Sentiment analysis and LSF

Replication of Lima-Lopes (2020) - Part 3

Rodrigo Esteves de Lima-Lopes
State University of Campinas
rll307@unicamp.br

Contents

1 Introduction

This script will discuss how to prepare our dictionaries and count the sentiments. My analysis replicates Lima-Lopes (2020) and is based on SFL (Systemic-Functional Linguistics) approach (Halliday and Mathissen, 2013), mostly specifically the Appraisal System (Martin and White, 2005).

Our objectives here are twofold:

1. Prepare the dictionaries
2. Process the counting and normalisation

In both cases, we are using Quanteda as our main package.

2 Packages

```
library(quanteda)
library(dplyr)
```

Each package has a function:

- `quanteda`: text analysis (counting the features)
- `dplyr`: data manipulation

3 Dictionaries

Dictionaries are formal classes which associate keys (some classificatory values) with elements in a list. The idea of a dictionary is to identify `character/strings` with a pre-defined category, in order to tell us how any of each class is present in a corpus. It is important to notice that a corpus might have one or more texts in its structure.

There are a number of possibilities, packages and methods, in `R` to work with dictionaries. In this tutorial, we are going to use `quanteda` for its easiness of use and because it brings tools for other analysis as we will discuss latter.

3.1 Creating the dictionaries

Our first step is to create a number of vectors with each categories we would need to analyse in the corpus.

```
adj.1 <-  c("squat", "filthy", "dirty", "behaviour", "very", "unrestricted", "awful",
            "black", "hungry", "old", "dangerous", "fake", "legitimate", "bad", "bogus",
            "hard", "racist", "temporary", "high", "proper english",  "proper ticket",
            "english", "properly", "significant", "top", "white citizen", "high hate",
            "high increase", "high since", "high on record", "genuine", "poor", "destitute",
            "frustrate", "poor level english", "european squatter", "poor foreigner", "contamination",
            "hovel", "smuggler", "smuggle", "big", "older", "higher", "proper tickets",
            "english properly", "white citizens", "highest since", "highest on record",
            "frustrated", "poor levels of english", "european squatters", "poor foreigners",
            "hovels", "smuggled", "biggest", "english proper",'increase', 'white', 'citizen',
            'ticket','foreigner') |> unique()

adj.2 <- c("unskilled migrant", "low-skilled", "skilled-worker", "jobless",  "qualification",
           "sharp mind", "skilled migration", "qualify", "honest", "workforce", "academic",
           "bright", "clean", "damage", "exceptional",  "extraordinary", "lower",
           "unemployed", "highly-skilled", "sharp", "unskilled worker", "unskille migrant",
           "low-skill", "skill-work", "job", "skill migration", "unemploy", "highly-skill",
           "unskill worker", "exception", "sharp", "mind", "skilled","migration",
           "skill","unskill","worker", "work","low","ordinary
           workers","workers","ordinary","welfare") |> unique()

adj.3 <-  c("legal", "illegal", "unlawful", "unlawfully", "asylum seekers", "refugee",
            "genuine refugees", "commonwealth", "unauthorized", "temporary migrants",
            "unauthorized", "genuine", "legal", "illegal", "unlawful", "unlawfully",
            "asylum seeker", "refugee", "genuine refugee", "commonwealth", "unauthorise",
            "temporary migrant", "migrant", "migrants", "unauthorize", "unauthorize") |> unique()

MC.1 <- c("spain", "from outside eu", "ethnic minorities", "portugal", "non-eea countries",
          "from europe", "from eastern europe", "from poorer eu countries", "beyond",
          "europe", "romania", "bulgaria", "afghanistan", "bulgarian", "syrian",
          "vietnamese", "mexican", "iranian", "iran", "foreigner", "foreign", "nigerian",
          "african", "indian", "iraqi", "turkish", "kurdish", "muslim", "asian",
          "eu migrants", "europeans", "european workers", "euro-immigrants", "european-based migration"
          "europe's immigration", "eastern european", "eastern europe", "east europeans",
          "mediterranean", "east", "cultural", "greek", "international", "overseas", "american",
          "global", "native", "polish", "white immigrants", "britons", "native counterparts",
          "local", "australian", "national", "former", "jamaican", "economic refugees", "economic migra
          "eu citizens", "indian national", "goa", "nigeria", "eastern european workers",
```

```
          "migrant workers", "spain", "outside eu", "ethnic minority", "portugal", "non-eea country",
          "europe", "eastern europe", "from poor eu country", "europe", "from romania and bulgaria", "a
          "foreigner", "foreign", "nigerian", "african", "indian", "iraqus", "turkish",
          "kurdish", "muslim", "asian", "eu migrant", "european", "european worker",
          "euro-immigrant", "european-base", "europe 's immigration", "eastern european", "eastern euro
          "east european", "mediterranean", "east", "cultural", "greek",
          "international", "overseas", "american", "global", "native", "polish", "white immigrant", "br
          "jamaican", "economic", "refugee", "economic migrant", "eu citizen", "indian national", "goa"
          'eu',"abroad",'eurozone')  |> unique()

MC.2 <-  c("home secretary", "theresa may", "mrs. may", "cabinet source", "mr johnson",
           "chairman", "mr farage", "nigel farage", "boris johnson", "trump", "spokesperson",
           "Migrationwatch", "Official statisticians", "statistics", "study", "she", "labour leader",
           "australian", "latest figures", "latest migration", "latest research", "latest census", "hon
           "added", "say", "prime minister", "president", "home secretary", "theresa may", "mr. may", "
           "boris johnson", "trump", "spokesperson", "Migrationwatch", "Official statistician", "statis
           "latest migration", "latest research", "latest census", "honest", "sure",
           "spokeswoman", "spokesman", "economist", "economist", "mayor", "add", "say",
           "prime minister", "president", "may", "johnson", "minister", "secretary",
           "secretary", "farage", "nigel", "boris")  |> unique()

MC.3 <- c("particularly", "very", "dramatic", "passionately", "directly", "quickly",
          "completely", "certainly", "controversi", "nearly", "actu", "illeg", "below",
          "longer", "work abroad", "politic", "immediately respond", "immediately",
          "clearly", "absurd", "bigger", "disastrous", "diverse", "easy", "harder",
          "awkward", "easier", "effective", "greater", "impossible", "minimum",
          "negative", "embarrassing", "medical", "radical", "rude", "attractive", "stronger", "tougher"
          "uncontrolled", "weak", "large", "proper policy", "proper border controls", "proper controls"
          "mass", "imigration", "mass migration", "official", "huge", "right", "strongest",
          "clearest", "easiest", "estimated", "lowest", "surest", "trickiest", "unrestricted",
          "hopeful", "housing", "arriving", "massive", "particular", "very", "dramatic",
          "passionate", "direct", "quick", "complete", "certain", "controverse", "near",
          "actu", "illeg", "below", "long", "work abroad", "politic", "immediate respond",
          "immediate", "clear", "absurd", "big", "disastrous", "diverse", "easy", "hard",
          "awkward", "easy", "effective", "great", "impossible", "minimum", "negative", "embarrass",
          "medical", "radical", "rude", "attractive", "strong", "tough", "uncontrol", "weak", "large",
          "high migrant demand", "high level", "less", "total", "mass", "imigration", "mass migration",
          "sure", "tricky", "unrestricted", "hopeful", "house", "arrive", "massive") |> unique()

vp <- c("show", "stop illegal immigrants", "stop migrants", "says", "say", "said", "squat",
        "stow", "stowaway", "coming", "born", "work", "go", "think", "left", "allow", "warn",
        "warned", "live", "trying", "stagnate", "dwindle", "trafficking", "traffick",
        "collapse", "flee", "cost", "added", "stop", "migrants", "show", "stop illegal immigrant",
        "stop migrant", "say", "say", "say", "squat", "stow", "stowaway", "coming",
        "born", "work", "go", "think", "left", "allow", "warn", "warn", "live", "try",
        "stagnate", "dwindle", "traffic", "traffick", "collapse", "flee", "cost", "add",
        "stop", "migrant") |> unique()


EM <- c("should", "would", "must", "need", "will", "can", "can't", "shouldn't",
        "might","may","must", "have to","could","ought to")
```

```
NP.1 <- c("poorer EU countries", "poorer member states", "poorer EU member", "European economies",
          "eurozone", "euro", "economy", "Economic", "welfare", "housing", "Euro",
          "recession", "single market", "dwindle", "cost", "poor EU country", "poor member state",
          "poor EU member", "European economy", "eurozone", "euro", "economy", "Economic",
          "welfare", "house", "Euro", "recession", "single market", "dwindle", "cost",
          "poor", "EU", "country", "states", "member", "European", "Europe")  |> unique()

NP.2 <- c("control", "borders", "border", "wave", "waves", "surge", "increase",
          "uncontrolled", "immediately blocked", "minimum", "mass migration",
          "unrestricted", "weak", "student visa", "student visas", "proper border controls",
          "proper controls", "highest", "high migrant demand", "high levels",
          "visa system", "mass", "visa", "demand", "level", "levels", "control",
          "border", "border", "wave", "wave", "surge", "increase", "uncontrol",
          "immediately block", "minimum", "mass migration", "unrestricted", "weak",
          "student visa", "student visa", "proper border control", "proper control",
          "high", "high migrant demand", "high level", "visa system", "mass", "visa",
          "demand", "level") |> unique()
```

Now we create a list with our vectors

```
dic.list <- list(adj.1, adj.2, adj.3, MC.1, MC.2, MC.3, vp, EM, NP.1, NP.2)
```

Then we name such a list

```
names(dic.list) <- c("adj.1", "adj.2", "adj.3", "MC.1", "MC.2", "MC.3", "vp", "EM", "NP.1", "NP.2")
```

Finally, we create the dictionary

```
dic.brexit <- dictionary(dic.list)
```

3.2 Creating the corpus

Before we apply the dictionary, we will create a corpus like we did in **02_Sentiment_pre_processing_network**. However, we will not lemmatise the corpus.

```
# one by one
TS.Corpus <- corpus(TS.df, text_field = 'Content')
DS.Corpus <- corpus(DS.df, text_field = 'Content')
TT.Corpus <- corpus(TT.df, text_field = 'Content')

#General
News.Corpus <- rbind(TT.df,TS.df, DS.df)
News.Corpus <- corpus(News.Corpus, text_field = 'Content')

#tokens
TS.tokens  <- tokens(TS.Corpus,
                     what = "word",
                     remove_punct = TRUE,
                     remove_symbols = TRUE,
                     remove_numbers = TRUE,
                     remove_url = TRUE,
```

```
                      split_hyphens = FALSE,
                      include_docvars = TRUE,
                      padding = FALSE,
                      verbose = quanteda_options("verbose")
)

DS.tokens <- tokens(DS.Corpus,
                    what = "word",
                    remove_punct = TRUE,
                    remove_symbols = TRUE,
                    remove_numbers = TRUE,
                    remove_url = TRUE,
                    split_hyphens = FALSE,
                    include_docvars = TRUE,
                    padding = FALSE,
                    verbose = quanteda_options("verbose")
)
TT.tokens <- tokens(TT.Corpus,
                    what = "word",
                    remove_punct = TRUE,
                    remove_symbols = TRUE,
                    remove_numbers = TRUE,
                    remove_url = TRUE,
                    split_hyphens = FALSE,
                    include_docvars = TRUE,
                    padding = FALSE,
                    verbose = quanteda_options("verbose")
)
News.tokens <- News.Corpus |>
  tokens(what = "word",
         remove_punct = TRUE,
         remove_symbols = TRUE,
         remove_numbers = TRUE,
         remove_url = TRUE,
         split_hyphens = FALSE,
         include_docvars = TRUE,
         padding = FALSE,
         verbose = quanteda_options("verbose")
  )
```

3.3 Counting the occurences

Now let us do the actual counting

```
News.counting <- dfm(News.tokens, dictionary = dic.brexit)
```

```
## Warning: 'dictionary' and 'thesaurus' are deprecated; use dfm_lookup() instead
```

```
DS.counting <- dfm(DS.tokens, dictionary = dic.brexit)
```

```
## Warning: 'dictionary' and 'thesaurus' are deprecated; use dfm_lookup() instead
```

```
TS.counting <- dfm(TS.tokens, dictionary = dic.brexit)
```

```
## Warning: 'dictionary' and 'thesaurus' are deprecated; use dfm_lookup() instead
```

```
TT.counting <- dfm(TT.tokens, dictionary = dic.brexit)
```

```
## Warning: 'dictionary' and 'thesaurus' are deprecated; use dfm_lookup() instead
```

Saving as data frame for latter use

```
DS.counting <- convert(DS.counting, to = "data.frame")
TS.counting <- convert(TS.counting, to = "data.frame")
TT.counting <- convert(TT.counting, to = "data.frame")
News.counting <- convert(News.counting, to = "data.frame")
```

Viewing the results

```
News.counting |> head()
```

```
##   doc_id adj.1 adj.2 adj.3 MC.1 MC.2 MC.3 vp EM NP.1 NP.2
## 1    TT1     0     9     1   10    1    4  4  8   14    1
## 2    TT2     4     7     0   12    5   12  7 27   12   10
## 3    TT3     7    12     6   17   23   15 22 36   19    4
## 4    TT4     3     2    15   12    7   25 15 16    6    8
## 5    TT5     1     5     6   13    7    5 21  3   25    3
## 6    TT6     0     5     1    1    1    3  8  4    0    3
```

3.3.1 Normalising and preparing the analysis

Now we are going to add a total column and use the name of the documents to name de rows

```
News.counting$Total <- rowSums(News.counting[,2:10])
rownames(News.counting) <- News.counting$doc_id
```

Now we normalise and round the data

```
News.Cal  <- News.counting |>
  mutate_at(
    vars(-matches(c("Total", "doc_id"))),
    ~ .x * 100 / Total) |>
  mutate_at(
    vars(-matches(c("Total", "doc_id"))),
    ~ round(.x, 2)
  )
```

Now let us see it

```
News.Cal |> head()
```

```
##     doc_id adj.1 adj.2 adj.3  MC.1  MC.2  MC.3    vp    EM  NP.1  NP.2 Total
## TT1    TT1  0.00 17.65  1.96 19.61  1.96  7.84  7.84 15.69 27.45  1.96    51
## TT2    TT2  4.65  8.14  0.00 13.95  5.81 13.95  8.14 31.40 13.95 11.63    86
## TT3    TT3  4.46  7.64  3.82 10.83 14.65  9.55 14.01 22.93 12.10  2.55   157
## TT4    TT4  2.97  1.98 14.85 11.88  6.93 24.75 14.85 15.84  5.94  7.92   101
## TT5    TT5  1.16  5.81  6.98 15.12  8.14  5.81 24.42  3.49 29.07  3.49    86
## TT6    TT6  0.00 21.74  4.35  4.35  4.35 13.04 34.78 17.39  0.00 13.04    23
```