Word order in UD

10/15/2024

This is the code used for Section 4.1.1 of the paper 'Why we need a gradient approach to word order'. https://doi.org/10.1515/ling-2021-0098

Read the data

The data originates from the corpora of the Universal Dependencies.

```
# read relevant packages
library(tidyverse)
library(zoo)
# remove scientific notations such as 10000=1e04
options(scipen=999,stringsAsFactors = FALSE)

# have the list of languages
Languages <- c("Arabic","Basque", "Bulgarian", "Catalan" , "Chinese", "Croatian", "Danish", "Dutch", "E

# take a smaller list if needed
Languages <- c("Basque","Catalan")

# open an empty table to store the output
data <- NULL %>% as.data.frame()

# extract data for the list of languages
for(z in c(1:length(Languages))){
  # take all the files in the folder of that language
  files <- list.files(paste("data_raw/UD/",Languages[z],"/",sep=""))
  # create and view an object with file names and full paths
  f <- file.path("data_raw/UD/",Languages[z], files)
  d <- lapply(f, FUN = function(files){read.delim(files,
                                                  header = FALSE,
                                                  comment.char = "#",
                                                  stringsAsFactors = FALSE)})
  # combine all the files that were read
  merge.data <- plyr::rbind.fill(d)
  # add the language annotations
  merge.data <- merge.data %>%
    mutate(Language = Languages[z])
  # merge with the entire data
  data <- rbind(data, merge.data)
}
# remove not used vectors
rm(merge.data, d)
```

```
# arrange the columns of the table
data <- data %>%
  select(ID_word = V1, Tag = V6, POS = V4, Lemma = V3,
         Dependency = V7, Role = V8, Language)

# print the data as a table
data %>% write.csv("data_raw/UD.csv",
                   row.names = FALSE,
                   fileEncoding = "UTF-8")

# visual check
glimpse(data)
```

```
## Rows: 652,498
## Columns: 7
## $ ID_word    <chr> "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "1", "2"~
## $ Tag        <chr> "Case=Ine|Definite=Def|Number=Sing", "_", "_", "_", "NumTyp~
## $ POS        <chr> "PROPN", "CCONJ", "PUNCT", "DET", "NUM", "NOUN", "DET", "VE~
## $ Lemma      <chr> "Atenas", "ordea", ",", "beste", "bost", "jarduera", "gehia~
## $ Dependency <chr> "8", "8", "8", "6", "6", "8", "6", "0", "8", "8", "2", "5",~
## $ Role       <chr> "obl", "advmod", "punct", "det", "nummod", "nsubj", "det", ~
## $ Language   <chr> "Basque", "Basque", "Basque", "Basque", "Basque", "Basque",~
```

Subject and Object

We first need to re-arrange the UD data by adding sentence IDs.

```
# if your computer is slow, can read the output file from the previous chunk
data <- read.csv("data_raw/UD.csv")

# adding start and end of sentences
data <- data %>%
  # change IDs to numeric
  mutate(ID_word = as.numeric(ID_word)) %>%
  # add gap of IDs between consecutive pair of words
  mutate(diff = ID_word - lag(ID_word, default = first(ID_word))) %>%
  # change NAs to 0s if needed
  replace(is.na(.), 0) %>%
  # add labels
  mutate(ID_sentence = case_when(diff < 0 ~ "New_sentence",
                                 diff >= 0 ~ "In"))

# change new sentence markers to sentence number
data$ID_sentence[which(data$ID_sentence == "New_sentence")] <- 2:(length(data$ID_sentence[which(data$ID
# manually add the start of the first sentence
data$ID_sentence[1] <- 1

# arrange the data
data <- data %>%
  # change the sentence ID to numeric
  mutate(ID_sentence = as.numeric(ID_sentence)) %>%
  # remove the diff column
  select(-diff)

# change NAs to the sentence ID
data$ID_sentence <- na.locf(data$ID_sentence)
# print the data as a table
data %>% write.csv("data_tidy/UD.csv",
                   row.names = FALSE,
                   fileEncoding = "UTF-8")

# visual check
glimpse(data)
```

```
## Rows: 652,498
## Columns: 8
## $ ID_word     <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 1, 2, 3, 4, 5, 6, 7, 8, 1, ~
## $ Tag         <chr> "Case=Ine|Definite=Def|Number=Sing", "_", "_", "_", "NumTy~
## $ POS         <chr> "PROPN", "CCONJ", "PUNCT", "DET", "NUM", "NOUN", "DET", "V~
## $ Lemma       <chr> "Atenas", "ordea", ",", "beste", "bost", "jarduera", "gehi~
## $ Dependency  <chr> "8", "8", "8", "6", "6", "8", "6", "0", "8", "8", "2", "5"~
## $ Role        <chr> "obl", "advmod", "punct", "det", "nummod", "nsubj", "det",~
## $ Language    <chr> "Basque", "Basque", "Basque", "Basque", "Basque", "Basque"~
## $ ID_sentence <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3~
```

for each sentence extract the order of SVO (this chunk for sampling).

```
# if needed, read the output from the previous chunks
data <- read.csv("data_tidy/UD.csv")

# open an empty table
data.sample <- NULL %>% as.data.frame()

for(w in seq(20,2000,20)){
tmp <- data %>%
  filter(Role %in% c("nsubj","obj")) %>%
  filter(POS == "NOUN") %>%
  # take relevant columns
  select(ID_sentence, ID_word, Dependency, Role, Language) %>%
  # change columns to numeric if needed
  mutate(Dependency = as.numeric(Dependency),
         ID_word = as.numeric(ID_word)) %>%
  #can play with this setting to see if want to take samples by roles too
  group_by(Language,Role) %>%
  sample_n(size = w, replace = T) %>%
  ungroup() %>%
  mutate(position = ID_word-Dependency) %>%
  mutate(position = case_when(position < 0 ~ "before_verb",
                              position > 0 ~ "after_verb"))  %>%
  # add the ratio
  group_by(Language, position) %>%
  mutate(count = n()) %>%
  group_by(Language) %>%
  mutate(total = n()) %>%
  ungroup() %>%
  mutate(ratio = count/total) %>%
  mutate(seq = w)

data.sample <- rbind(data.sample, tmp)

}
glimpse(data.sample)
```

```
## Rows: 404,000
## Columns: 10
## $ ID_sentence <int> 4313, 5966, 6967, 407, 2471, 5471, 3267, 3488, 5574, 4053,~
## $ ID_word     <dbl> 8, 14, 17, 1, 1, 1, 4, 18, 5, 13, 1, 10, 10, 5, 8, 1, 1, 2~
## $ Dependency  <dbl> 6, 19, 23, 4, 4, 4, 10, 16, 3, 16, 3, 22, 8, 6, 10, 3, 7, ~
## $ Role        <chr> "nsubj", "nsubj", "nsubj", "nsubj", "nsubj", "nsubj", "nsu~
## $ Language    <chr> "Basque", "Basque", "Basque", "Basque", "Basque", "Basque"~
## $ position    <chr> "after_verb", "before_verb", "before_verb", "before_verb",~
## $ count       <int> 7, 33, 33, 33, 33, 33, 33, 7, 7, 33, 33, 33, 7, 33, 33, 33~
## $ total       <int> 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40, 40~
## $ ratio       <dbl> 0.175, 0.825, 0.825, 0.825, 0.825, 0.825, 0.825, 0.175, 0.~
## $ seq         <dbl> 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20~
```

for each sentence extract the order of SVO

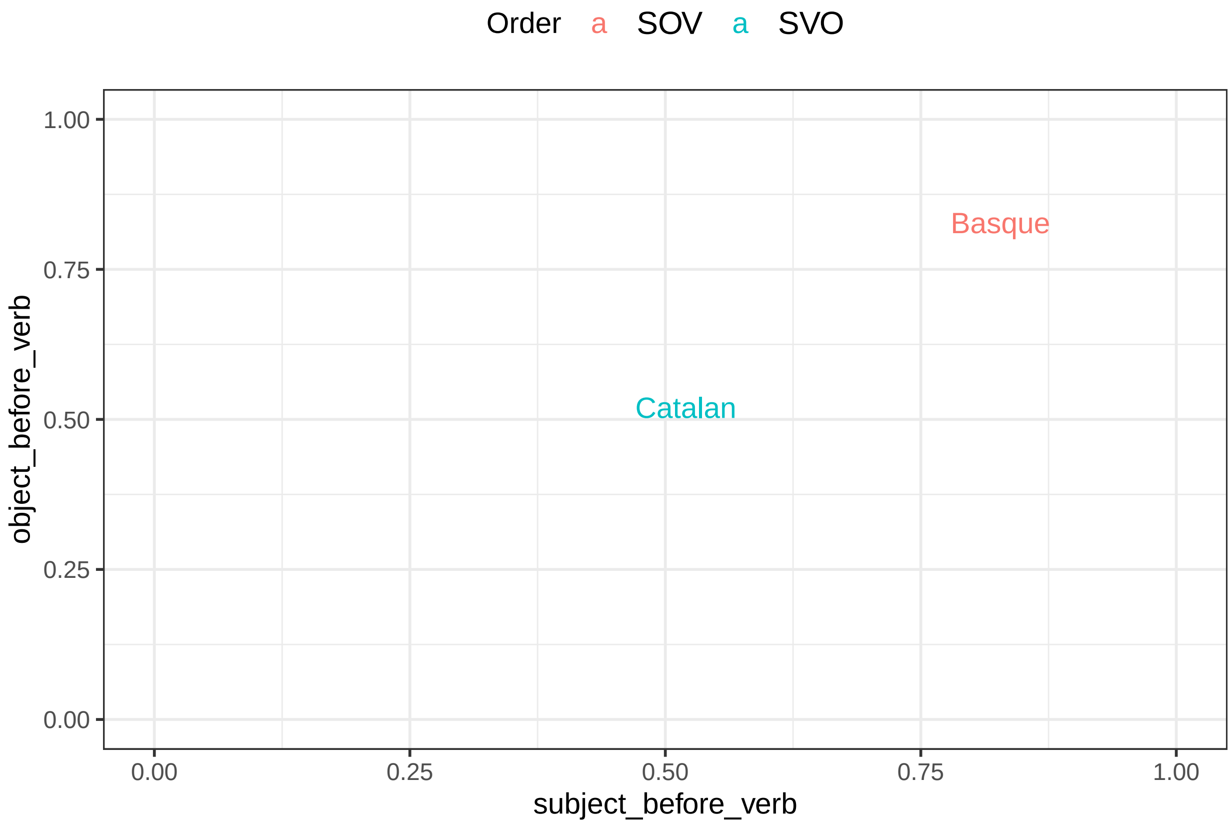
```
# if needed, read the output from the previous chunks
data <- read.csv("data_tidy/UD.csv")

# get the ratio
data <- data %>%
  # only keep the words that are subjects and objects
  filter(Role %in% c("nsubj","obj")) %>%
  # take relevant columns
  select(ID_sentence, ID_word, Dependency, Role, Language) %>%
  # change columns to numeric if needed
  mutate(Dependency = as.numeric(Dependency),
         ID_word = as.numeric(ID_word)) %>%
  # get the relative position
  mutate(position = ID_word-Dependency) %>%
  mutate(position = case_when(position < 0 ~ "before_verb",
                              position > 0 ~ "after_verb")) %>%
  # add the ratio
  group_by(Language, position) %>%
  mutate(count = n()) %>%
  group_by(Language) %>%
  mutate(total = n()) %>%
  ungroup() %>%
  mutate(ratio = count/total)
```

Plot the results with all the data

```
# can change between data or data sample
data %>%
  # take the ratio of S/O before verb
```

```
filter(position == "before_verb") %>%
# take relevant columns
select(Language, Role, ratio) %>%
# remove duplicates
distinct() %>%
# change the format to wide
pivot_wider(names_from = Role, values_from = ratio) %>%
# annotate values from WALS
mutate(Order = case_when(Language == "Arabic" ~ "VSO",
                         Language == "Basque" ~ "SOV",
                         Language == "Bulgarian" ~ "SVO",
                         Language == "Catalan" ~ "SVO",
                         Language == "Chinese" ~ "SVO",
                         Language == "Croatian" ~ "SVO",
                         Language == "Danish" ~ "SVO",
                         Language == "Dutch" ~ "No dominant",
                         Language == "English" ~ "SVO",
                         Language == "Estonian" ~ "SVO",
                         Language == "Finnish" ~ "SVO",
                         Language == "French" ~ "SVO",
                         Language == "Galician" ~ "",
                         Language == "German" ~ "No dominant",
                         Language == "Hebrew" ~ "SVO",
                         Language == "Hindi" ~ "SOV",
                         Language == "Hungarian" ~ "No dominant",
                         Language == "Indonesian" ~ "SVO",
                         Language == "Italian" ~ "SVO",
                         Language == "Japanese" ~ "SOV",
                         Language == "Korean" ~ "SOV",
                         Language == "Latin" ~ "",
                         Language == "Latvian" ~ "SVO",
                         Language == "Norwegian" ~ "SVO",
                         Language == "Persian" ~ "SOV",
                         Language == "Polish" ~ "SVO",
                         Language == "Portuguese" ~ "SVO",
                         Language == "Romanian" ~ "SVO",
                         Language == "Russian" ~ "SVO",
                         Language == "Serbian" ~ "SVO",
                         Language == "Slovak" ~ "",
                         Language == "Slovenian" ~ "SVO",
                         Language == "Spanish" ~ "SVO",
                         Language == "Swedish" ~ "SVO",
                         Language == "Ukrainian" ~ "SVO"
                         )) %>%
# remove not used languages
filter(!Language %in% c("Galician", "Latin", "Slovak")) %>%
# rename columns for plot
rename(subject_before_verb = nsubj, object_before_verb = obj) %>%
# make the plot
ggplot(aes(x = subject_before_verb, y= object_before_verb)) +
geom_text(aes(label = Language, color = Order), size = 4) +
# theme settings
theme_bw() +
  theme(legend.position = "top",
        legend.text = element_text(size = 12)) +
  # set length of x and y axes
  xlim(c(0,1)) +
  ylim(c(0,1))
```



```
#ggsave("order_200.png", dpi = 300, width = 8, height = 6)
```