Dictionary Validation

This R markdown document contains the code to validate the climate change dictionary used to identify climate change related speeches in the Congressional Record and Hansard.

Setup

Labelling climate change speeches

To validate the climate change dictionary, we first need to create a labelled dataset of climate change speeches from Hansard and the Congressional Record. To do this, we first take a random selection of speeches from Hansard made in 2008. Selecting speeches from 2008 means that the sample is not representative of the entire dataset. However, given that it was the year the UK Climate Change Act was passed, it ensures we find some positive examples of climate change speeches. After making this random selection, we do the same for the Congressional Record, instead selecting speeches from 2009. 2009 was the year that the American Clean Energy and Security Act stalled in the Senate.

Sampling Hansard and the Congressional Record

Loading Hansard

```
hansard <- read_csv(paste0(HANSARD_PATH, "hansard.csv")) |>
    filter(year == 2008)
```

Randomly selecting 500 speeches

```
set.seed(42)
hansard_sample <- hansard |>
    sample_n(500)
write_csv(hansard_sample, paste0(DATA_PATH, "hansard_sample.csv"))
```

Loading the Congressional Record

```
congressional_record <-
    read_csv(paste0(CONGRESSIONAL_RECORD_PATH, "congressional_record.csv")) |>
    filter(year == 2009)
```

Randomly selecting 500 speeches

```
set.seed(42)
congressional_record_sample <- congressional_record |>
    sample_n(500)
write_csv(
    congressional_record_sample,
    paste0(DATA_PATH, "congressional_record_sample.csv")
)
```

At this point, human coding is used to label the speeches as either climate change related or not.

Testing dictionary performance

We can now test the performance of the climate change dictionary on the labelled dataset based on the rules used in `filtering.ipynb`.

Test set preprocessing

Congressional Record

```
classified_congressional_record <- read_csv(
    paste0(DATA_PATH, "classified_congressional_record_sample.csv")
)

classified_congressional_record_corpus <-
    corpus(classified_congressional_record, text_field = "cleaned_stems")

classified_congressional_record_dfm <-
    classified_congressional_record_corpus |>
    tokens() |>
    dfm()
```

Hansard

```
classified_hansard <-
    read_csv(paste0(DATA_PATH, "classified_hansard_sample.csv"))

classified_hansard_corpus <-
    corpus(classified_hansard, text_field = "cleaned_stems")

classified_hansard_dfm <-
    classified_hansard_corpus |>
    tokens() |>
    dfm()
```

Validating performance

Performance statistics function

```
calculate_confusion_statistics <- function(dfm, dictionary, prop_threshold) {
    # Perform the dictionary lookup on the dfm
    dict_scores <- dfm_lookup(dfm, dictionary = dictionary)
    dict_scores <- convert(dict_scores, to = "data.frame")

    # Add the calculated columns to the original data
    df <- as.data.frame(docvars(dfm))
    df <- df %>%
        mutate(
            num_climate_stems = dict_scores$climate_stems,
            prop_climate_stems = num_climate_stems / stem_count
        ) %>%
        mutate(
            predicted_climate_change_content = if_else(
                prop_climate_stems > prop_threshold,
                TRUE,
                FALSE
            )
        )

    # Create the confusion matrix
    confusion_table <- table(
        predicted_classification = df$predicted_climate_change_content,
        actual_classification = df$climate_change_content
    )
```

```
    # Compute confusion matrix statistics
    confusion_statistics <- confusionMatrix(confusion_table, positive = "TRUE")

    return(confusion_statistics)
}
```

`climate_stems` dictionary

Initialising the dictionary

```
climate_stems <-
    read_csv(paste0(DICITIONARIES_PATH, "climate_stems.csv"))

climate_stems_list <- list(
    climate_stems = climate_stems$stem
)

climate_stems_dictionary <-
    dictionary(climate_stems_list)
```

Congressional Record

```
calculate_confusion_statistics(
    classified_congressional_record_dfm,
    climate_stems_dictionary,
    0.2
)
```

```
## Confusion Matrix and Statistics
##
##                          actual_classification
## predicted_classification FALSE TRUE
##                    FALSE   448   32
##                    TRUE      9   11
##
##                Accuracy : 0.918
##                  95% CI : (0.8904, 0.9405)
##     No Information Rate : 0.914
##     P-Value [Acc > NIR] : 0.4135945
##
##                   Kappa : 0.3116
##
```

```
##  Mcnemar's Test P-Value : 0.0005908
##
##             Sensitivity : 0.2558
##             Specificity : 0.9803
##          Pos Pred Value : 0.5500
##          Neg Pred Value : 0.9333
##              Prevalence : 0.0860
##          Detection Rate : 0.0220
##    Detection Prevalence : 0.0400
##       Balanced Accuracy : 0.6181
##
##        'Positive' Class : TRUE
##
```

Hansard

```
calculate_confusion_statistics(
    classified_hansard_dfm,
    climate_stems_dictionary,
    0.2
)
```

```
## Confusion Matrix and Statistics
##
##                          actual_classification
## predicted_classification FALSE TRUE
##                    FALSE   459   12
##                    TRUE     14   15
##
##                Accuracy : 0.948
##                  95% CI : (0.9247, 0.9658)
##     No Information Rate : 0.946
##     P-Value [Acc > NIR] : 0.4723
##
##                   Kappa : 0.5082
##
##  Mcnemar's Test P-Value : 0.8445
##
##             Sensitivity : 0.5556
##             Specificity : 0.9704
##          Pos Pred Value : 0.5172
##          Neg Pred Value : 0.9745
##              Prevalence : 0.0540
##          Detection Rate : 0.0300
```

```
##    Detection Prevalence : 0.0580
##       Balanced Accuracy : 0.7630
##
##        'Positive' Class : TRUE
##
```

shortened_climate_stems dictionary

Initialising the dictionary

```
shortened_climate_stems <-
    read_csv(paste0(DICITIONARIES_PATH, "shortened_climate_stems.csv"))

shortened_climate_stems_list <- list(
    climate_stems = shortened_climate_stems$stem
)

shortened_climate_stems_dictionary <-
    dictionary(shortened_climate_stems_list)
```

Congressional Record

```
calculate_confusion_statistics(
    classified_congressional_record_dfm,
    shortened_climate_stems_dictionary,
    0.015
)
```

```
## Confusion Matrix and Statistics
##
##                          actual_classification
## predicted_classification FALSE TRUE
##                    FALSE   453   29
##                    TRUE      4   14
##
##                Accuracy : 0.934
##                  95% CI : (0.9086, 0.9541)
##     No Information Rate : 0.914
##     P-Value [Acc > NIR] : 0.06079
##
##                   Kappa : 0.4301
##
##  Mcnemar's Test P-Value : 2.943e-05
```

```
##
##             Sensitivity : 0.3256
##             Specificity : 0.9912
##          Pos Pred Value : 0.7778
##          Neg Pred Value : 0.9398
##              Prevalence : 0.0860
##          Detection Rate : 0.0280
##    Detection Prevalence : 0.0360
##       Balanced Accuracy : 0.6584
##
##        'Positive' Class : TRUE
##
```

Hansard

```
calculate_confusion_statistics(
    classified_hansard_dfm,
    shortened_climate_stems_dictionary,
    0.04
)
```

```
## Confusion Matrix and Statistics
##
##                          actual_classification
## predicted_classification FALSE TRUE
##                    FALSE   471   18
##                    TRUE      2    9
##
##                Accuracy : 0.96
##                  95% CI : (0.9389, 0.9754)
##     No Information Rate : 0.946
##     P-Value [Acc > NIR] : 0.0951929
##
##                   Kappa : 0.4567
##
##  Mcnemar's Test P-Value : 0.0007962
##
##             Sensitivity : 0.3333
##             Specificity : 0.9958
##          Pos Pred Value : 0.8182
##          Neg Pred Value : 0.9632
##              Prevalence : 0.0540
##          Detection Rate : 0.0180
##    Detection Prevalence : 0.0220
```

```
##       Balanced Accuracy : 0.6646
##
##        'Positive' Class : TRUE
##
```