Session 7 POS NER

Christian Kahmann

12 10 2020

This tutorial shows how to use Part-o-Speech-tagging (POS) with the openNLP package.

Annotate POS

We extract proper nouns (tag NNP for singular and tag NNPS for plural proper nouns) from paragraphs of president's speeches.

```
options(stringsAsFactors = FALSE)
library(quanteda)
library(NLP)

# read suto paragraphs
textdata <- read.csv("data/sotu_paragraphs.csv", sep = ";", encoding = "UTF-8")
english_stopwords <- readLines("resources/stopwords_en.txt", encoding = "UTF-8")

# Create corpus object
sotu_corpus <- corpus(textdata$text, docnames = textdata$doc_id)

require(openNLP)
require(openNLPdata)

# openNLP annotator objects
sent_token_annotator <- Maxent_Sent_Token_Annotator()
word_token_annotator <- Maxent_Word_Token_Annotator()
pos_tag_annotator <- Maxent_POS_Tag_Annotator()
annotator_pipeline <- Annotator_Pipeline(
    sent_token_annotator,
    word_token_annotator,
    pos_tag_annotator
)

# function for annotation
annotateDocuments <- function(doc, pos_filter = NULL) {
  doc <- as.String(doc)
  doc_with_annotations <- annotate(doc, annotator_pipeline)
  tags <- sapply(subset(doc_with_annotations, type=="word")$features, `[[`, "POS")
  tokens <- doc[subset(doc_with_annotations, type=="word")]
  if (!is.null(pos_filter)) {
    res <- tokens[tags %in% pos_filter]
  } else {
    res <- paste0(tokens, "_", tags)
  }
  res <- paste(res, collapse = " ")
  return(res)
}
```

```
# run annotation on a sample of the corpus
annotated_corpus <- lapply(texts(sotu_corpus)[1:10], annotateDocuments)

# Have a look into the first annotated documents
annotated_corpus[1]
```

```
## $`1`
## [1] "Fellow-Citizens_NNS of_IN the_DT Senate_NNP and_CC House_NNP of_IN Representatives_NNPS :_:"
```

```
annotated_corpus[2]
```

```
## $`2`
## [1] "I_PRP embrace_VBP with_IN great_JJ satisfaction_NN the_DT opportunity_NN which_WDT now_RB presen
```

Filter NEs for further applications

We annotate the first paragraphs of the corpus, extract proper nouns, also referred to as Named Entities (NEs) such as person names, locations etc., and compute significance of co-occurrence of them.

```
sample_corpus <- sapply(texts(sotu_corpus)[1:1000], annotateDocuments, pos_filter = c("NNP", "NNPS"))

# Binary term matrix
require(Matrix)
minimumFrequency <- 2
filtered_corpus <- corpus(sample_corpus)

binDTM <- filtered_corpus %>%
  tokens(what = "fastestword") %>%
  tokens_tolower() %>%
  dfm() %>%
  dfm_weight(scheme = "boolean")

# Matrix multiplication for cooccurrence counts
coocCounts <- t(binDTM) %*% binDTM

source("resources/calculateCoocStatistics.R")
# Definition of a parameter for the representation of the co-occurrences of a concept
# Determination of the term of which co-competitors are to be measured.
coocTerm <- "indians"
coocs <- calculateCoocStatistics(coocTerm, binDTM, measure="LOGLIK")
print(coocs[1:20])
```

```
##        ohio      wabash    illinois   augustine     indiana   territory
##   24.124563    8.226914    8.226914    8.226914    7.107320    6.702815
##      united   pensacola      states     florida   executive     general
##    6.641194    6.264679    5.695900    5.599043    5.149651    5.036511
##     georgia      indian  chickasaws mississippi       creek     western
##    5.036511    4.727217    4.086121    3.425899    3.106909    3.106909
##    floridas       union
##    3.106909    2.530202
```

German language support

For German language support run

```
# install.packages("openNLPmodels.de", repos = "http://datacube.wu.ac.at")
# require("openNLPmodels.de")
```