Text Analytics

Devon Stone

13/09/2018

Section 3: Text Analysis

This markdown file displays the text analysis techniques used on the provided set of presidential speeches. The markdown first steps through the pre-processing steps completed and then highlights the various techniques used.

Libraries Used

```
library(quanteda)
```

```
## Package version: 1.3.4
```

```
## Parallel computing: 2 of 4 threads used.
```

```
## See https://quanteda.io for tutorials and examples.
```

```
##
## Attaching package: 'quanteda'
```

```
## The following object is masked from 'package:utils':
##
##     View
```

```
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'tm'
```

```
## The following objects are masked from 'package:quanteda':
##
##     as.DocumentTermMatrix, stopwords
```

```
library(readtext)
library(topicmodels)
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

Read in Speeches

The first step in the process is to bring in the dataset. The presidential speeches analysed were provided as text files. The code below reads all text files in the folder named 'data' and stores them in a dataframe.

```
# load data in the format of president followed by text
data <- readtext('../data/speeches')
head(data)
```

```
## readtext object consisting of 6 documents and 0 docvars.
## # data.frame [6 × 2]
##   doc_id           text
## * <chr>            <chr>
## 1 1994_deKlerk.txt "\"28 Februar\"..."
## 2 1994_Mandela.txt "\"24 May 199\"..."
## 3 1995_Mandela.txt "\"17 Februar\"..."
## 4 1996_Mandela.txt "\"9 February\"..."
## 5 1997_Mandela.txt "\"7 February\"..."
## 6 1998_Mandela.txt "\"6 February\"..."
```

After reading in the data the next process was completed was to extract the year and president of each speed from the file name of each speech

```
# create lists containing years and presidents
years <- sapply(strsplit(data[,1], "_"), "[[", 1)
pres <- sapply(strsplit(data[,1], "_"), "[[", 2)
pres <- sub(".txt", "", pres)


# create columns in the dataframe containing the years and presidents
data$year <- years
data$president <- pres

# check data
head(data)
```

```
## readtext object consisting of 6 documents and 2 docvars.
## # data.frame [6 × 4]
##   doc_id           text                year  president
## * <chr>            <chr>               <chr> <chr>
## 1 1994_deKlerk.txt "\"28 Februar\"..." 1994  deKlerk
## 2 1994_Mandela.txt "\"24 May 199\"..." 1994  Mandela
## 3 1995_Mandela.txt "\"17 Februar\"..." 1995  Mandela
## 4 1996_Mandela.txt "\"9 February\"..." 1996  Mandela
## 5 1997_Mandela.txt "\"7 February\"..." 1997  Mandela
## 6 1998_Mandela.txt "\"6 February\"..." 1998  Mandela
```

Data Preparation

Before any analysis can be completed some necessary data processing steps need to be followed.

Tokenization

We break the documents down into individual word tokens. Text is broken into word tokens as words are often the most semantically meaningful components of text and therefore we are able to perform the most meaningful analysis on the words themselves, as opposed to the entire sentences.

```
# tokenization
# we remove punctuation from the documents in this step
toks <- tokens(as.character(data['text']), remove_punct = TRUE)

# we can examples of the tokenized words from the 1st document
options(max.print=10)
toks[1]
```

```
## tokens from 1 document.
## text1 :
##  [1] "28"         "February"   "1994"       "Mr"         "Speaker"
##  [6] "This"       "Parliament" "has"        "convened"   "to"
##  [ reached getOption("max.print") -- omitted 2020 entries ]
```

Normalization

We now normalize the words by ensuring all tokens are in lower case and stemming the words. Stemming involves removing the suffixes and plural attachments of words, which can be considered as variations of root words. Stemming ensures that only the root word of each token is considered the analysis. For example, if we stem the word running it would be seen as run.

```
toks <- tokens_tolower(toks)
toks <- tokens_wordstem(toks)
```

Stopwords

We now remove stopwords such as “an, a and the” from each tokenized document so that only informative words remain.

```
# remove stopwords
sw <- stopwords("english")
toks <- tokens_remove(toks, sw)

# we can examples of the tokenized words from the 1st document
options(max.print=10)
toks[1]
```

```
## tokens from 1 document.
## text1 :
##  [1] "28"         "februari"   "1994"       "mr"         "speaker"
##  [6] "parliament" "conven"     "adopt"      "import"     "amend"
##  [ reached getOption("max.print") -- omitted 1042 entries ]
```

Document-term matrix

We now place the manipulated tokens into a bag of a bag-of-words format called a document-term matrix (DTM). The DTM is in the following format: rows are sentences, columns are words, and cells indicate how often each term occurred in each sentence.

```
# convert speeches to a corpus of words
corp <- corpus(data)

# we can lower, stem, tokenize and remove punction while creating the dtm
dtm <- dfm(corp, tolower=TRUE, stem=TRUE, remove_punct=TRUE, remove=stopwords("engl
ish"))
dtm
```

```
## Document-feature matrix of: 30 documents, 7,510 features (83.4% sparse).
```

Filtering and weighting

Beyond stopwords, there are still words in the DTM that are not meaningful. These meaningless words are often very common to many sentences and can be removed using their high frequencies. Very rare words can make analysis more difficult and removing them can lead to better accuracies in category prediction (in the case of this assignment, president prediction).

We can further increase category prediction accuracy by assigning weights to each tokenized word in the DTM. The weights assign an information value to each word. A popular weighting method is term frequency-inverse document frequency (tf-idf).

```
# create the freq of each word
word_freq <- docfreq(dtm)

# only take words that appear more than twice
dtm <- dtm[, word_freq > 2]
dtm
```

```
## Document-feature matrix of: 30 documents, 2,974 features (64.4% sparse).
```

```
# give item values to each word
weighted_dtm <- dfm_tfidf(dtm)
weighted_dtm
```

```
## Document-feature matrix of: 30 documents, 2,974 features (64.4% sparse).
```

We can see the effects of filtering in difference between the sparsity percentages of the DTM created before filtering and the one after filtering.

Analysis

We are going to perform both deductive and inductive techniques on the dataset. The deductive methods involve classifying the words in sentence into a predefined category. While, the inductive techniques involve finding meaningful patterns in the words themselves, rather than categorising the words.

Counting and dictionary

The dictionary approach involves counting the occurrence of predefined concepts in the sentences. This is a very simple approach and requires strong user input. We create a dictionary so that we can gain an insight into the major topics of each speech. The dictionary below is designed to capture… The dictionary should be created using stem words.

```
# change max print again
options(max.print=1000)

# create dictionary
dict <- dictionary(list(economy=c("job*", "econom*", "grow*", "business*", "finan*"
, "inflat*", "imorts*", "export*"), law=c("constitu*", "amend*", "law*"), toursim=c
("touris*", "travel*"), culture=c("communit*", "informal settle*", "culture*"), gov
ernment=c("government*", "minist*", "provinc*")))

# create the lookup dtm
dict_dtm <- dfm_lookup(dtm, dict, nomatch = "unmatched")
print(dict_dtm, ndoc=30)
```

```
## Document-feature matrix of: 30 documents, 6 features (2.78% sparse).
## 30 x 6 sparse Matrix of class "dfm"
##                     features
## docs                 economy law toursim culture government unmatched
##   1994_deKlerk.txt         7  59       0       1         19       855
##   1994_Mandela.txt        42   5       3       8          5      2137
##   1995_Mandela.txt        17  22       0       3         18      2953
##   1996_Mandela.txt        45  23       1      24         15      3023
##   1997_Mandela.txt        26  14       1      15         12      2900
##   1998_Mandela.txt        40  11       1      14         15      3383
##   1999-2_Mandela.txt      32   8       3      11          6      3145
##   1999-6_Mandela.txt      35  11       1       9         14      2773
##   2000_Mbeki.txt          68  11       3       7         12      2892
##   2001_Mbeki.txt          55   6       6       4         11      3376
##   2002_Mbeki.txt          57   7       2      18         11      4013
##   2003_Mbeki.txt          71   9       3       9         21      4769
##   2004-2_Mbeki.txt        44   9       0       3          9      3095
##   2004-5_Mbeki.txt        59   8       0      10         15      2855
##   2005_Mbeki.txt          62  12       1      19         16      4463
##   2006_Mbeki.txt          62   5       1       6         12      3845
##   2007_Mbeki.txt          31  14       3       5         17      3701
##   2008_Mbeki.txt          31  14       3       5         17      3697
##   2009_Motlanthe.txt      71   9       0       7          8      3692
##   2009_Zuma.txt           39   4       1      12          8      2551
##   2010_Zuma.txt           33   1       2      10          6      2215
##   2011_Zuma.txt           63   3       8      10          7      2766
##   2012_Zuma.txt           59  10       1       8         19      2740
##   2013_Zuma.txt           38  17       2       8         15      2991
##   2014-2_Zuma.txt         45  12       2       9          7      3024
##   2014-6_Zuma.txt         46   3       3       5         12      2551
##   2015_Zuma.txt           62   8       4       7          7      2797
##   2016_Zuma.txt           72   7       3       8         12      2681
##   2017_Zuma.txt           56  14       4       7         15      2983
##   2018_Ramaphosa.txt      80   6       4       8         10      2716
```

We can see from the above counts the major topics covered in each speech. President de Klerk's speech in 1994 (pre the first fully free elections) contained the most words based around law and the

constitution. While President Ramaphosa's 2018 contained the most words relating to the country's economy.

Topic Modeling

We are going to build on the above section by removing the human input from the "topic" generation and turn our focus away from the topics in each speach but rather focus on words that lie in similar topics. We are going to do this through Topic Modelling, which is in an unsupervised learning technique where the goal is to group words in different documents into like "topics". The topic modelling technique we are going to use is Latent Dirichlet Allocation (LDA). LDA is famous for producing human-readable topic groups. The input for an LDA model is a bag-of-words in matrix form (DTM created above). An LDA model is essentially a two-layer aggregation. One for the distribution of topics across a document and the other for distribution of each word within a particular topic.

LDA invloves giving each tokenized word a weight, as shown in the `weighted_dtm`. The Image 1 shows the exact process of LDA. The alpha symbol represents the Dirichlet prior on the per-document topic distributions, beta is the Dirichlet prior on the per-topic word distribution, theta represents the topic distribution for document n, Z is the is the topic for the n-th word in document m and W is a specific word.

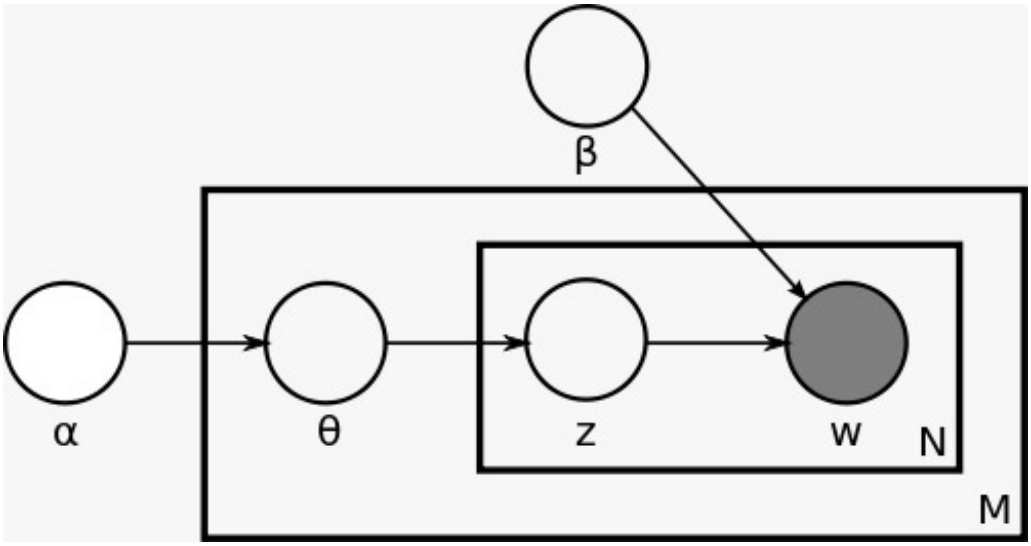


Image 1: Latent Dirichlet Allocation

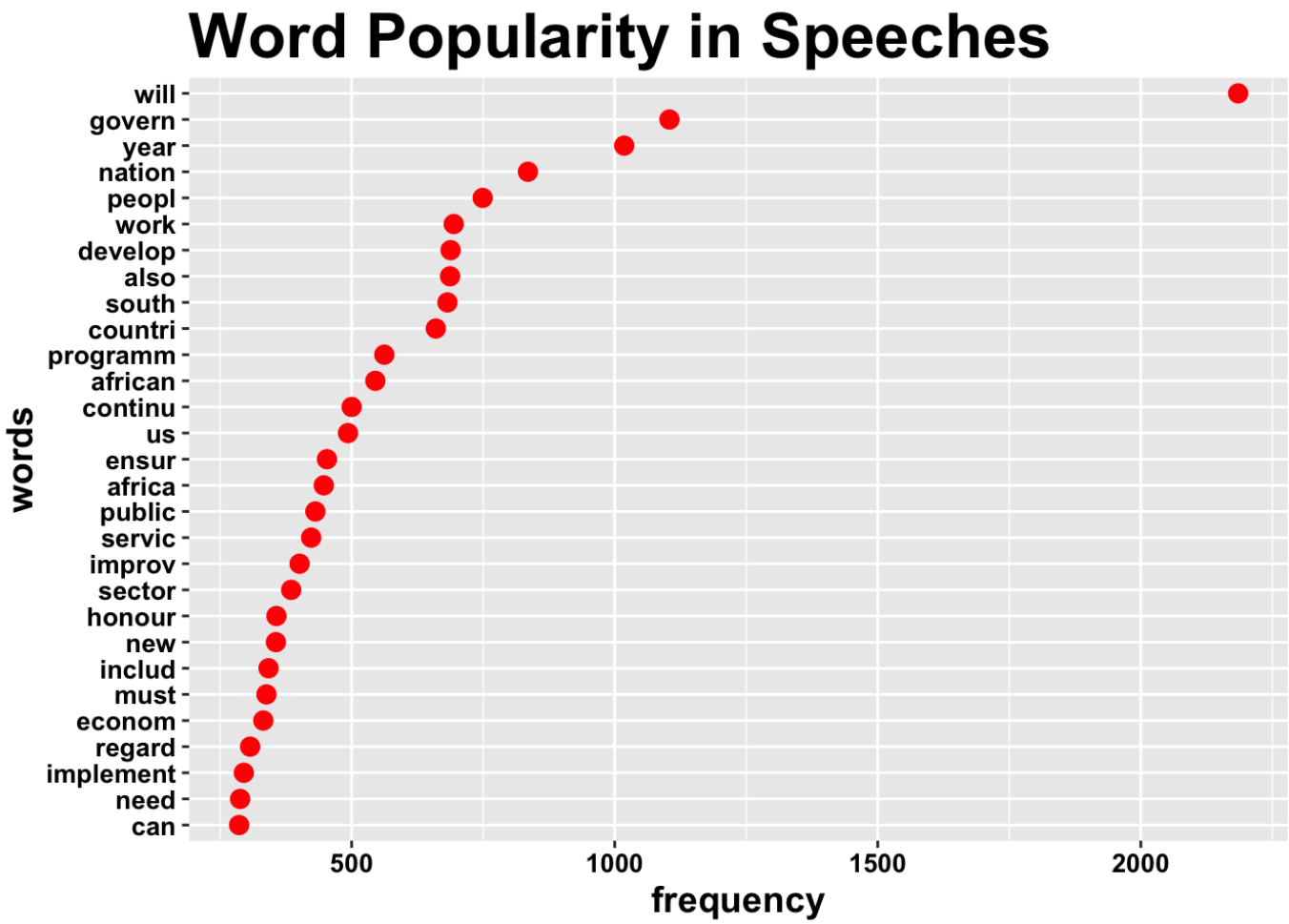
Most Common Words

We start by displaying the 30 most common words across all the speeches in the dataset so we can get a slightly better understanding of the common words across the presidential speeches.

```
# freqency of words in the corpus and sort the list
freq <- sort(colSums(dtm), decreasing=TRUE)

# make the freq table a dtaframe
freq <- data.frame(words = names(freq), frequency=freq)
freq$words = reorder(freq$words, freq$frequency)
head(freq)
```

```
##         words frequency
## will     will      2185
## govern govern      1104
## year     year      1018
## nation nation       835
## peopl   peopl       749
## work     work       694
```

```
# plot the words and their frequency
ggplot(freq[1:29,], aes(x=words, y=frequency)) + geom_point(size=3, colour="red") +
coord_flip() +ggtitle("Word Popularity in Speeches") +
theme(axis.text.x=element_text(size=10,face="bold", colour="black"), axis.text.y=el
ement_text(size=10,colour="black",
face="bold"), axis.title.x=element_text(size=14, face="bold"), axis.title.y=element
_text(size=14,face="bold"),
plot.title=element_text(size=24,face="bold"))
```

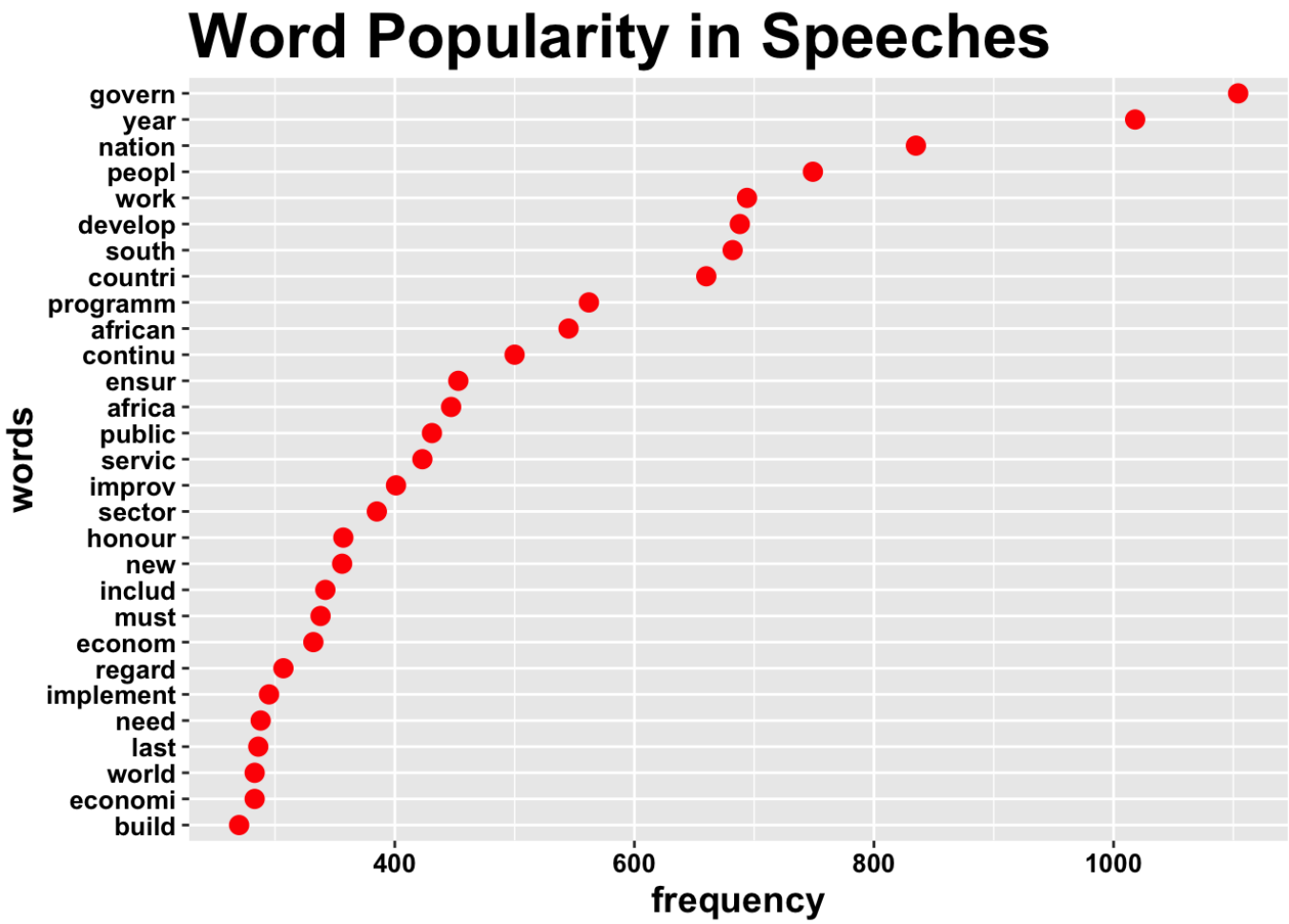


We can see from the above plot that the word appears significantly more popular than any of the other words. We will remove this word in case it skews the model. We will also remove other words like also, us and can that could be considered as stopwords.

```
# remove will
n_corp <- tm_map(Corpus(VectorSource(data)), removeWords, c("will", "can", "us", "a
lso"))
n_corp <- corpus(n_corp)
n_dtm <- dfm(n_corp, tolower=TRUE, stem=TRUE, remove_punct=TRUE, remove=stopwords("
english"))

# recalculate freq
n_freq <- sort(colSums(n_dtm), decreasing=TRUE)

# make the freq table a dtaframe
n_freq <- data.frame(words = names(n_freq), frequency=n_freq)
n_freq$words = reorder(n_freq$words, n_freq$frequency)
head(n_freq)
```

```
##           words frequency
## govern   govern      1104
## year       year      1018
## nation   nation       835
## peopl     peopl       749
## work       work       694
## develop develop       688
```

```
# make the plot
ggplot(n_freq[1:29,], aes(x=words, y=frequency)) + geom_point(size=3, colour="red")
+ coord_flip() +ggtitle("Word Popularity in Speeches") +
theme(axis.text.x=element_text(size=10,face="bold", colour="black"), axis.text.y=el
ement_text(size=10,colour="black",
face="bold"), axis.title.x=element_text(size=14, face="bold"), axis.title.y=element
_text(size=14,face="bold"),
plot.title=element_text(size=24,face="bold"))
```



Creating the Model

Now we create the LDA model.

```
# set a seed
set.seed(42)

# create model
lda_mod <- LDA(n_dtm, method = "Gibbs", k = 10)
```

```
# show model results
terms(lda_mod, 5)
```

```
##      Topic 1   Topic 2    Topic 3      Topic 4  Topic 5   Topic 6
## [1,] "regard"  "shall"    "year"       "shall"  "peopl"   "economi"
## [2,] "municip" "start"    "south"      "energi" "mandela" "black"
## [3,] "object"  "new"      "compatriot" "matter" "countri" "student"
## [4,] "month"   "cours"    "honour"     "effici" "nelson"  "transform"
## [5,] "per"     "question" "work"       "provid" "decad"   "mine"
##      Topic 7   Topic 8   Topic 9   Topic 10
## [1,] "fellow"  "job"     "govern"  "constitut"
## [2,] "forward" "road"    "year"    "import"
## [3,] "declin"  "mr"      "nation"  "elect"
## [4,] "mandela" "rand"    "develop" "govern"
## [5,] "former"  "corrupt" "peopl"   "matter"
```

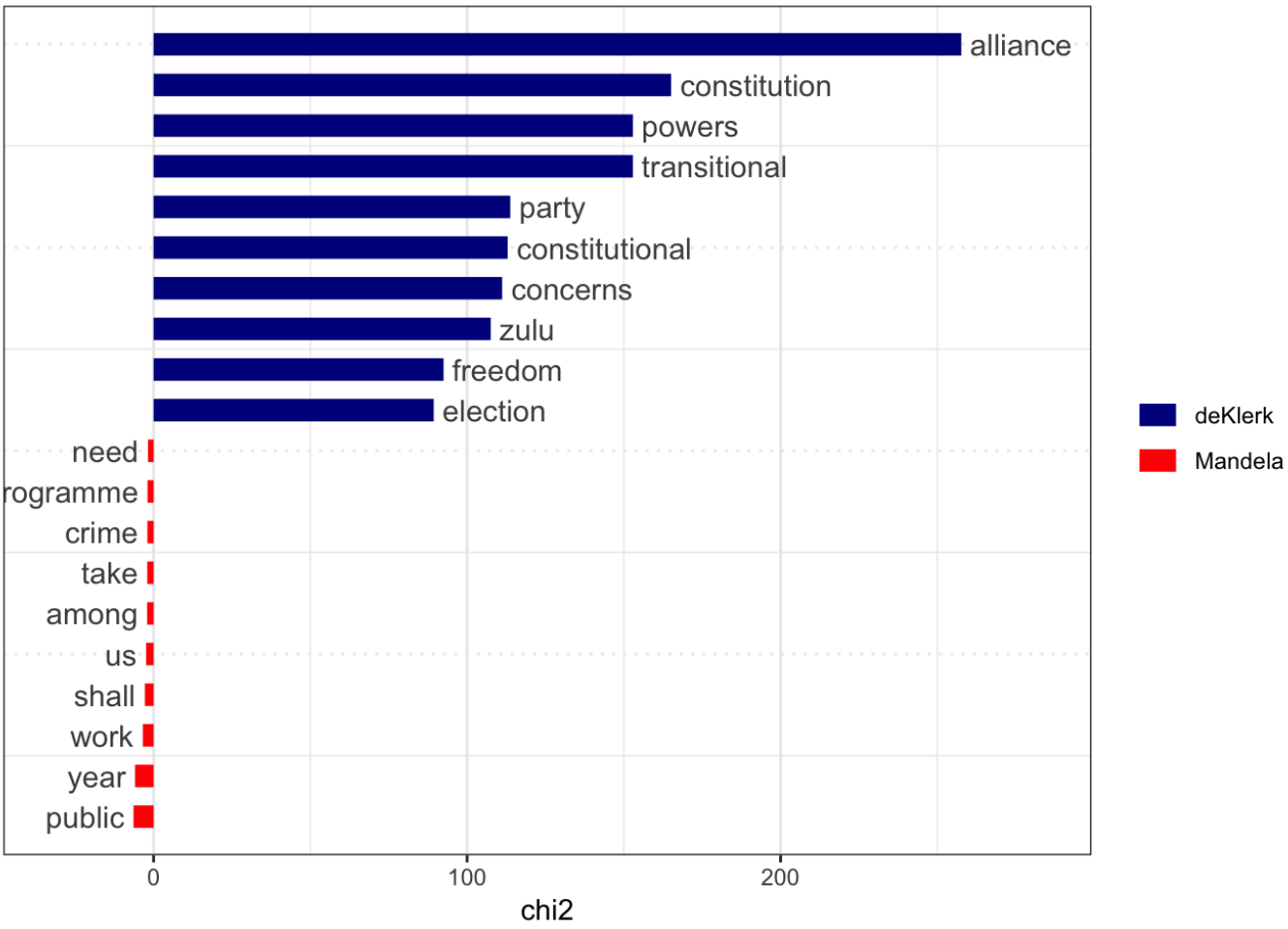
We can see that LDA model did not cluster the words that definitively but we can see certain topics have a trend. Topic 2 appears to be about starting something new. Topic 6 appears to be about

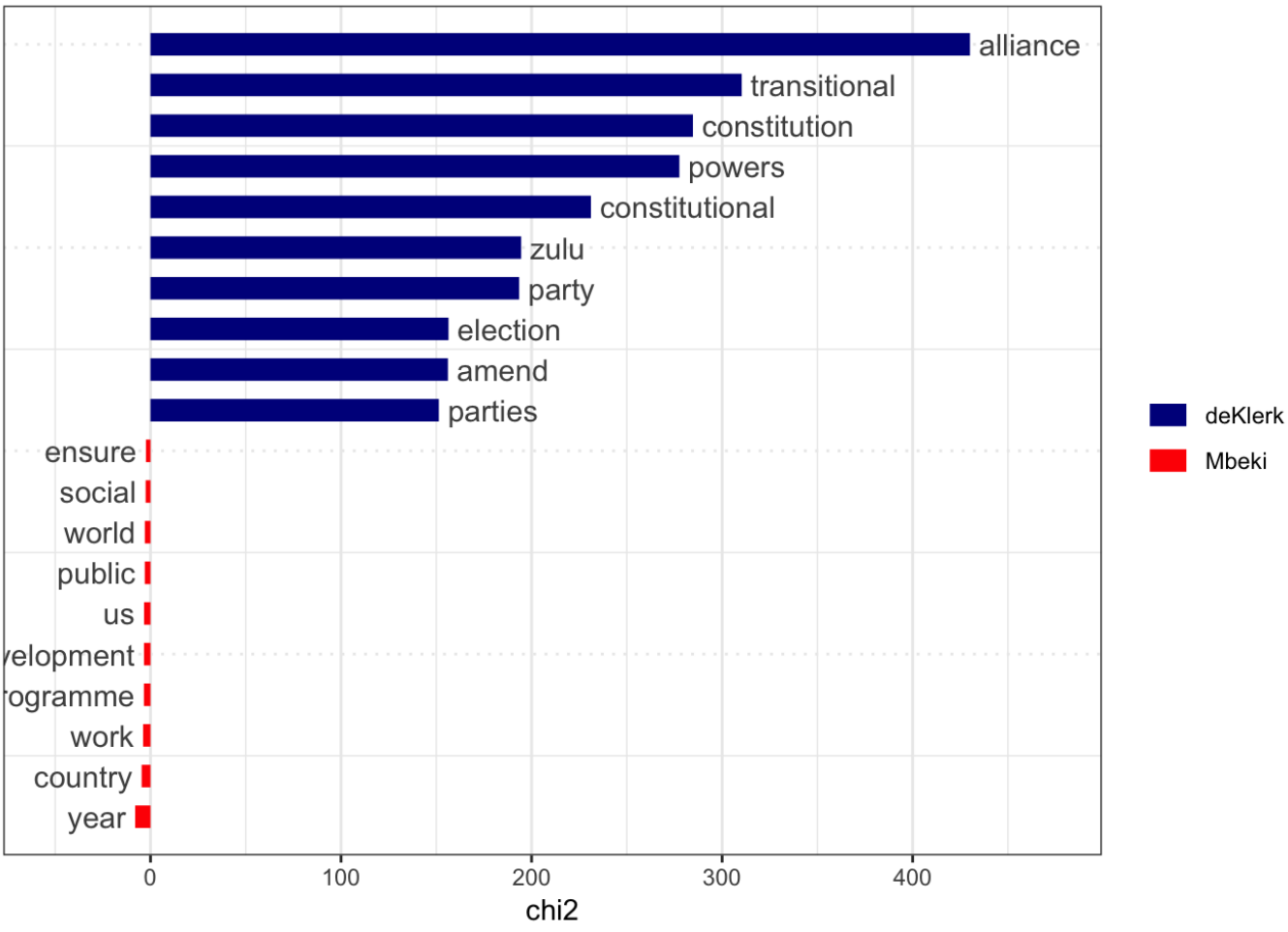
transformation and economics. Topic 10 appears to be about law and governance.

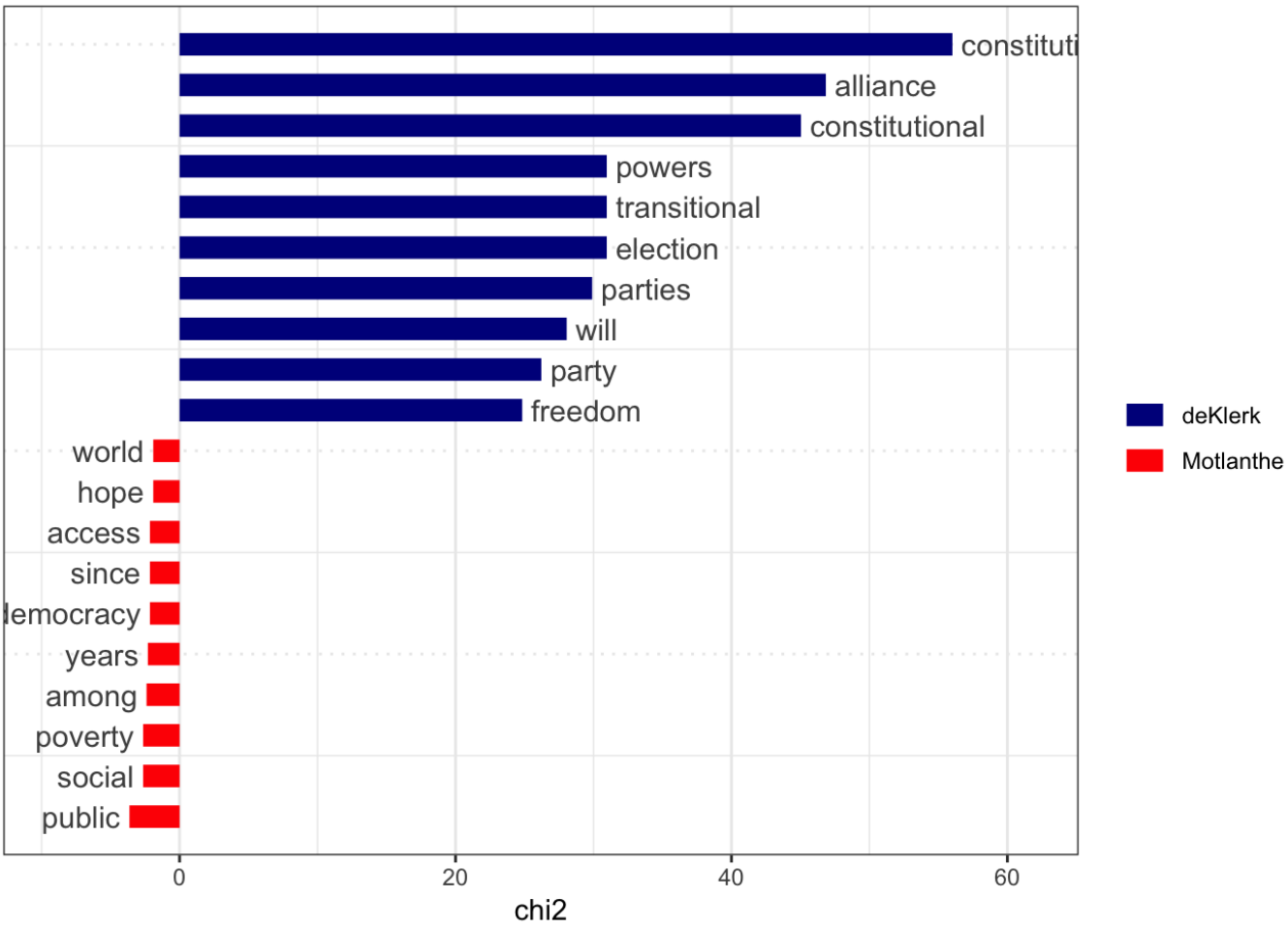
Statistics

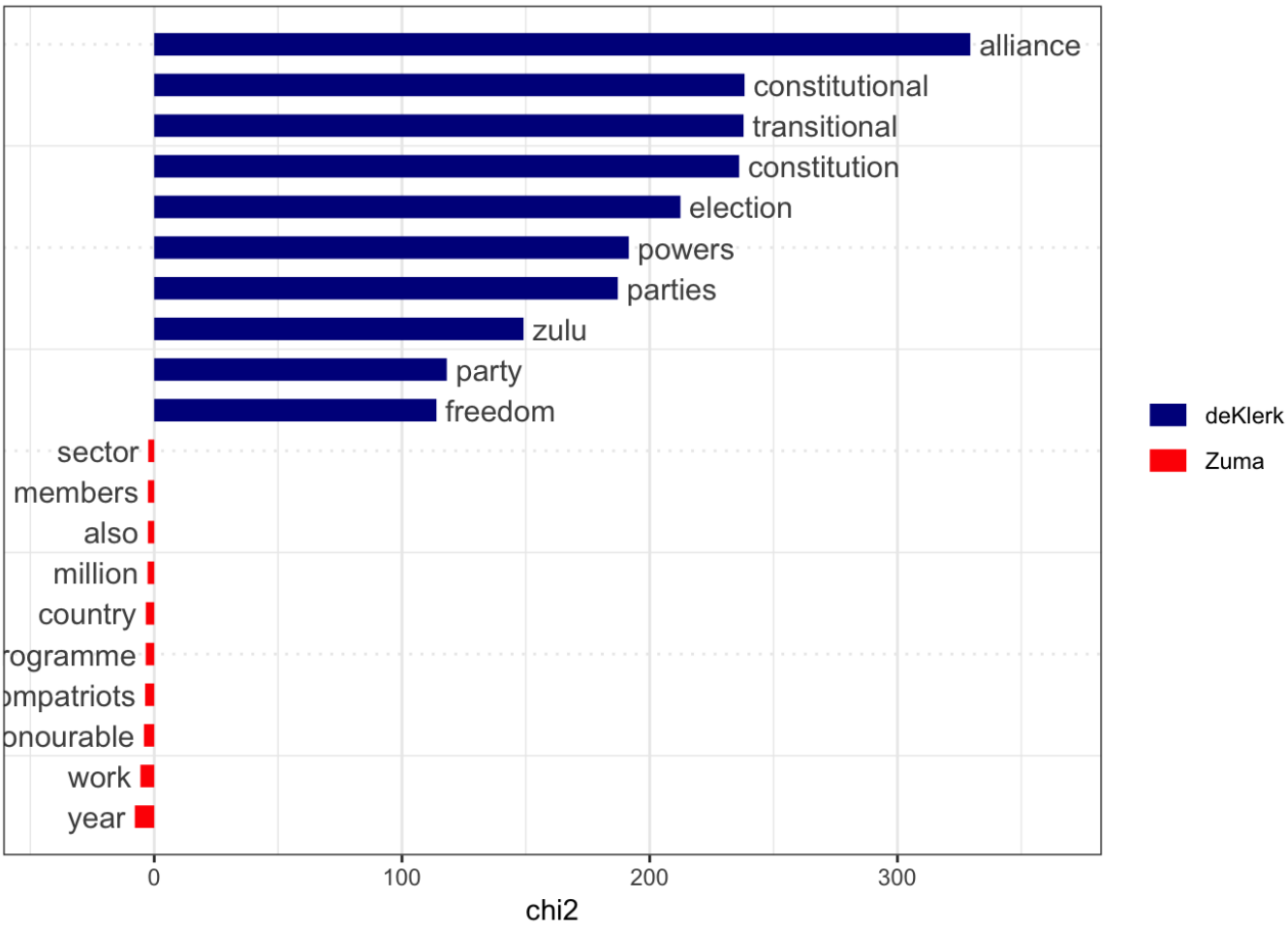
A particularly useful technique is to compare the term frequencies of two corpora, or between two subsets of the same corpus. For instance, we can see which president was more likely to say a certain word in the speech when compared to another. In addition to providing a way to quickly explore how this topic is discussed in the corpus, this can provide input for developing better queries.

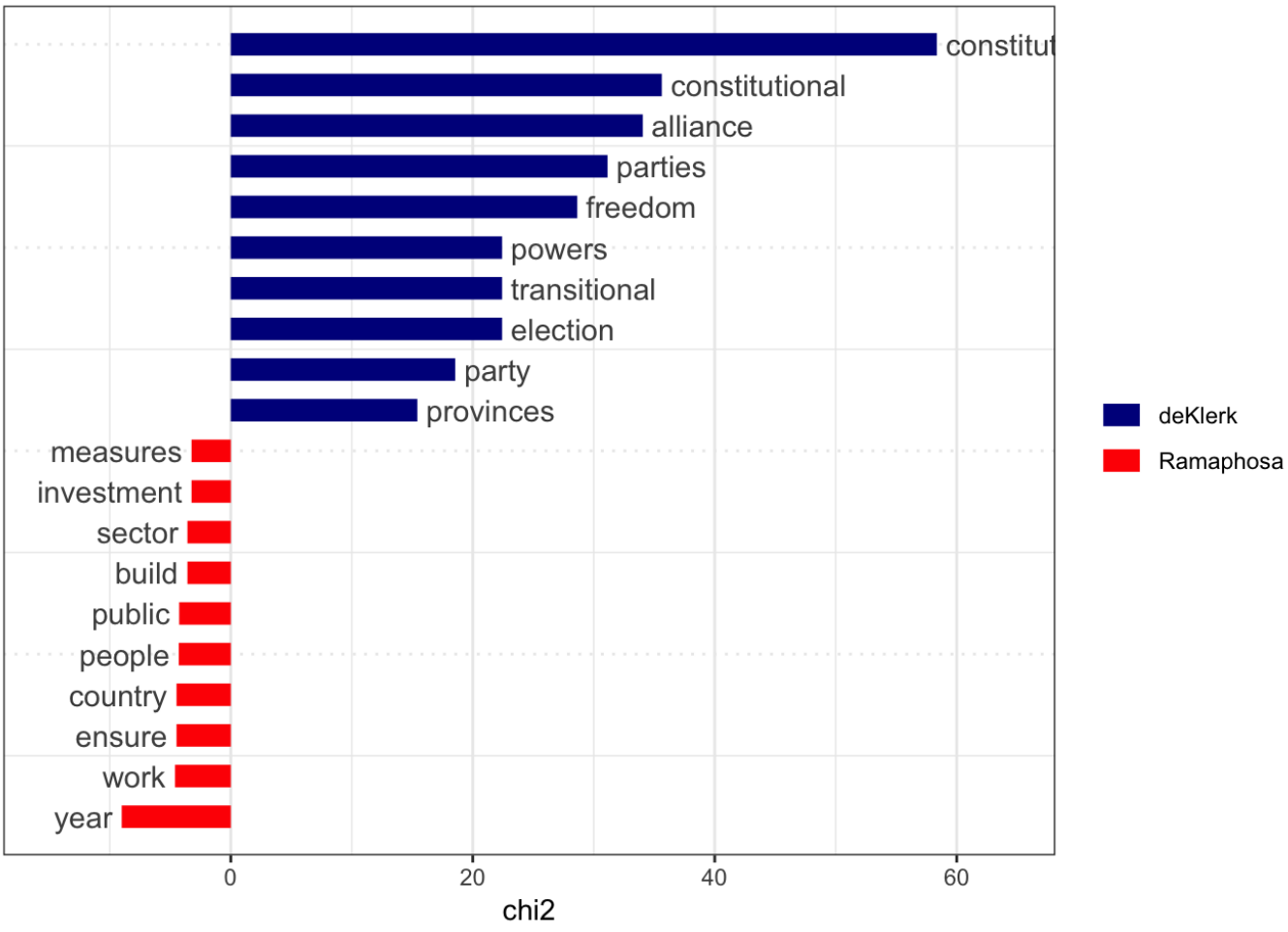
```
# list of unique presidents
presidents <- unique(pres)

# set layout
par(mfrow=c(15,1))

# creare a loop to go through all presidential combinations
for (i in 1:(length(presidents) -1)){

  for (j in (i+1):(length(presidents))){


        # corpus of only two presidents
        corp_zu_man <- corpus_subset(corp, president %in% c(presidents[i], presiden
ts[j]))

        # DTM from the above corpus
        dtm_zu_man <- dfm(corp_zu_man, groups = "president", tolower=TRUE, stem =
FALSE,       remove_punct=TRUE, remove=stopwords("english"))

        # plot the comaprison
        keyness <- textstat_keyness(dtm_zu_man, target = presidents[i])
        plot <- textplot_keyness(keyness, show_legend = TRUE, color = c("darkblue",
"red"), n = 10L,  labelsize = 4)

        print(plot)

  }
}
```

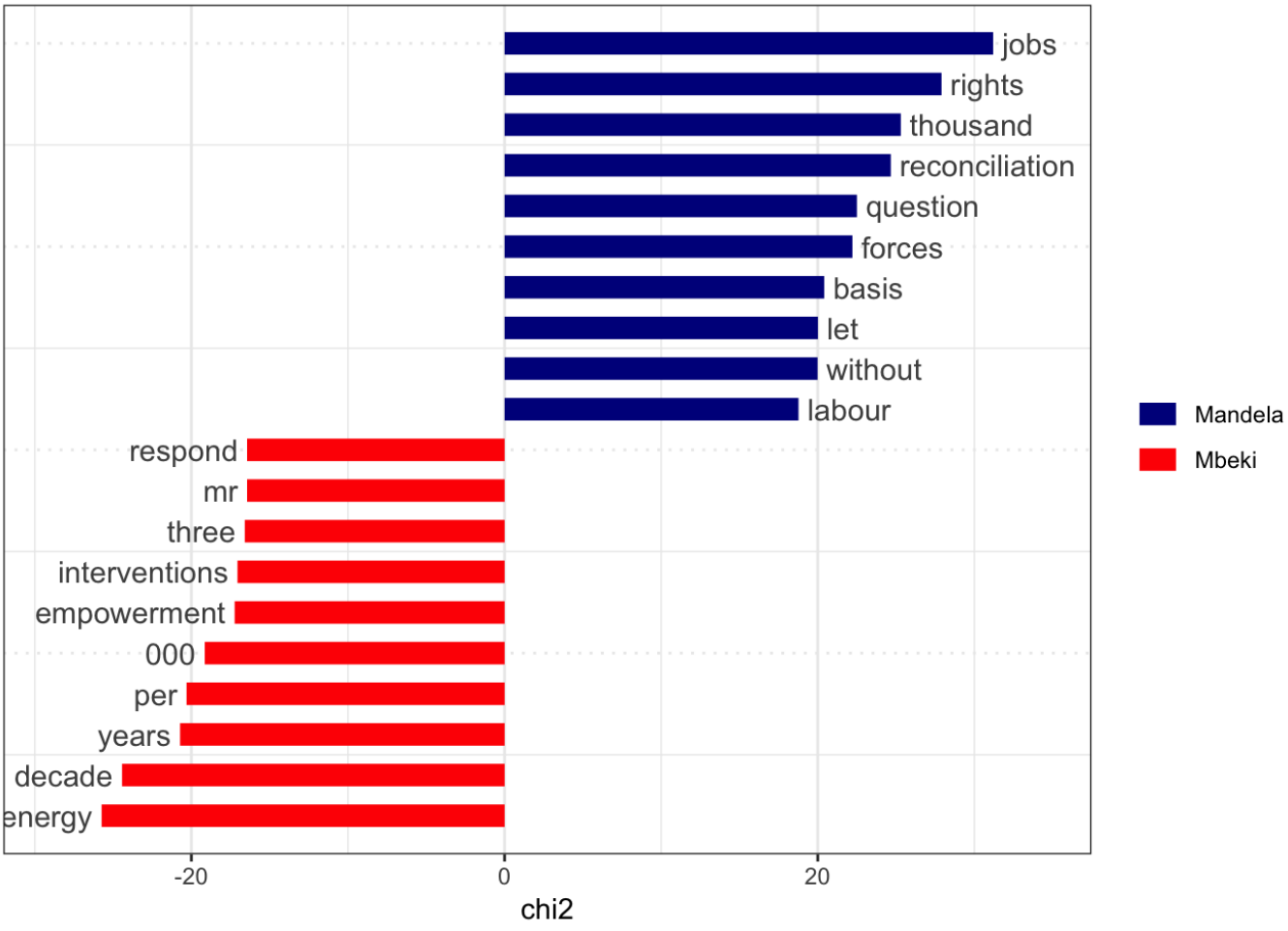

alliance
constitution
powers
transitional
party
constitutional
concerns
zulu
freedom
election
need
rogramme
crime
take
among
us
shall
work
year
public
0
100
200
chi2
deKlerk
Mandela

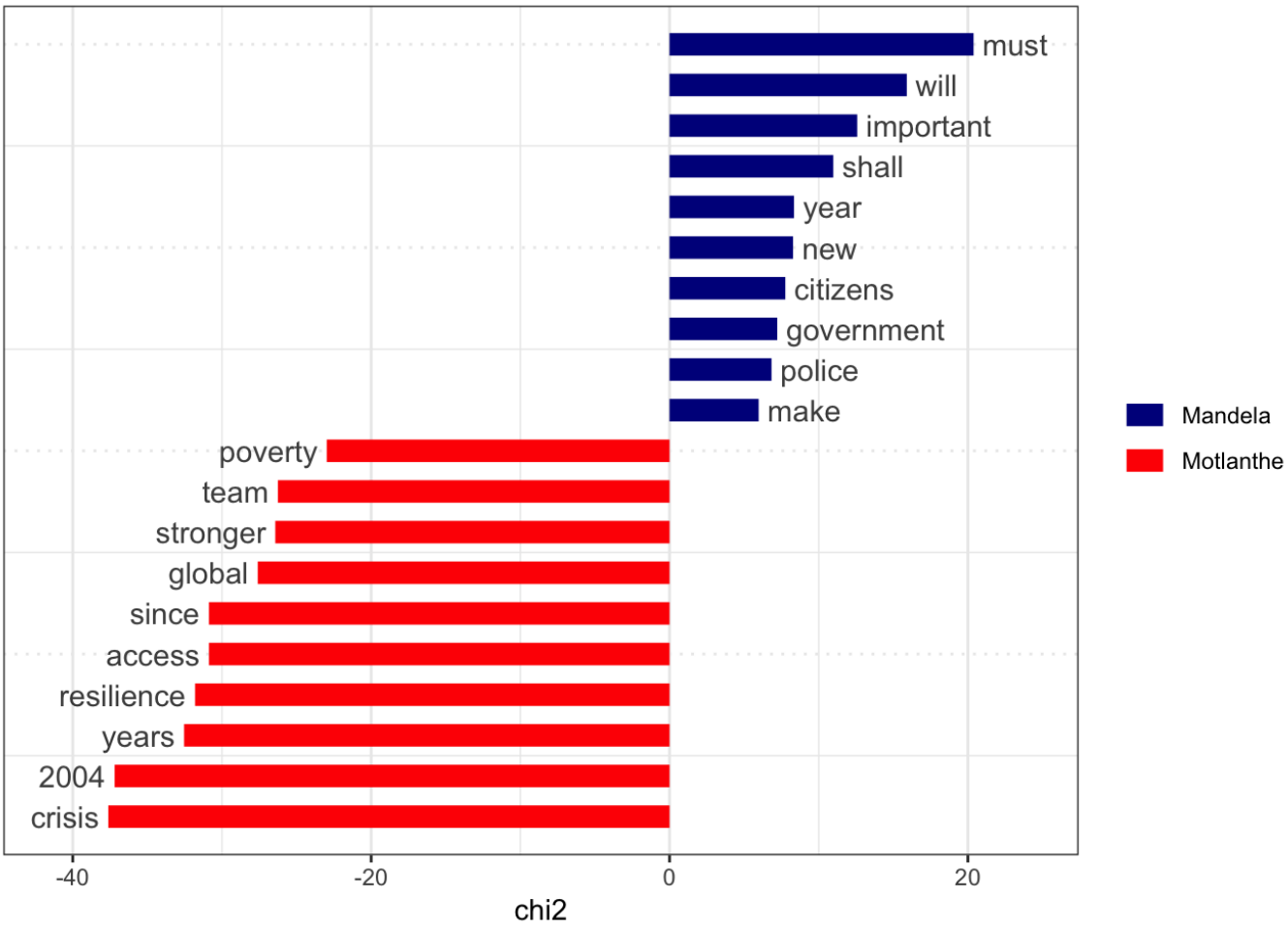

alliance
transitional
constitution
powers
constitutional
zulu
party
election
amend
parties
ensure
social
world
public
us
velopment
rogramme
work
country
year
0
100
200
300
400
chi2
deKlerk
Mbeki

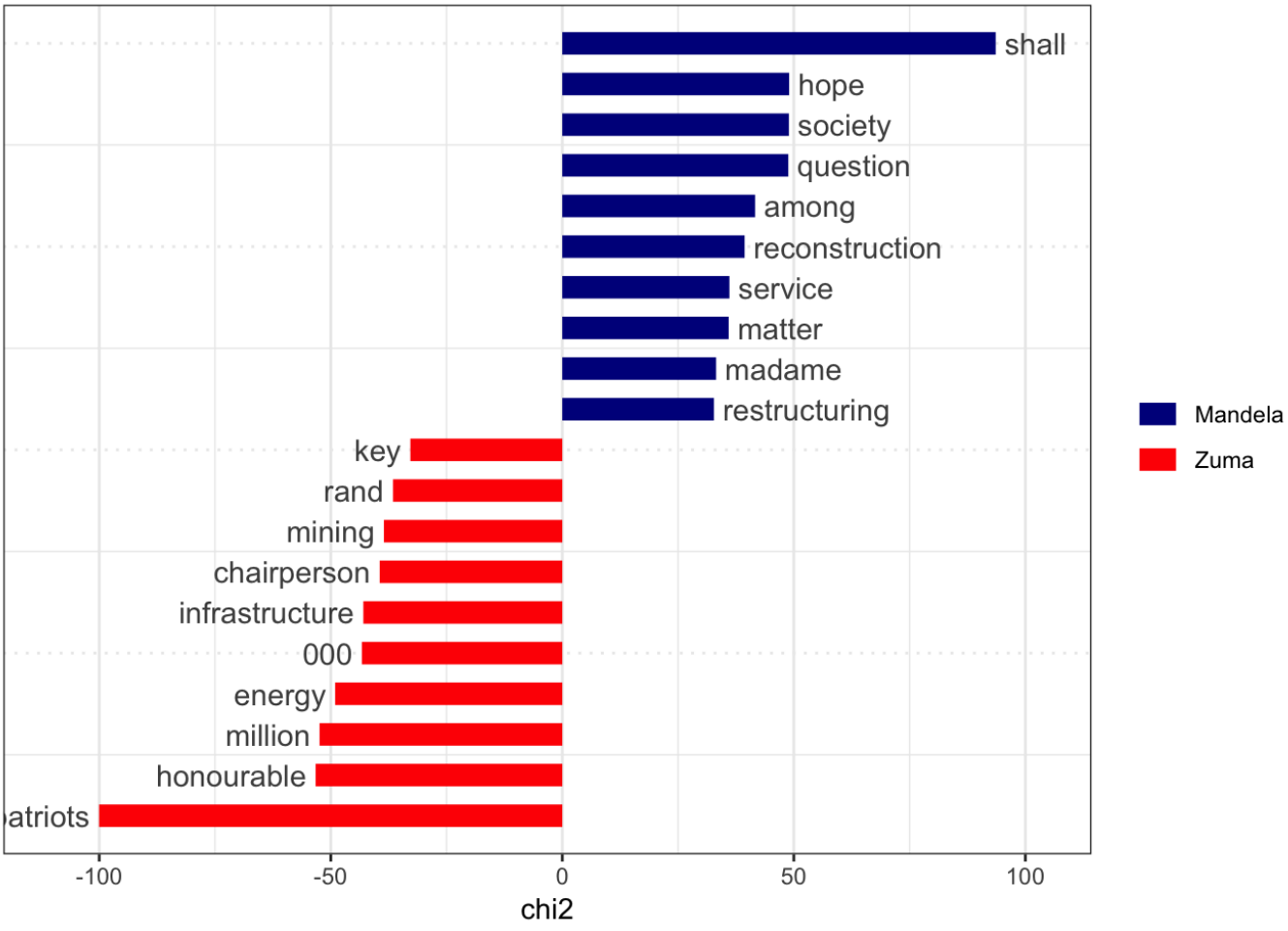

constituti
alliance
constitutional
powers
transitional
election
parties
will
party
freedom
world
hope
access
since
lemocracy
years
among
poverty
social
public
0
20
40
60
chi2
deKlerk
Motlanthe

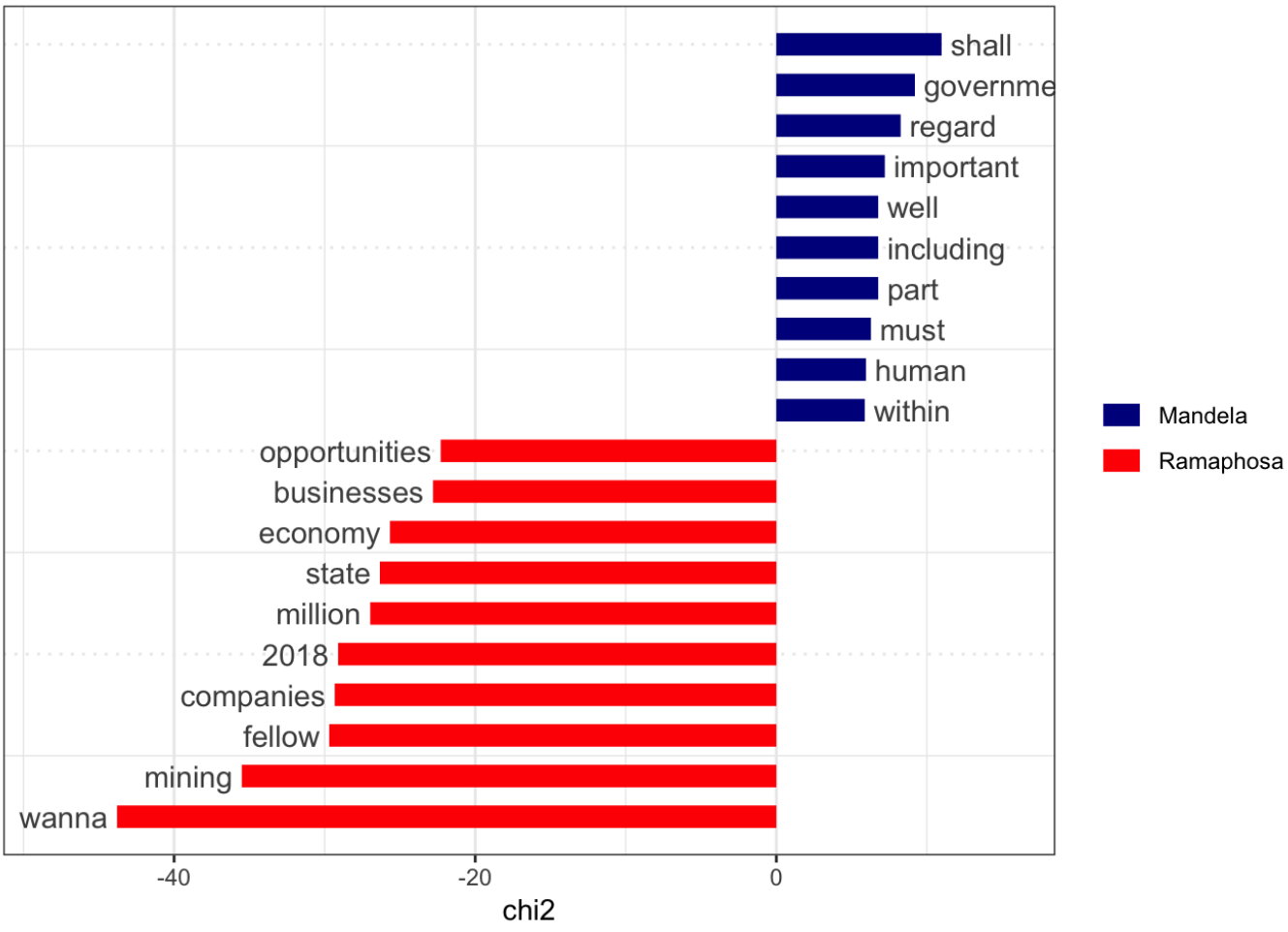

alliance
constitutional
transitional
constitution
election
powers
parties
zulu
party
freedom
sector
members
also
million
country
rogramme
mpatriots
onourable
work
year
0
100
200
300
chi2
deKlerk
Zuma

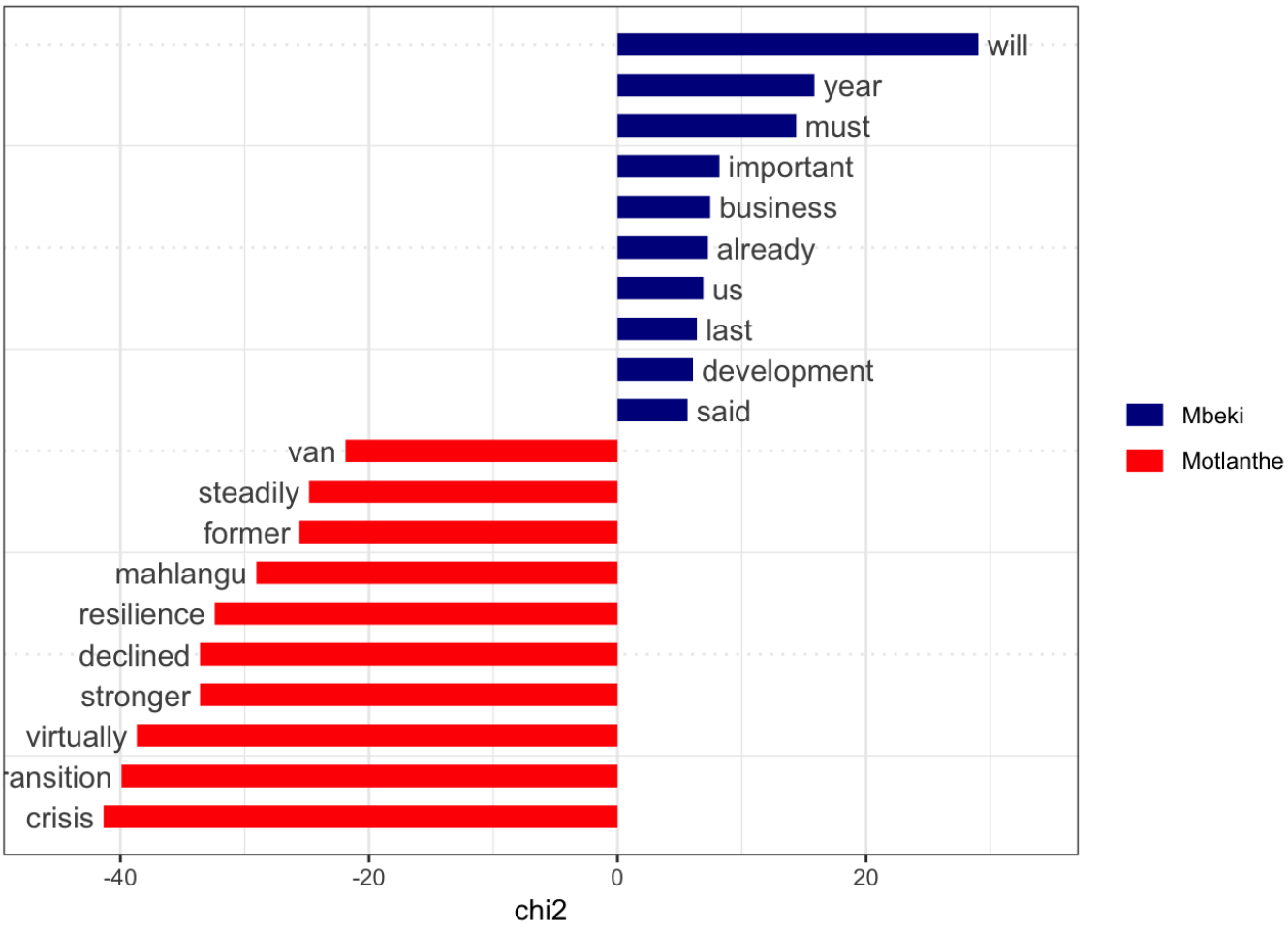

constitut
constitutional
alliance
parties
freedom
powers
transitional
election
party
provinces
measures
investment
sector
build
public
people
country
ensure
work
year
0
20
40
60
chi2
deKlerk
Ramaphosa

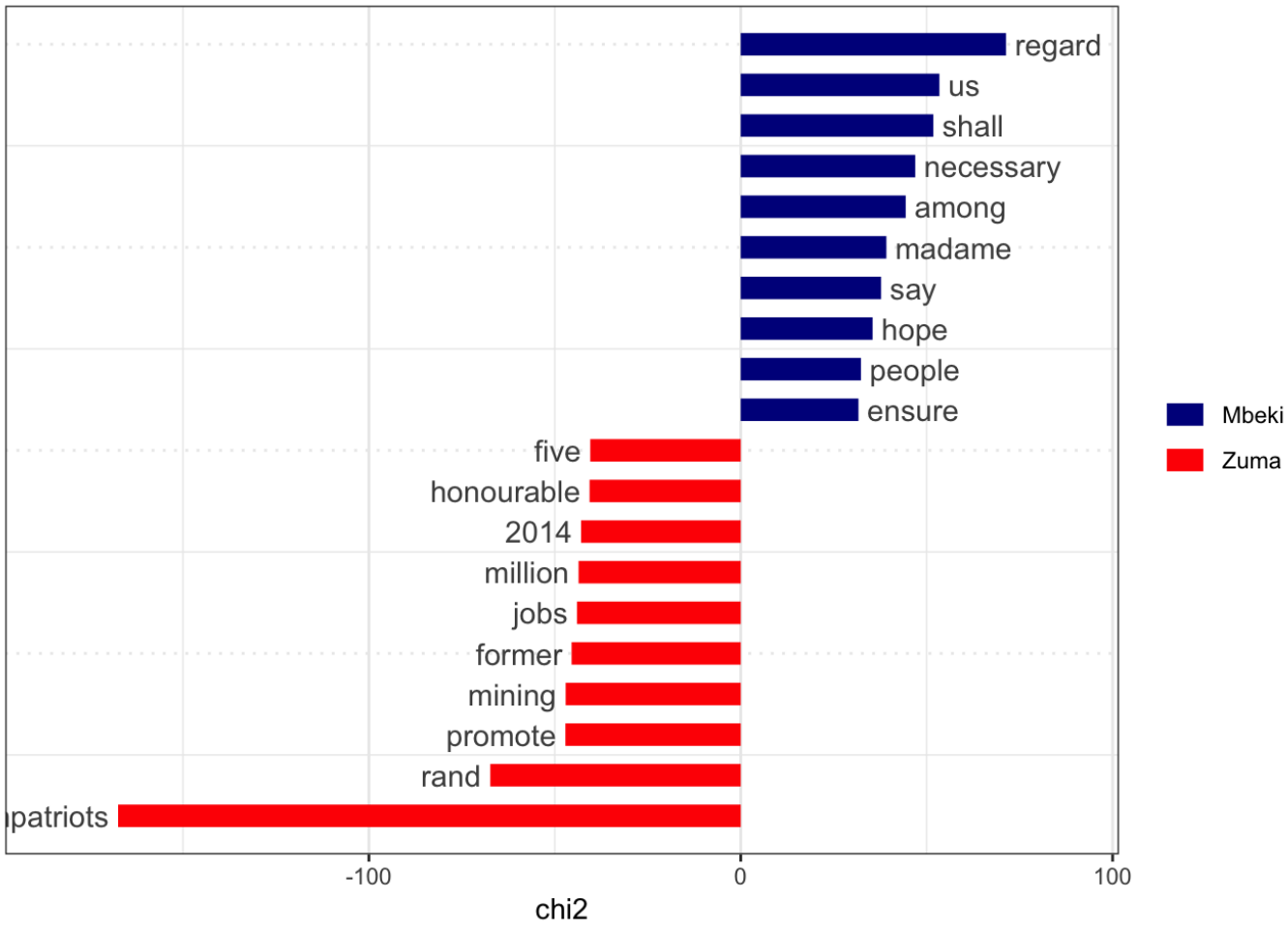

jobs
rights
thousand
reconciliation
question
forces
basis
let
without
labour
respond
mr
three
interventions
empowerment
000
per
years
decade
energy
-20
0
20
chi2
Mandela
Mbeki

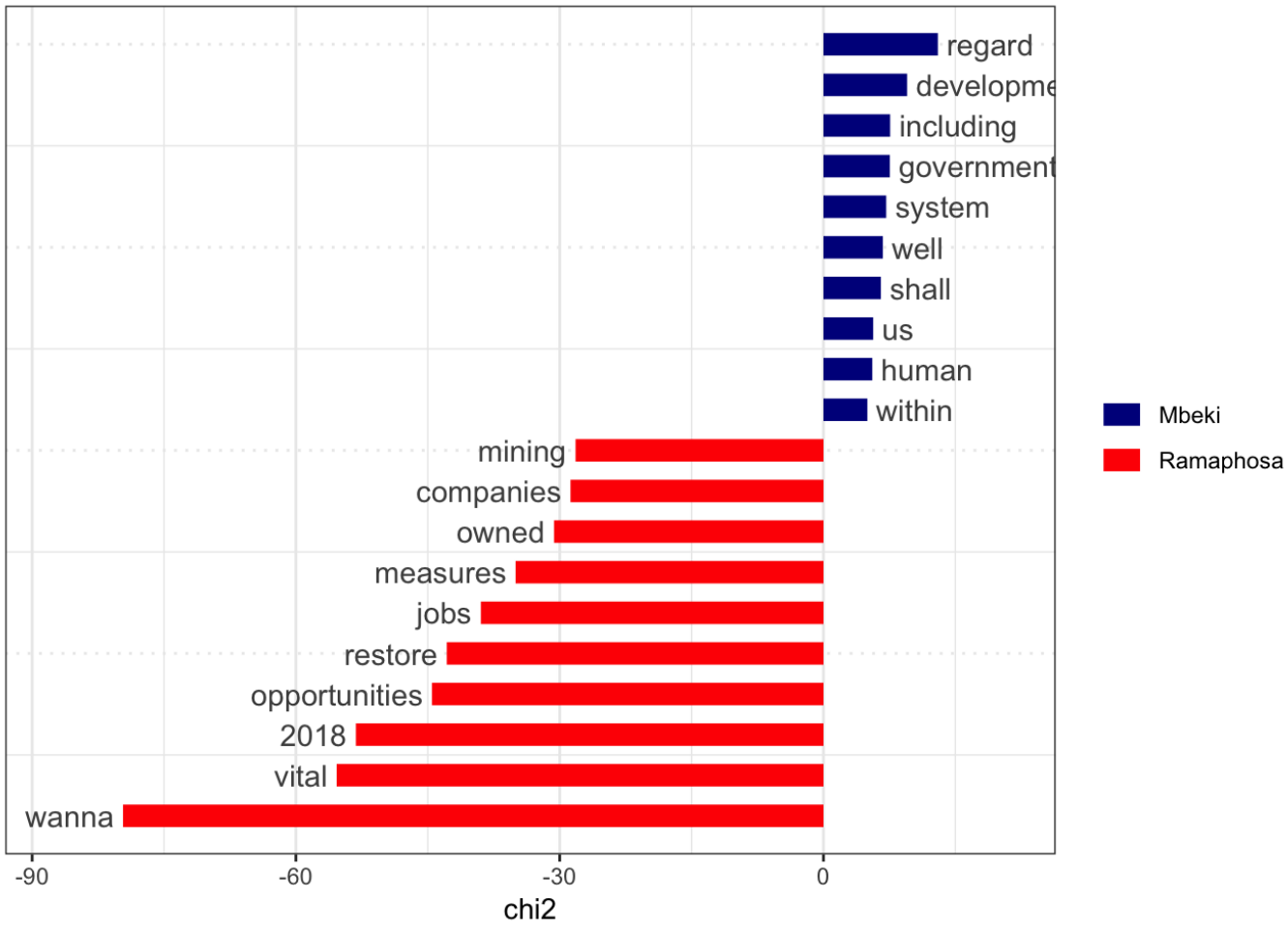

must
will
important
shall
year
new
citizens
government
police
make
poverty
team
stronger
global
since
access
resilience
years
2004
crisis
-40
-20
0
20
chi2
Mandela
Motlanthe

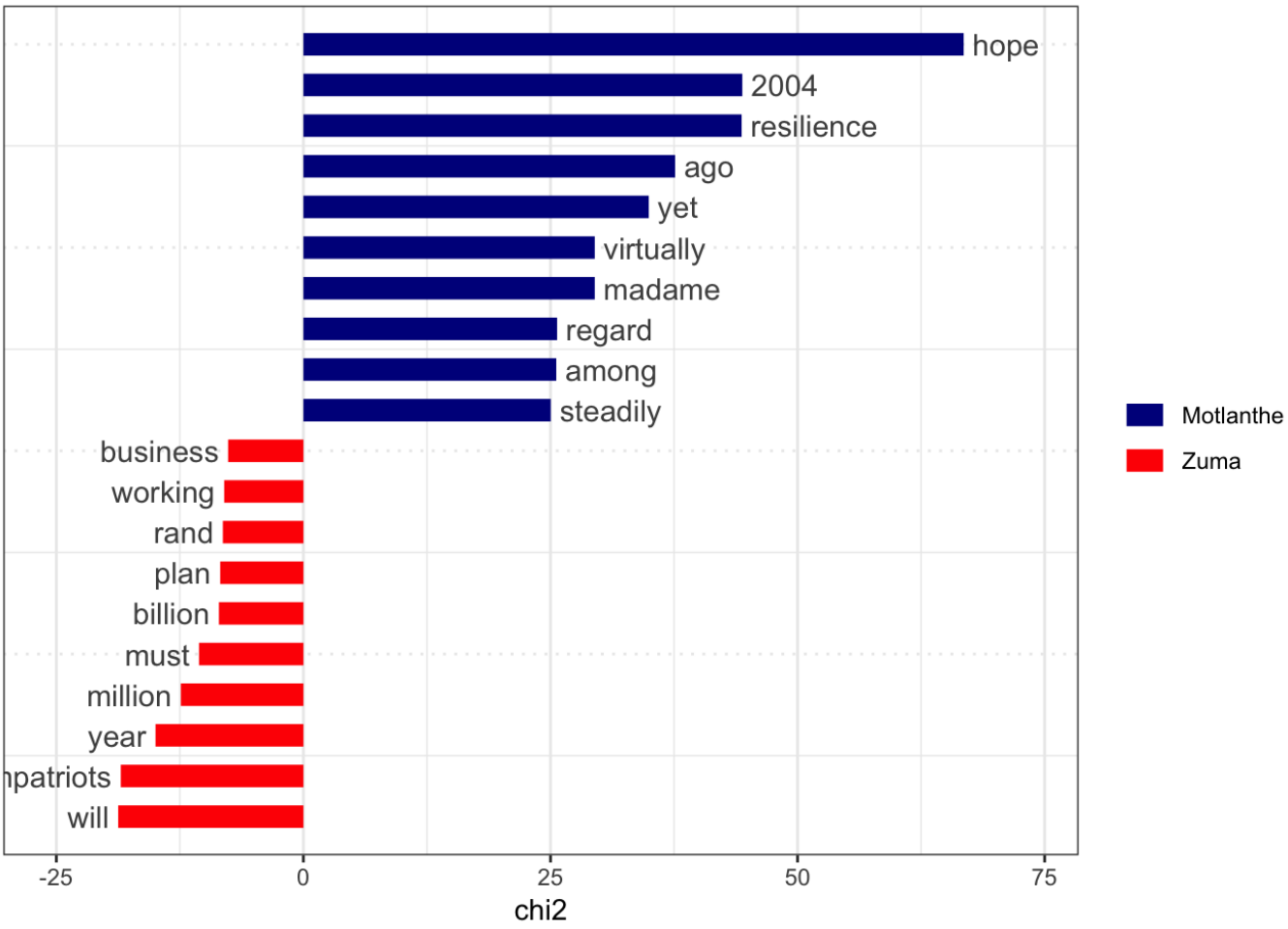

shall
hope
society
question
among
reconstruction
service
matter
madame
restructuring
key
rand
mining
chairperson
infrastructure
000
energy
million
honourable
atriots
-100
-50
0
50
100
chi2
Mandela
Zuma

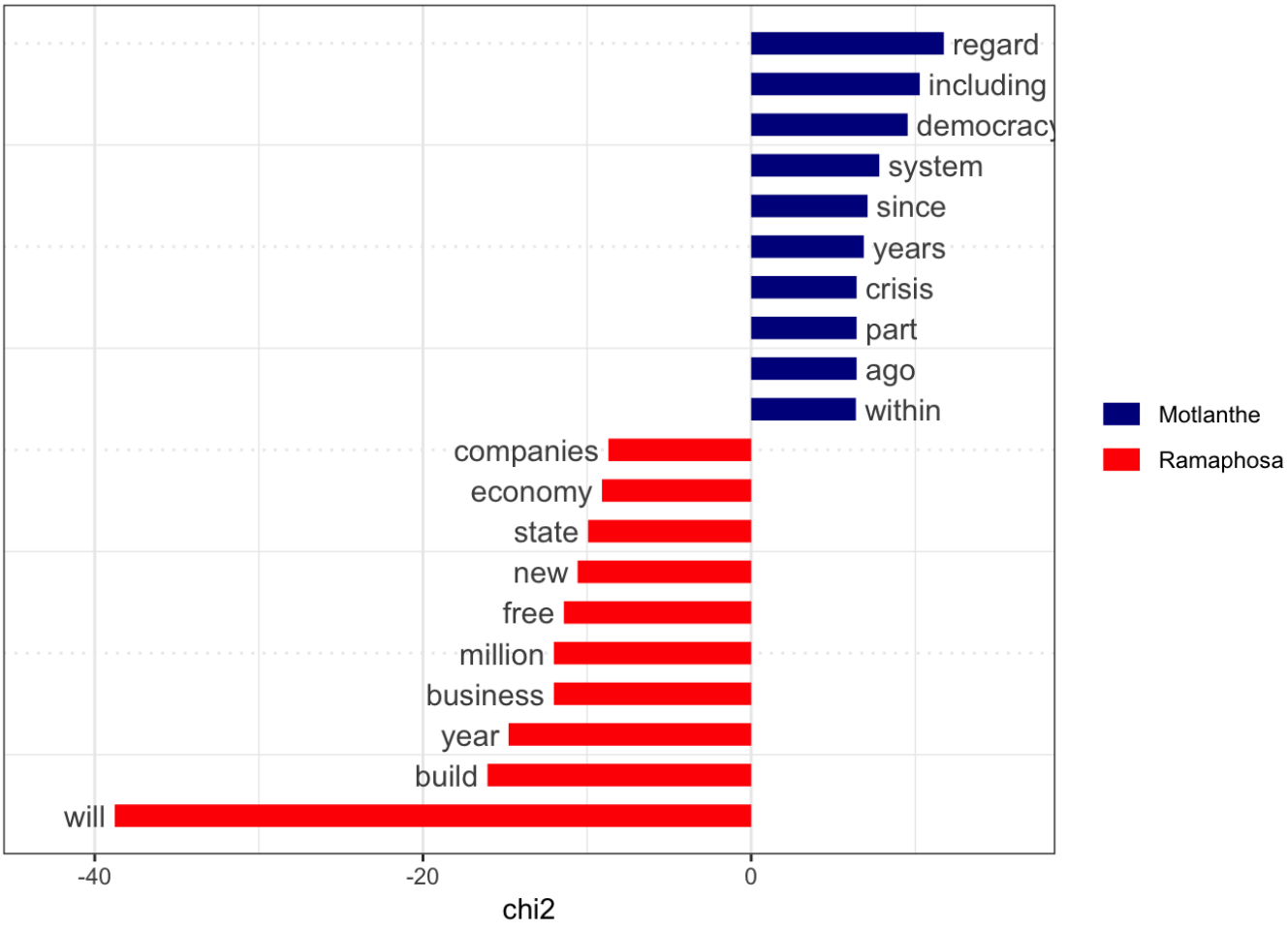

shall
governme
regard
important
well
including
part
must
human
within
opportunities
businesses
economy
state
million
2018
companies
fellow
mining
wanna
-40
-20
0
chi2
Mandela
Ramaphosa

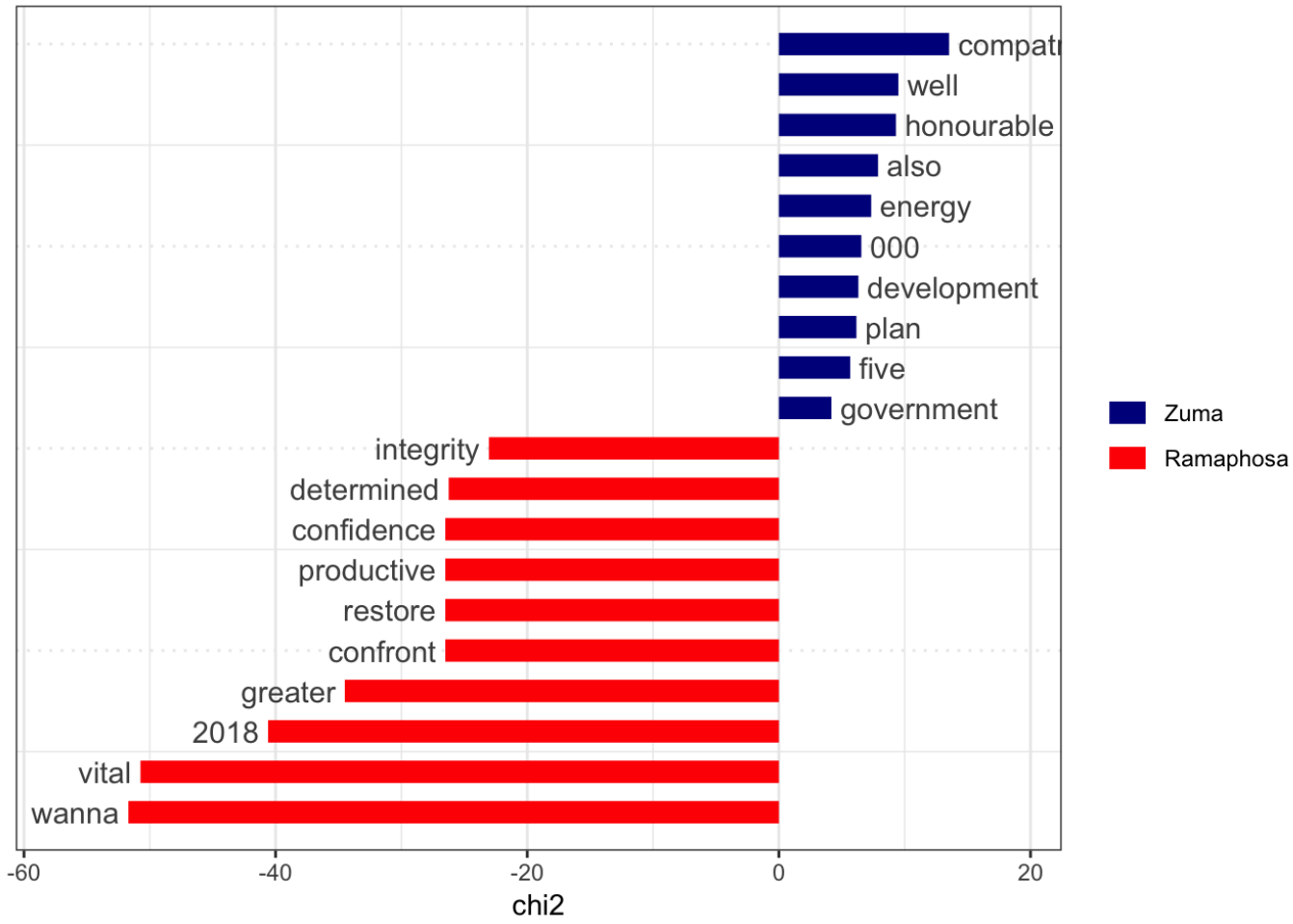

will
year
must
important
business
already
us
last
development
said
van
steadily
former
mahlangu
resilience
declined
stronger
virtually
ransition
crisis
-40
-20
0
20
chi2
Mbeki
Motlanthe


regard
us
shall
necessary
among
madame
say
hope
people
ensure
five
honourable
2014
million
jobs
former
mining
promote
rand
patriots
-100
0
100
chi2
Mbeki
Zuma


regard
developme
including
government
system
well
shall
us
human
within
mining
companies
owned
measures
jobs
restore
opportunities
2018
vital
wanna
-90
-60
-30
0
chi2
Mbeki
Ramaphosa


hope
2004
resilience
ago
yet
virtually
madame
regard
among
steadily
business
working
rand
plan
billion
must
million
year
will
-25
0
25
50
75
chi2
Motlanthe
Zuma


regard
including
democracy
system
since
years
crisis
part
ago
within
companies
economy
state
new
free
million
business
year
build
will
-40
-20
0
chi2
Motlanthe
Ramaphosa



For example, in the first plot shown we can see that President Mandela was more likey to sue the words jobs, rights and thousand when compared to President Mbeki. While President Mbeki was more likely to use energy, decade and years.