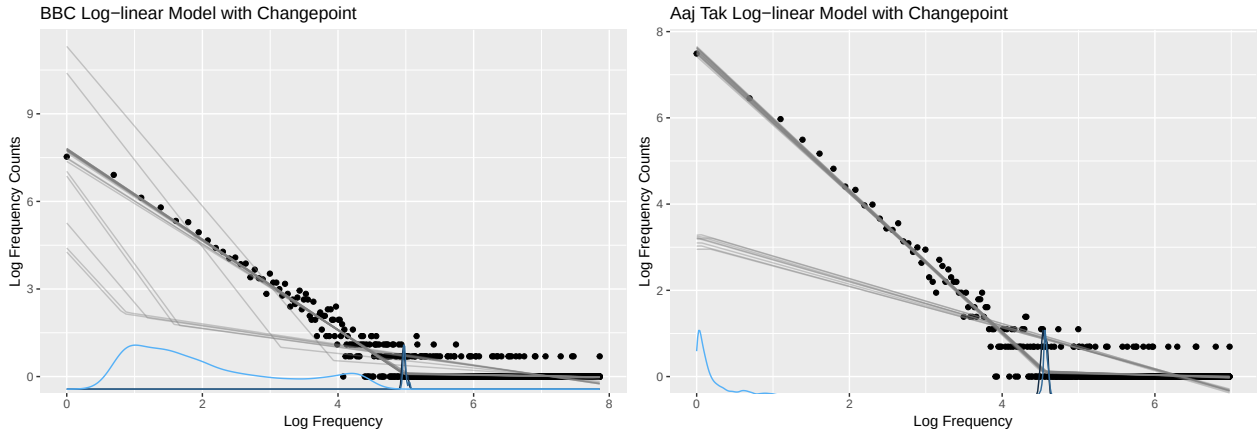


Figure 4: Log-Linear Changepoint models

When the model is fit from the "mcp" package with one changepoint, we find the probability of BBC containing more undiscovered unique words than Aaj Tak is 0.914. However, the posterior distributions for the intercepts (in table 4) show large variance and do not allow for a precise estimate of the total number of unique words. Additionally, there is no theoretical justification for or interpretation a discontinuity. So next, we will turn to a multinomial distribution to represent the sampling distribution of the frequency counts.

Note: I am somewhat inclined to continue developing this model with the log-linear smoother, but maybe with some more detailed prior elicitation for the changepoint to improve the fit. I might try to implement the model myself since mcp feels like a bit of a black box to me and it is rather slow. Right now I am planning to use this and the subsequent models on the data from December, then check again with a new test set of scraped articles to compare methods.

Dirichlet-Multinomial Model

A popular estimator for the coverage of a sample of discrete species (in this case unique words) is the Good-Turing estimator, which assumes that the proportion of unobserved words in the population is similar to the proportion of single words in the sample. This estimate is typically refined with some smoother, but here we will take a more naive approach letting $p_0 = p_1$. Hereafter, let:

- K be the number of unique words in a sample,
- i indicate frequency,
- Y_i denote the count of unique words appearing i times in the sample,
- p_i indicate the proportion of unique words occurring i times, and
- n be the sample size of individual words.

We can assume $Y|n, k, p \sim \text{Multinomial}(n, k, p)$, or that frequency counts are assigned randomly to each frequency based on that particular frequency's p_i. Then, since the multinomial distribution is a multivariate generalization of the binomial distribution, p is distributed according to the Dirichlet distribution –the multivariate generalization of the beta distribution, or $p \sim \text{Dirichlet}(\alpha)$, where α is a vector of weights of length $\max(i)$.

Thus, we can derive a conjugate prior from:

- $p(Y \mid p, n) = \prod_{i=1}^{n} p_i^{Y_i}$
- $p(p) = \text{Dirichlet}(\alpha) = \frac{1}{\beta(\alpha)} \prod_{i=1}^{n} p_i^{\alpha_i - 1}$
- $p(p \mid Y) \propto \prod_{i=1}^{n} p_i^{Y_i} \prod_{i=1}^{n} p_i^{\alpha_i - 1}$

Which is the kernel of $\text{Dirichlet}(\alpha_i + Y_i)$

In this analysis, I have set α to length n in order to leave some weight on the unlikely possibility that only one unique word is sampled (i.e. $Y_n = 1$). After an initial trial with the flat prior $\alpha_i = 1$, which put too much weight on rare frequencies giving very small estimates of p_1. I have chosen instead to set $\alpha_i = \frac{1}{i}$ to approximate the rapid decay we expect to see in our observations.

I have built a Monte Carlo sampler to draw $3,000$ posterior samples from $p \mid Y$ based on the algorithm:

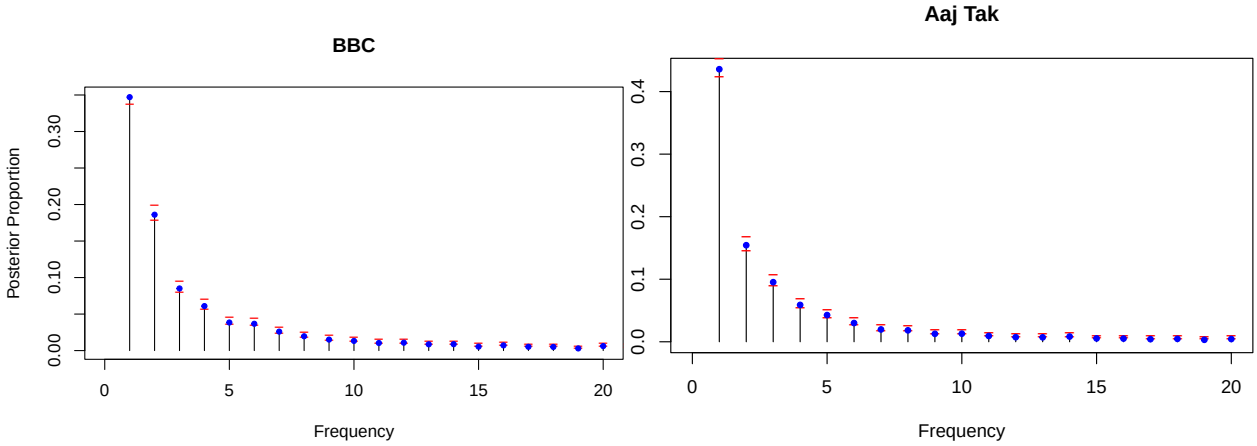
- $\gamma_i \sim \text{Gamma}(\alpha + Y_i, 1)$
- $p_i = \frac{\gamma_i}{\sum_{i=1}^{n} \gamma_i}$

Table 4: First 5 posterior p_i for each sample.

	p_1	p_2	p_3	p_4	p_5
BBC					
2.5%	0.335	0.176	0.078	0.055	0.034
mean	0.347	0.186	0.085	0.061	0.038
97.5%	0.359	0.197	0.093	0.068	0.044
Aaj Tak					
2.5%	0.421	0.143	0.086	0.052	0.037
mean	0.436	0.154	0.095	0.059	0.043
97.5%	0.451	0.166	0.104	0.066	0.049

Table 5: Posterior predictive means and quantiles for Y0, Y1, and N.

	Y_0	Y_1	N
BBC			
2.5%	1,764.00	1,761.00	7,123.00
mean	1,859.25	1,858.44	7,218.25
97.5%	1,953.03	1,955.00	7,312.02
Aaj Tak			
2.5%1	1,697.00	1,700.97	5,796.00
mean1	1,787.30	1,786.83	5,886.30
97.5%1	1,875.00	1,876.00	5,974.00



\begin{figure}
\caption{Posterior means of the first 20 pi, with 95% error bars.} \end{figure}

Figure 4 illustrates the posterior distributions of the first 20 p_i for each source with means and 95% credible intervals, and Table 3 gives the same numbers for the first 5 p_i. $p_{1,\text{AT}}$ is clearly higher than $p_{1,\text{BBC}}$, with $Pr(p_{1,\text{AT}} > p_{1,\text{BBC}}) \approx 1$ from these samples. If we treat p_1 as an estimator for p_0, this is perfectly reasonable, since the source with the smaller sample size should have captured fewer of the possible unique words.

Following the intuition of this estimator, we can build a posterior predictive distribution by sampling $\tilde{Y}$ from Multinomial$(\tilde{Y} \mid p, Y)$, then estimate N, the total number of possible unique words, by sampling $\tilde{Y}_0$ from Binomial$(\tilde{Y}_0 \mid k, p_1)$. Table 4 gives posterior predictive ranges for $\tilde{Y}_0, \tilde{Y}_1$ and the predicted value of N. We find that despite the higher proportion of undiscovered words in Aaj Tak, the BBC has a higher number of predicted total unique words and based on Monte Carlo integreation of this sample, $Pr(N_{\text{BBC}} > N_{\text{AT}}) \approx 1$. It is interesting to note that the estimated numbers of unobserved unique words are very similar, and

$Pr(Y_{0,\text{AT}} > Y_{0,\text{BBC}}) = 0.13$

Note: at this point I've read three articles relating the Good-Turing estimator of discovery probability to Poisson-Dirichlet processes as well as the Bayesian Nonparametrics textbook (ed. Nils Lid Hjort et. al.), and I've been taking a stochastic processes course this semester so the math is becoming much more clear in my head, but I'm really feeling stuck on an implementation. Otherwise, I'm struggling to think of a good smoother other than the log-linear model to infer p_0.

Discussion

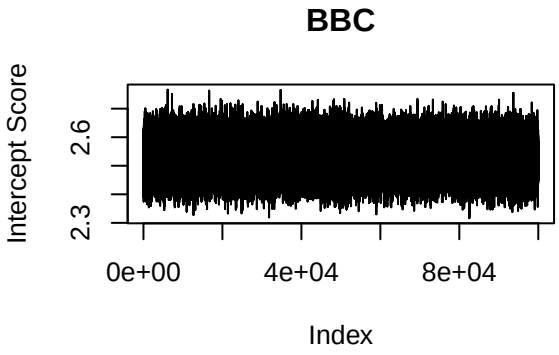
We have found that a log-linear model is insufficient to estimate the number of unobserved words due to the sparsity of high frequency observations. While it may be desirable to improve the model fit by adding a discontinuity since the interpretation of the intercept intercept as the number of unobserved unique words is convenient, there is no reason to suspect that any such linear relationship truly exists, and I am suspicious of including model parameters with no real meaning.

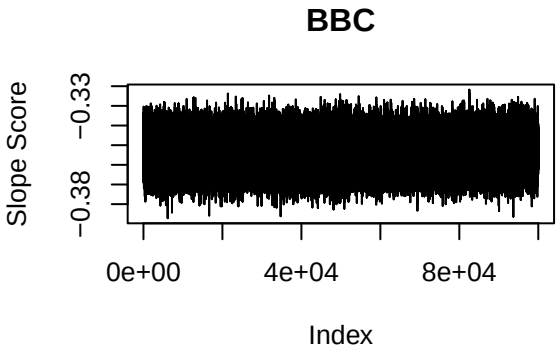
On the other hand, it is far more reasonable to assume that the frequencies of unique words can be sampled from a multinomial distribution. This does assume that the occurrence of the words are independent of one another –which is certainly not true of natural language– but in the long run, the multinomial distribution does reflect the patterns we would expect to see. Our analysis of the posterior predictive distributions of word frequencies on the two websites lead us to conclude that the BBC probably employs a larger lexicon of unique words than Aaj Tak.

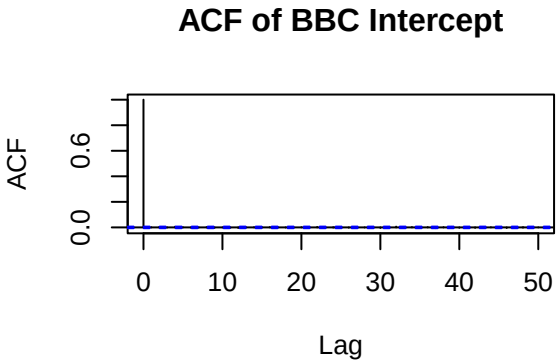
The multinomial model suffers from a dependence on K, the number of unique words in the sample, in the posterior predictive distribution which is difficult to overcome. We cannot treat K as an unknown parameter, since it is a linear combination of the frequency counts and therefore not random once the data are known. Additionally, it is only observed once in each sample, making it impossible for us to explore its variation. Future work might employ hierarchical models with a more diverse dataset. For example, the samples might be divided into categories based on the topics of the articles in which the words originally appeared in order to understand how K can change. Otherwise, the data could be examined from a nonparametric perspective employing Poisson-Dirichlet process models to analyse the discovery probability without making assumptions about the dimensions of p or the hyper parameters.

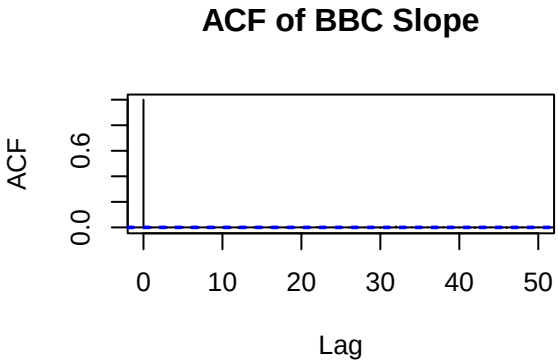
Appendecies

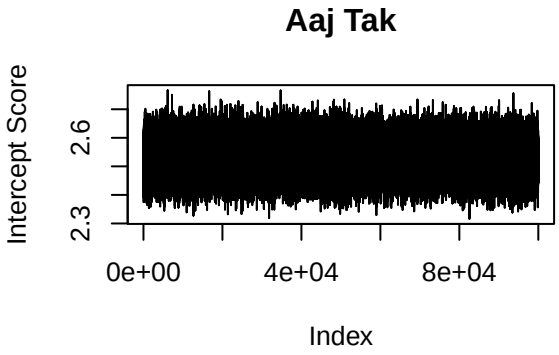
Appendix A: Supplamental Figures

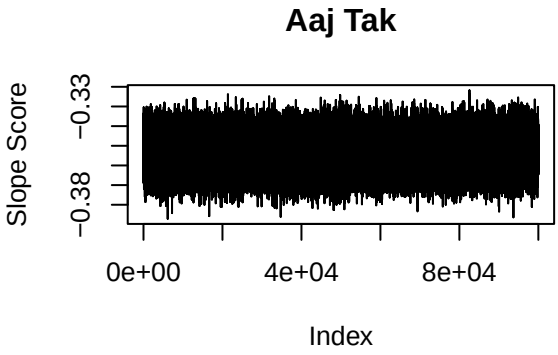


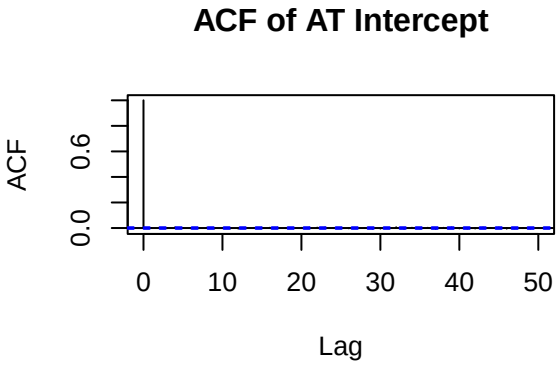


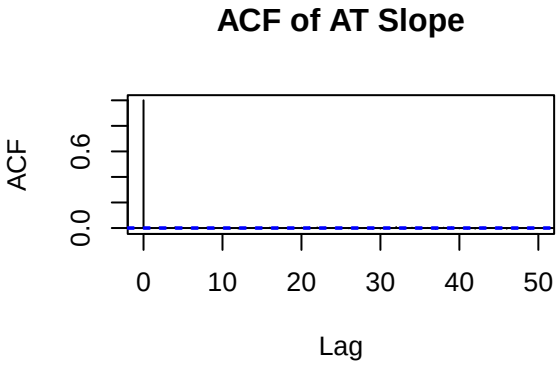












Appendix B: R Codes

Data Preparation

```
format.file <- function(filename){
  file <- read.delim(filename,
                     encoding = 'UTF-8',
                     quote = "")

  vec <- unlist(file)
  names(vec) <- NULL

  words <- unlist(strsplit(vec, split = " "))

  nopunct <- gsub("[[:punct:][:blank:]]+",
                  " ", words)
  clean_words <- unlist(strsplit(nopunct, split = " "))
  nospace <- gsub(rawToChar(as.raw("0xa0")),
                  "", clean_words)
  Hindi_words <- gsub("([A-Z]*[a-z]*[0-9]*)*",
                      "", nospace)


  str_subset(str_split(Hindi_words, " "), ".+")
}
```

```
AT_files <- paste("Aaj_Tak1/",list.files("Aaj_Tak1"), sep = "")
BBC_files <- paste("BBC1/",list.files("BBC1"), sep = "")

AT <- unlist(sapply(AT_files[1:5], format.file))
names(AT) <- NULL
BBC <- unlist(sapply(BBC_files[1:5], format.file))
names(BBC) <- NULL
```

```
freq_table <- function(counts){
  freq <- numeric(max(counts))
  for(i in 1:max(counts)){
    freq[i] <- sum(counts == i)
  }
  freq
}
```

Linear Model

```
prior_BBC <- list(g = length(BBC_freqs),
                  nu0 = 2,
                  s20 = 2)
prior_AT <- list(g = length(AT_freqs),
                  nu0 = 2,
                  s20 = 2)

BBC_lm <- zellnor.lm(log(BBC_freqs+1), log(1:length(BBC_freqs)),
                     prior_BBC)
AT_lm <- zellnor.lm(log(AT_freqs+1), log(1:length(AT_freqs)),
                     prior_AT)
```

Dirichlet-Multinomial Model

```
my_prior <- function(n){
  alpha <- 1/1:n
  alpha
}

BBC_prior <- my_prior(n_BBC)
AT_prior <- my_prior(n_AT)
```

```
DM_post <- function(freqs, alpha_0 = 1, reps = 3e+3){
  k <- sum(freqs)
  m <- length(freqs)
  n <- sum(freqs*1:m)

  freqs <- c(freqs, rep(0, n-m))

  gam_post <- matrix(rgamma(reps*n, alpha_0+freqs, 1),
                     ncol = n, nrow = reps,
                    byrow = T)
  P_post <- matrix(numeric(n*reps),
                    ncol = n, nrow = reps)
  for(i in 1:reps){
      P_post[i,] <- gam_post[i,]/sum(gam_post[i,])
    }
  P_post
}


BBC_post <- DM_post(BBC_freqs, BBC_prior)
AT_post <- DM_post(AT_freqs, AT_prior)
```

```
DM_post_pred <- function(freqs, post){
  k <- sum(freqs)
  m <- length(freqs)
  n <- sum(freqs*1:m)

  Y_post <- matrix(numeric(n*nrow(post)), byrow =T,
                    nrow = nrow(post),
                    ncol = ncol(post))
  Y_0 <- numeric(nrow(post))
  for(i in 1:nrow(post)){
    Y_post[i,] <- rmultinom(1, k, post[i,])
    Y_0[i] <- rbinom(1,k,post[i,1])
  }
  N_post <- rowSums(Y_post)+Y_0
  list(N = N_post, Y = Y_post, Y_0 = Y_0)
}

BBC_post_pred <- DM_post_pred(BBC_freqs, BBC_post)
AT_post_pred <- DM_post_pred(AT_freqs, AT_post)
```