Handout 12: Case Study with US State of the Union Data

Taylor Arnold

Loading and parsing the data

The full text of all the State of the Union addresses through 2016 are available in the R package **sotu**, available on CRAN. The package also contains meta-data concerning each speech that we will add to the document table while annotating the corpus. The code to run this annotation is given by:

```
library(sotu)
library(cleanNLP)

data(sotu_text)
data(sotu_meta)
init_spaCy()
sotu <- cleanNLP::run_annotators(sotu_text, as_strings = TRUE,
                                 meta = sotu_meta)
```

The annotation object, which we will use in the example in the following analysis, is stored in the object `sotu`.

Exploratory analysis

Simple summary statistics are easily computed off of the token table. To see the distribution of sentence length, the token table is grouped by the document and sentence id and the number of rows within each group are computed. The percentiles of these counts give a quick summary of the distribution.

```
library(ggplot2)
library(dplyr)
cleanNLP::get_token(sotu) %>%
  count(id, sid) %$%
  quantile(n, seq(0,1,0.1))
```

```
##   0%  10%  20%  30%  40%  50%  60%  70%  80%
##    1   11   16   19   23   27   31   37   44
##  90% 100%
##   58  681
```

The median sentence has 28 tokens, whereas at least one has over 600 (this is due to a bulleted list in one of the written addresses being treated as a single sentence) To see the most frequently used nouns in the dataset, the token table is filtered on the universal part of speech field, grouped by lemma, and the number of rows in each group are once again calculated. Sorting the output and selecting the top 42 nouns, yields a high level summary of the topics of interest within this corpus.

```
cleanNLP::get_token(sotu) %>%
  filter(upos == "NOUN") %>%
  count(lemma) %>%
  top_n(n = 42, n) %>%
  arrange(desc(n)) %>%
  use_series(lemma)
```

```
##  [1] "year"          "country"
##  [3] "people"        "government"
##  [5] "law"           "time"
##  [7] "nation"        "who"
##  [9] "power"         "interest"
## [11] "world"         "war"
## [13] "citizen"       "service"
## [15] "duty"          "part"
## [17] "system"        "peace"
## [19] "right"         "man"
## [21] "program"       "policy"
## [23] "work"          "act"
## [25] "state"         "condition"
## [27] "subject"       "legislation"
## [29] "force"         "effort"
## [31] "treaty"        "purpose"
## [33] "what"          "land"
## [35] "business"      "action"
## [37] "measure"       "tax"
## [39] "way"           "question"
## [41] "relation"      "consideration"
```

The result is generally as would be expected from a corpus of government speeches, with references to proper nouns representing various organizations within the government and non-proper nouns indicating general topics of interest such as `tax''`,law“, and “peace”.

The length in tokens of each address is calculated similarly by grouping and summarizing at the document id level. The results can be joined with the document table to get the year of the speech and then piped in a **ggplot2** command to illustrate how the length of the State of the Union has changed over time.

```
cleanNLP::get_token(sotu) %>%
count(id) %>%
left_join(cleanNLP::get_document(sotu)) %>%
ggplot(aes(year, n)) +
  geom_line(color = grey(0.8)) +
  geom_point(aes(color = sotu_type)) +
  geom_smooth()
```

Here, color is used to represent whether the address was given as an oral address or a written document. The output shows that their are certainly time trends to the address length, with the form of the address (written versus spoken) also having a large effect on document length.

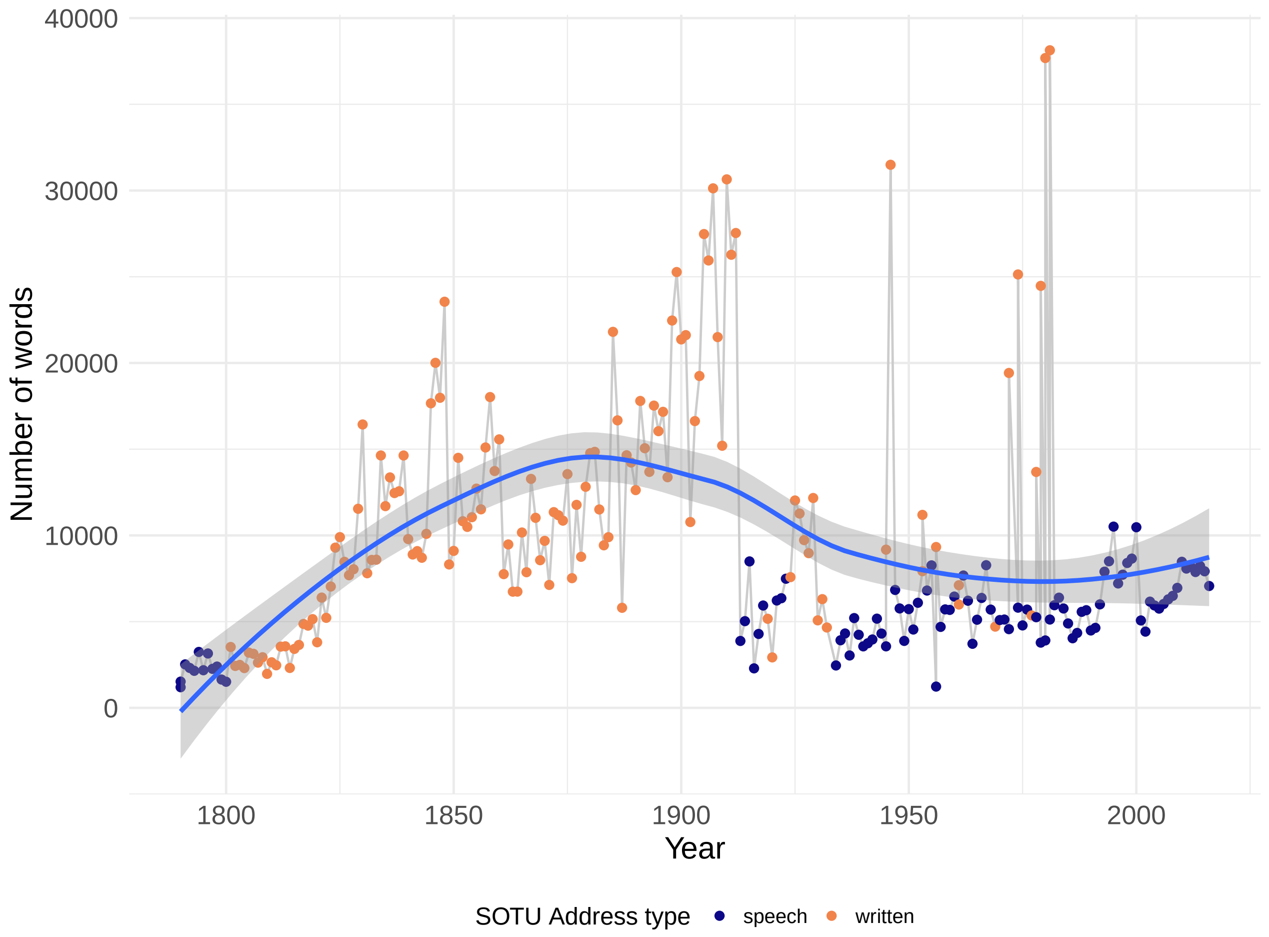

Figure 1: Length of each State of the Union address, in total number of tokens. Color shows whether the address was given as a speech or delivered as a written document.

Finding the most used entities from the entity table over the time period of the corpus yields an alternative way to see the underlying topics. A slightly modified version of the code snippet used to find the top nouns in the dataset can be used to find the top entities. The get_token function is replaced by get_entity and the table is filtered on entity_type rather than the universal part of speech code.

```
cleanNLP::get_entity(sotu) %>%
  filter(entity_type == "GPE") %>%
  count(entity) %>%
  top_n(n = 26, n) %>%
  arrange(desc(n)) %>%
  use_series(entity)
```

```
##  [1] "the United States"
##  [2] "America"
##  [3] "States"
##  [4] "Mexico"
##  [5] "Great Britain"
##  [6] "Spain"
##  [7] "Washington"
##  [8] "China"
##  [9] "Executive"
## [10] "France"
## [11] "Cuba"
## [12] "Japan"
## [13] "Texas"
## [14] "Russia"
## [15] "The United States"
## [16] "Germany"
## [17] "United States"
## [18] "California"
## [19] "Nicaragua"
## [20] "the Soviet Union"
## [21] "Mississippi"
## [22] "Iraq"
## [23] "Alaska"
## [24] "U.S."
## [25] "Philippines"
## [26] "Panama"
## [27] "the District of Columbia"
```

The ability to redo analyses from a slightly different perspective is a direct consequence of the tidy data model supplied by **cleanNLP**. The top locations include some obvious and some less obvious instances. Those sovereign nations included such as Great Britain, Mexico, Germany, and Japan seem as expected given either the United State's close ties or periods of war with them. The top states include the most populous regions (New York, California, and Texas) but also smaller states (Kansas, Oregon, Mississippi), the latter being more surprising.

One of the most straightforward way of extracting a high-level summary of the content of a speech is to extract all direct object object dependencies where the target noun is not a very common word. In order to do this for a particular speech, the dependency table is joined to the document table, a particular document is selected, and relationships of type 'dobj'' (direct object) are filtered out. The result is then joined to the data setword_frequency', which is included with **cleanNLP**, and pairs with a target occurring less than 0.5% of the time are selected to give the final result. Here is an example of this using the first address made by George W. Bush in 2001:

```
cleanNLP::get_dependency(sotu, get_token = TRUE) %>%
  left_join(get_document(sotu)) %>%
  filter(year == 2001, relation == "dobj") %>%
  select(id = id, start = word, word = lemma_target) %>%
  left_join(word_frequency) %>%
  filter(frequency < 0.001) %>%
  select(id, start, word) %$%
  sprintf("%s => %s", start, word)
```

```
##  [1] "take => oath"
##  [2] "using => statistic"
##  [3] "increasing => layoff"
##  [4] "protects => trillion"
##  [5] "makes => welcoming"
##  [6] "accelerating => cleanup"
##  [7] "fight => homelessness"
##  [8] "helping => neighbor"
##  [9] "allowing => taxpayer"
## [10] "provide => mentor"
## [11] "fight => illiteracy"
## [12] "promotes => compassion"
## [13] "asked => ashcroft"
## [14] "end => profiling"
## [15] "pay => trillion"
## [16] "throw => dart"
## [17] "restores => fairness"
## [18] "promoting => internationalism"
## [19] "makes => downpayment"
## [20] "discard => relic"
## [21] "confronting => shortage"
## [22] "directed => cheney"
## [23] "sound => footing"
## [24] "divided => conscience"
## [25] "done => servant"
```

Most of these phrases correspond with the "compassionate conservatism" that George W. Bush ran under in the preceding 2000 election. Applying the same analysis to the 2002 State of the Union, which came under the shadow of the September 11th terrorist attacks, shows a drastic shift in focus.

```
cleanNLP::get_dependency(sotu, get_token = TRUE) %>%
  left_join(get_document(sotu)) %>%
  filter(year == 2002, relation == "dobj") %>%
  select(id = id, start = word, word = lemma_target) %>%
  left_join(word_frequency) %>%
  filter(frequency < 0.0005) %>%
  select(id, start, word) %$%
  sprintf("%s => %s", start, word)
```

```
##  [1] "urged => follower"
##  [2] "called => troop"
##  [3] "brought => sorrow"
##  [4] "owe => micheal"
##  [5] "ticking => timebomb"
##  [6] "have => troop"
##  [7] "hold => hostage"
##  [8] "eliminate => parasite"
##  [9] "flaunt => hostility"
## [10] "develop => anthrax"
## [11] "put => troop"
## [12] "increased => vigilance"
## [13] "fight => anthrax"
## [14] "thank => attendant"
## [15] "defeat => recession"
## [16] "want => paycheck"
## [17] "set => posturing"
## [18] "enact => safeguard"
## [19] "embracing => ethic"
## [20] "owns => aspiration"
## [21] "containing => resentment"
## [22] "erasing => rivalry"
## [23] "embrace => tyranny"
```

Here the topics have almost entirely shifted to counter-terrorism and national security efforts.

Models

The `get_tfidf` function provided by **cleanNLP** converts a token table into a sparse matrix representing the term-frequency inverse document frequency matrix (or any intermediate part of that calculation). This is particularly useful when building models from a textual corpus. The `tidy_pca`, also included with the package, takes a matrix and returns a data frame containing the desired number of principal components. Dimension reduction involves piping the token table for a corpus into the `get_tfidf` function and passing the results to `tidy_pca`.

```
pca <- cleanNLP::get_token(sotu) %>%
  filter(pos %in% c("NN", "NNS")) %>%
  cleanNLP::get_tfidf(min_df = 0.05, max_df = 0.95,
                      type = "tfidf", tf_weight = "dnorm") %$%
  cleanNLP::tidy_pca(tfidf, get_document(sotu))
```

In this example only non-proper nouns have been included in order to minimize the stylistic attributes of the speeches in order to focus more on their content. A scatter plot of the speeches using these components is shown. There is a definitive temporal pattern to the documents, with the 20th century addresses forming a distinct cluster on the right side of the plot.

The output of the `get_tfidf` function may be given directly to the `LDA` function in the package **topicmodels**. The topic model function requires raw counts, so the type variable in `get_tfidf` is set to ‘`tf''; the results may then be directly piped toLDA`’.

```
library(topicmodels)
tm <- cleanNLP::get_token(sotu) %>%
  filter(pos %in% c("NN", "NNS")) %>%
  cleanNLP::get_tfidf(min_df = 0.05, max_df = 0.95,
                      type = "tf", tf_weight = "raw") %$%
  LDA(tf, k = 16, control = list(verbose = 1))
```

The topics, ordered by approximate time period, are visualized in the Figure. Most topics persist for a few decades and then largely disappear, though some persist over non-contiguous periods of the presidency. The Energy topic, for example, appears during the 1950s and crops up again during the energy crisis of the 1970s. The “world, man, freedom, force, defense” topic peaks during both World Wars, but is absent during the 1920s and early 1930s.

Finally, the **cleanNLP** data model is also convenient for building predictive models. The State of the Union corpus does not lend itself to an obviously applicable prediction problem. A classifier that distinguishes speeches made by George W. Bush and Barrack Obama will be constructed here for the purpose of illustration. As a first step, a term-frequency matrix is extracted using the same technique as was used with the topic modeling function. However, here the frequency

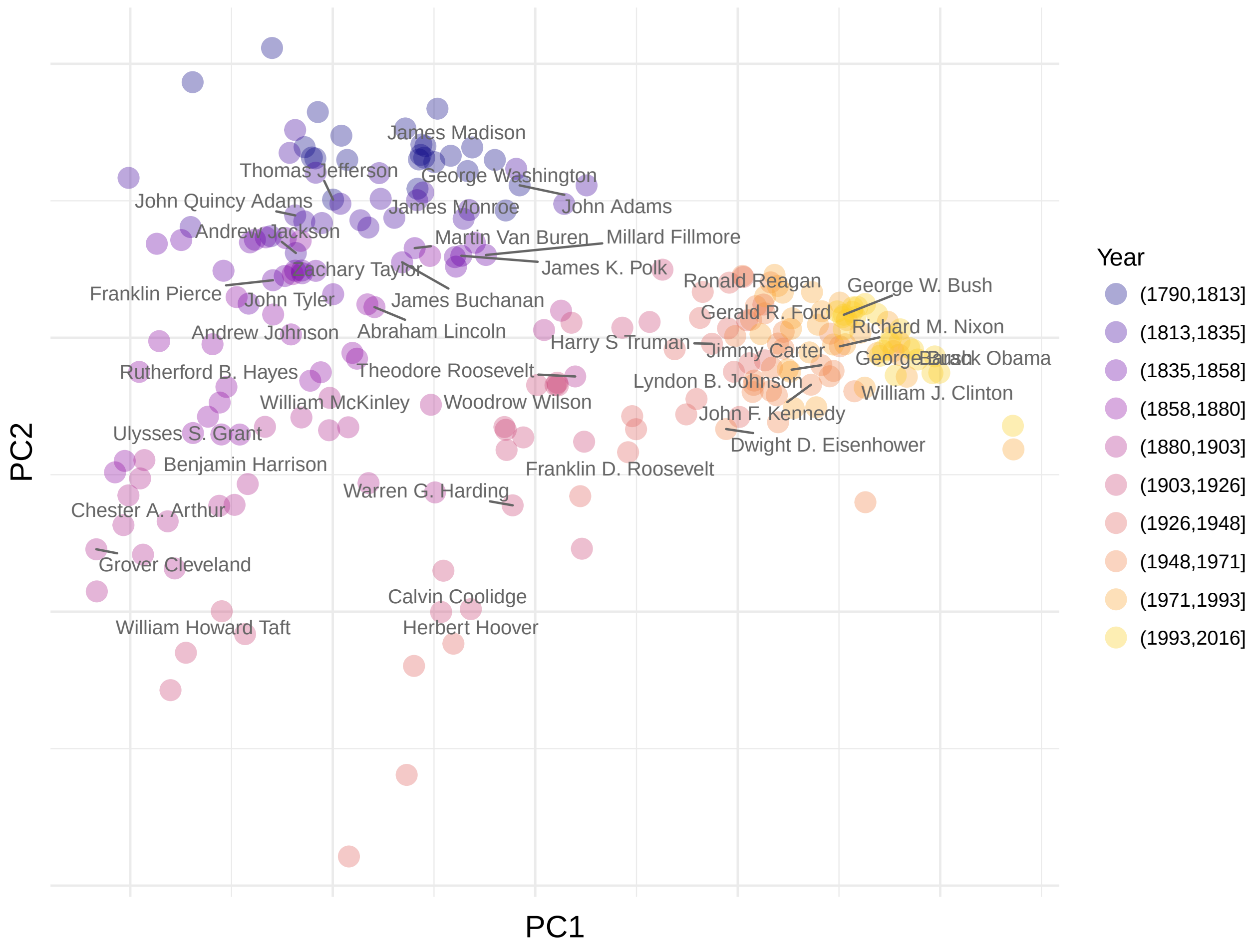

Figure 2: State of the Union Speeches, highlighting each President's first address, plotted using the first two principal components of the term frequency matrix of non-proper nouns.

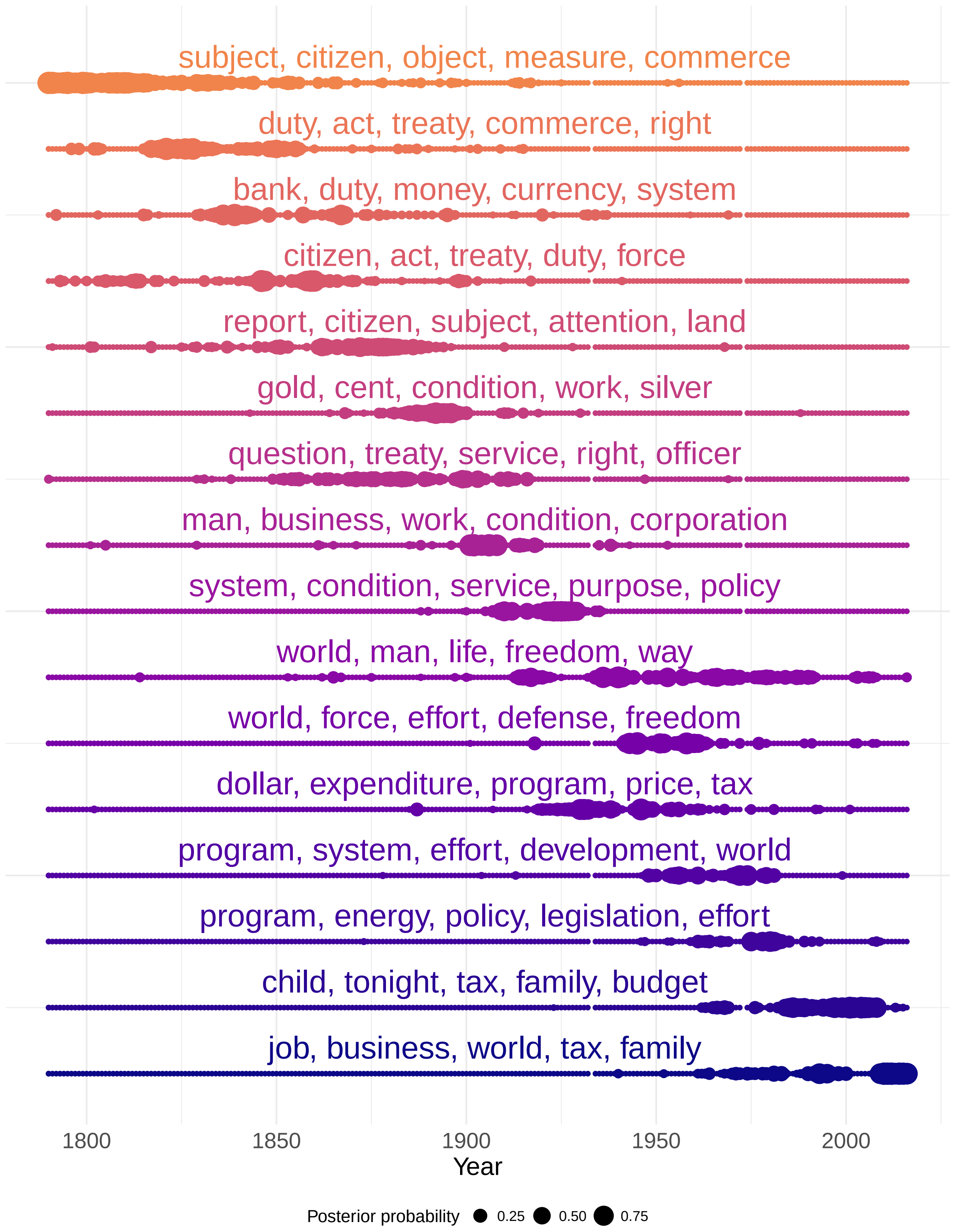

Figure 3: Distribution of topic model posterior probabilities over time on the State of the Union corpus. Top five words associated with each topic are displayed, with topics sorted by the median year of all documents placed into the respective topic using the maximum posterior probabilities.

is computed for each sentence in the corpus rather than the document as a whole. The ability to do this seamlessly with a single additional `mutate` function defining a new id illustrates the flexibility of the `get_tfidf` function.

```
df <- get_token(sotu) %>%
  left_join(get_document(sotu)) %>%
  filter(year > 2000) %>%
  mutate(new_id = paste(id, sid, sep = "-")) %>%
  filter(pos %in% c("NN", "NNS"))
mat <- get_tfidf(df, min_df = 0, max_df = 1, type = "tf",
                 tf_weight = "raw", doc_var = "new_id")
```

It will be nessisary to define a response variable y indicating whether this is a speech made by President Obama as well as a training flag indicating which speeches were made in odd numbered years. This is done via a separate table join and a pair of mutations.

```
meta <- data_frame(new_id = mat$id) %>%
  left_join(df[!duplicated(df$new_id),]) %>%
  mutate(y = as.numeric(president == "Barack Obama")) %>%
  mutate(train = year %in% seq(2001,2016, by = 2))
```

The output may now be used as input to the elastic net function provided by the **glmnet** package. The response is set to the binomial family given the binary nature of the response and training is done on only those speeches occurring in odd-numbered years. Cross-validation is used in order to select the best value of the model's tuning parameter.

```
library(glmnet)
model <- cv.glmnet(mat$tf[meta$train,], meta$y[meta$train], family = "binomial")
```

The algorithm does a very good job of separating the speeches. Looking at the odd years versus even years (the training and testing sets, respectively) indicates that the model has not been over-fit.

One benefit of the penalized linear regression model is that it is possible to interpret the coefficients in a meaningful way. Here are the non-zero elements of the regression vector, coded as whether the have a positive (more Obama) or negative (more Bush) sign:

```
beta <- coef(model, s = model[["lambda"]][11])[-1]
sprintf("%s (%d)", mat$vocab, sign(beta))[beta != 0]
```

These generally seem as expected given the main policy topics of focus under each administration. During most of the Bush presidency, as mentioned before, the focus was on national security and foreign policy. Obama, on the other hand, inherited the recession of 2008 and was far more focused on the overall economic policy.

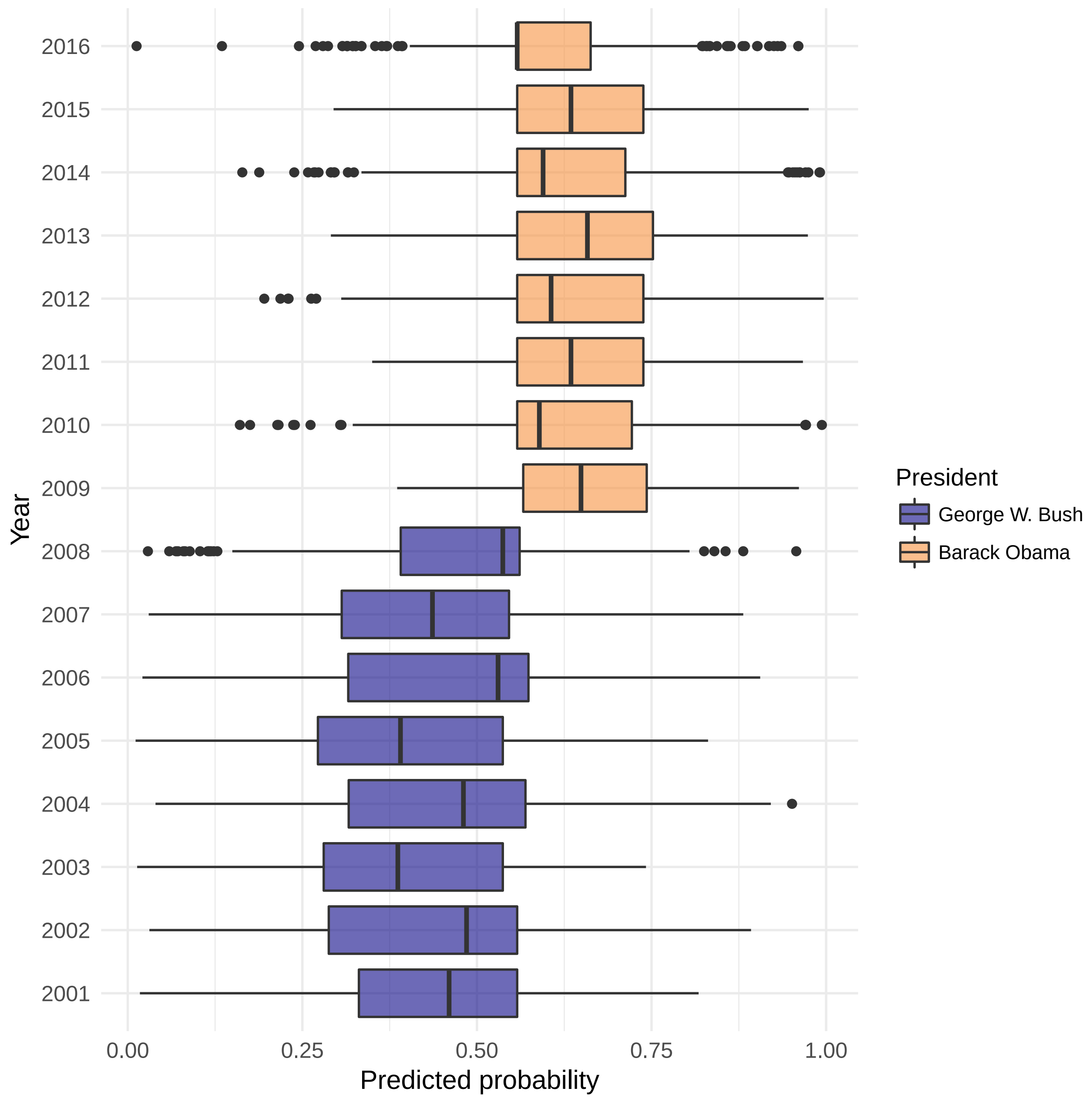

Figure 4: Boxplot of predicted probabilities, at the sentence level, for all 16 State of the Union addresses by Presidents George W. Bush and Barack Obama. The probability represents the extent to which the model believe the sentence was spoken by President Obama. Odd years were used for training and even years for testing. Cross-validation on the training set was used, with the one standard error rule, to set the lambda tuning parameter.