A Brief Analysis of Lyric Metrics

Julia Stelman

11/11/2019

Introduction

In this mini analysis of lyric data, I explore a few features of lyric exerpts from 100 songs in 5 different languages. I compare 20 songs written in each language. I want to know if languages spoken across a wider geographical area conform more or less to taste in song lyrics. In other words, do the songs written by, say, Spanish speaking artists vary more, lyrically, than songs written by, say, Dutch artists? It would make sense, as Spanish is spoken all over the world, heightening the potential that this language has been split into different dialects by divergent evolution. Meanwhile, Dutch is only spoken, officially, in the Netherlands, a region so small that it doesn't provide the Dutch language very much potential for forming individual dialects.

The following two languages make up the *widely spoken* group:

- English (en)
- Spanish (es)

The following three languages make up the *not widely spoken* group:

- Indonesian (id)
- Dutch (nl)
- Swahili (sw)

The distinction between the first and second groups is decided by whether a language is an official national language on at least two continents.

I've used the natural language processing package $quanteda$ to calculate the following metrics for the first few lines of lyrics in each of the 100 sampled songs. (20 songs were randomly sampled from each of the 5 languages.)

- Types –> distinct words
- Tokens –> words
- Type/Token Ratio (TTR)

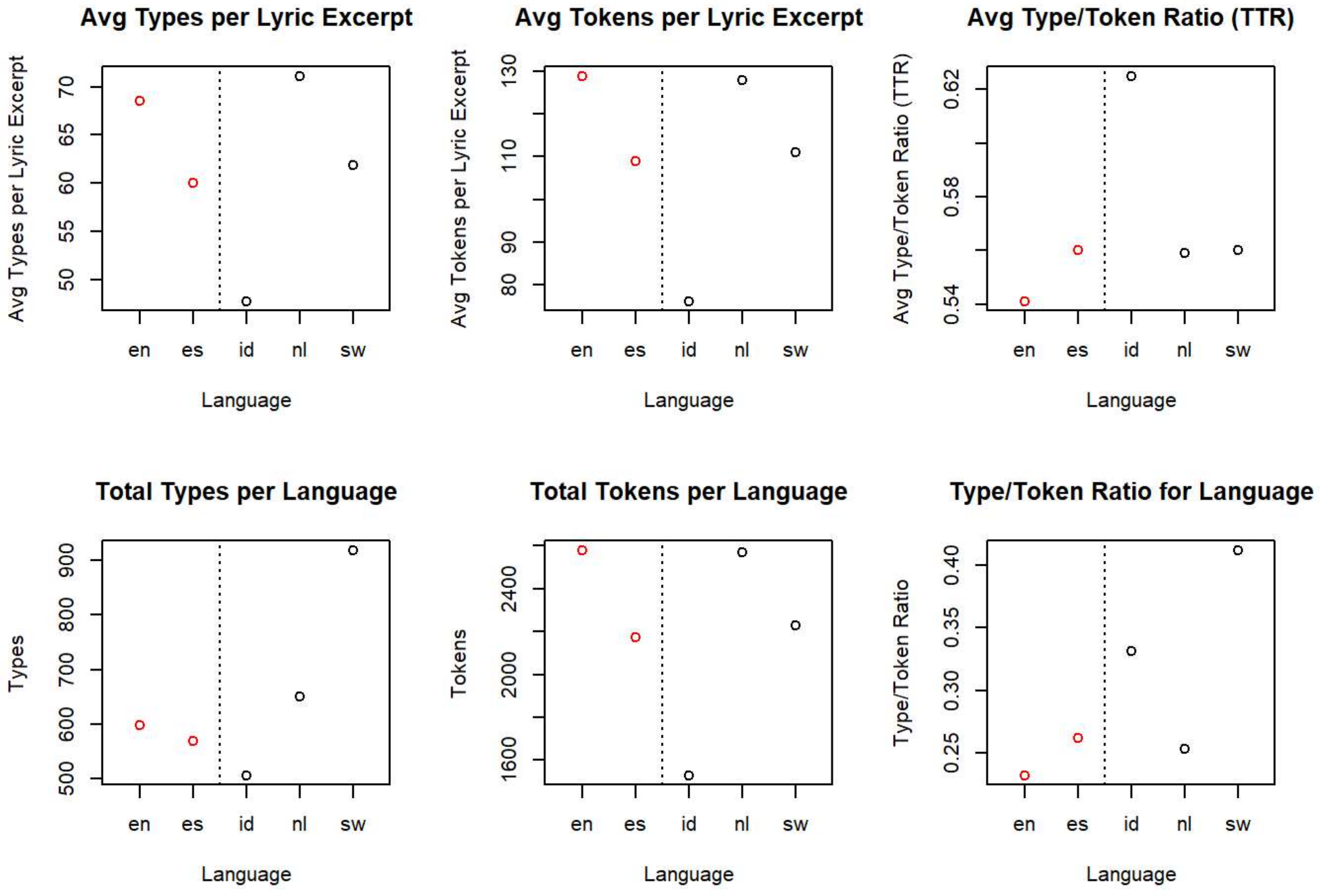
I summarized these as by-language averages in the table below.

Language	Avg Types per Lyric Excerpt	Avg Tokens per Lyric Excerpt	Avg Type/Token Ratio (TTR)
en	68.55	129	0.541
es	60.00	109	0.560
id	47.70	76	0.625
nl	71.15	128	0.559
sw	61.85	111	0.560

After frankensteining together the 20-song random sample of lyric excerpts from each language, I re-calculated the metrics again treating the combination of all sampled lyrics from one language as if it were from one song. They are shown in the table below. It's this information that will give me insight on what I'm really looking for.

Language	Types	Tokens	TTR
en	599	2580	0.232
es	569	2174	0.262
id	506	1528	0.331
nl	650	2569	0.253
sw	919	2229	0.412

In the plot below, the widely spoken languages are drawn in red as before.



It would seem that the Indonesian songs, on average, have rather few words, therefore inflating the Type/Token Ratio of individual Indonesian songs. This doesn't seem to apply when they are combined. It should be noted that TTR is a measure that is often criticized for how easily it can be manipulated by corpus size.

It looks like the words used in different Swahili songs have the least in common out of any of the five languages– and by a long shot! This is the absolute opposite of what I hypothesized earlier. I've thought of a few reasons for why this might be.

- Pop music has a more unified set of common words compared to many other genres. As opposed to artists who write lyrics in English or Spanish, artists who write songs in Swahili have a smaller target market. As a result, they may be more inclined to write song lyrics that they are inspired to write rather than song lyrics that will get them into the Top 40.
- The language detector is not perfect. Some songs get misclassified. Being as there was a small pool of Swahili songs to sample from in the first place, a few mistakes made a bigger difference in Swahili than in English and Spanish.

Thank you to Spotify, Musixmatch, Everynoise.com, and lang-detect for the help I got from your libraries, websites, and APIs in the data collection and cleaning phase of this project, which I did in Python.

Also thank you to the R libraries quanteda, tidyverse, DescTools, caret, dplyr, stringi, and knitr.

Appendix

```
AFsongDF <- read.csv("C:/Users/Julia Stelman/Desktop/Spotipy_language_experiments/AFsongDF.csv",
row.names=1)

dat <- na.exclude(AFsongDF[,c(-5,-16:-12,-23)])
dat$sid <- as.character(dat$sid)

source("C:/Users/Julia Stelman/Desktop/Semester I/Special Topics - Text Analysis/textstat_tools/
functions/keyness_functions.R")
source("C:/Users/Julia Stelman/Desktop/Semester I/Special Topics - Text Analysis/textstat_tools/
functions/dispersion_functions.R")
```

```r
# get the encoding types of each lyric excerpt.
stri_lang_enc = vapply(1:nrow(dat), function(x){
  stri_enc_detect(
    dat$lyrics[x], filter_angle_brackets = F
    )[[1]][c('Language','Encoding','Confidence')][1,] %>%
    unlist()},
  FUN.VALUE = c('Language','Encoding','Confidence')) %>% t() %>%
  as.data.frame()

# Make the text doc df for corpus analysis

lyrics_df2 <- dat %>%
  select(doc_id = sid, text = lyrics, lang) %>%
  cbind(stri_lang_enc) %>%
  mutate_if(is.factor, as.character) %>%
  # get rid of non-ISO encodings
  filter(Encoding %in% c("ISO-8859-1","ISO-8859-9","ISO-8859-2")) %>%
  # if lang.detect said English but R said something else, get rid of it
  dplyr::filter(ifelse(lang != "en", T, ifelse(lang == Language, T, F)))

langs <- data.frame(lang = c('en', 'es', 'id', 'sw', 'nl'), stringsAsFactors = F)

# only use the languages in the 5 specified
lyrics_df2 <- lyrics_df2 %>%
  inner_join(langs)

# clean up
rm(langs)

# take a smaller, evenly dispersed sample
set.seed(10)
## take an equal random sample from all five languages of interest
En_ids <- subset(x=lyrics_df2,subset = lang == 'en')$doc_id
## n = 20 because there are exactly 22 songs in Dutch, the language of the 5 with the fewest son
gs
En_ids <- sample(En_ids,20)
eS_ids <- subset(x=lyrics_df2,subset = lang == 'es')$doc_id
eS_ids <- sample(eS_ids,20)
Id_ids <- subset(x=lyrics_df2,subset = lang == 'id')$doc_id
Id_ids <- sample(Id_ids,20)
Nl_ids <- subset(x=lyrics_df2,subset = lang == 'nl')$doc_id
Nl_ids <- sample(Nl_ids,20)
sW_ids <- subset(x=lyrics_df2,subset = lang == 'sw')$doc_id
sW_ids <- sample(sW_ids,20)

esinw_df <- subset(x = lyrics_df2, select = names(lyrics_df2),
                   subset = doc_id %in% c(
                      En_ids, eS_ids, Id_ids, Nl_ids, sW_ids))
```

```
lyrics_corpus <- corpus(esinw_df)

# fetch a summary of the corpus composition
lyrics_sum <- summary(lyrics_corpus, tolower = T, n = 902) %>% select(- Sentences)

# Corpus composition table
lyrics_sum3ds <- lyrics_sum %>%
  mutate(Text = as.character(Text)) %>%
  mutate(TTR = Types/Tokens) %>%
  mutate(Language = lang)

# get all the averages we might want for the overall total
lyrics_sum_cct_avg <- lyrics_sum3ds %>%
  summarise("Texts" = length(Text),
            "Avg Tokens per Lyric Excerpt" = round(mean(Tokens)),
            "Tokens in Total" = sum(Tokens),
            "Avg Types per Lyric Excerpt" = round(mean(Types),2),
            "Avg Type/Token Ratio (TTR)" = round(mean(TTR),3)) %>%
  mutate(Language = "all") %>%
  select(Language, Texts, `Tokens in Total`, `Avg Tokens per Lyric Excerpt`, `Avg Types per Lyri
c Excerpt`, `Avg Type/Token Ratio (TTR)`)

# get the ones for each language
lyrics_sum_cct_va <- lyrics_sum3ds %>% group_by(Language) %>%
  summarise("Texts" = length(Text),
            "Avg Tokens per Lyric Excerpt" = round(mean(Tokens)),
            "Tokens in Total" = sum(Tokens),
            "Avg Types per Lyric Excerpt" = round(mean(Types),2),
            "Avg Type/Token Ratio (TTR)" = round(mean(TTR),3)) %>%  ungroup() %>%
  select(Language, Texts, `Tokens in Total`, `Avg Tokens per Lyric Excerpt`, `Avg Types per Lyri
c Excerpt`, `Avg Type/Token Ratio (TTR)`)

# bind them together
lyrics_sum_cct_va <- rbind(lyrics_sum_cct_va, lyrics_sum_cct_avg)

# now that we have that, lets choose just the parts we want to show in output this time
lyrics_sum_cct_simp <- lyrics_sum_cct_va  %>%
  filter(Language %in% c("en","es","id","nl","sw")) %>%
  select(Language, `Avg Types per Lyric Excerpt`, `Avg Tokens per Lyric Excerpt`, `Avg Type/Toke
n Ratio (TTR)`)

knitr::kable(lyrics_sum_cct_simp)
```

```
# frankenstein songs of the same language together
esinw_df_combined <- esinw_df %>% group_by(lang) %>%
  summarise("text" = paste(text, collapse = " ")) %>%
  select(doc_id = lang, text)

# And create a new corpus object based on this
lyrics_corpus_esinw <- corpus(esinw_df_combined)

# fetch a summary of the corpus composition
lyrics_sum_esinw <- summary(lyrics_corpus_esinw, tolower = T) %>% select(- Sentences)

# Choose the metrics that we want to output in the table
lyrics_sum3ds_esinw <- lyrics_sum_esinw %>%
  mutate(Text = as.character(Text),
         TTR = round(Types/Tokens,3)) %>%
  rename(Language = Text)

# output the table
knitr::kable(lyrics_sum3ds_esinw)
```

```
# set up the ploting region
par(mfrow = c(2,3))
# plot each of the three metrics from the first table and then each from the second preserving o
rder
plot(x = factor(lyrics_sum_cct_simp$Language),
     y = lyrics_sum_cct_simp$`Avg Types per Lyric Excerpt`,
     border = "white",
     xlab = "Language",
     ylab = "Avg Types per Lyric Excerpt",
     main = "Avg Types per Lyric Excerpt"
     ); points(x = factor(lyrics_sum_cct_simp$Language),
       y = lyrics_sum_cct_simp$`Avg Types per Lyric Excerpt`,
       col = c("red","red","black","black","black")
       ); abline(v = 2.5,lty = 3)
plot(x = factor(lyrics_sum_cct_simp$Language),
     y = lyrics_sum_cct_simp$`Avg Tokens per Lyric Excerpt`,
     border = "white",
     xlab = "Language",
     ylab = "Avg Tokens per Lyric Excerpt",
     main = "Avg Tokens per Lyric Excerpt"
     ); points(x = factor(lyrics_sum_cct_simp$Language),
       y = lyrics_sum_cct_simp$`Avg Tokens per Lyric Excerpt`,
       col = c("red","red","black","black","black")
       ); abline(v = 2.5,lty = 3)
plot(x = factor(lyrics_sum_cct_simp$Language),
     y = lyrics_sum_cct_simp$`Avg Type/Token Ratio (TTR)`,
     border = "white",
     xlab = "Language",
     ylab = "Avg Type/Token Ratio (TTR)",
     main = "Avg Type/Token Ratio (TTR)"
     ); points(x = factor(lyrics_sum_cct_simp$Language),
       y = lyrics_sum_cct_simp$`Avg Type/Token Ratio (TTR)`,
       col = c("red","red","black","black","black")
       ); abline(v = 2.5,lty = 3)
plot(x = factor(lyrics_sum3ds_esinw$Language),
     y = lyrics_sum3ds_esinw$Types,
     border = "white",
     xlab = "Language",
     ylab = "Types",
     main = "Total Types per Language"
     ); points(x = factor(lyrics_sum3ds_esinw$Language),
       y = lyrics_sum3ds_esinw$Types,
       col = c("red","red","black","black","black")
       ); abline(v = 2.5,lty = 3)
plot(x = factor(lyrics_sum3ds_esinw$Language),
     y = lyrics_sum3ds_esinw$Tokens,
     border = "white",
     xlab = "Language",
     ylab = "Tokens",
     main = "Total Tokens per Language"
     ); points(x = factor(lyrics_sum3ds_esinw$Language),
       y = lyrics_sum3ds_esinw$Tokens,
       col = c("red","red","black","black","black")
       ); abline(v = 2.5,lty = 3)
plot(x = factor(lyrics_sum3ds_esinw$Language),
     y = lyrics_sum3ds_esinw$TTR,
     border = "white",
     xlab = "Language",
     ylab = "Type/Token Ratio",
     main = "Type/Token Ratio for Language"
     ); points(x = factor(lyrics_sum3ds_esinw$Language),
       y = lyrics_sum3ds_esinw$TTR,
       col = c("red","red","black","black","black")
       ); abline(v = 2.5,lty = 3)
```