The Dutch Diminutive: Between Inflection and Derivation

Setup and Peprocessing Script

Daniil Bondarenko

2023-01-15

0. Important notice

This script is not strictly necessary for replication purposes and is bundled with the rest of the project for the sake of transparency. If you are only interested in replicating the findings on your machine, skip this script and proceed directly to "analysis.Rmd". However, if you want to trace every single data processing step undertaken in order to set up the experimental dataset for the current project, feel free to poke around!

1. Import all datasets for preprocessing and merging

The project simulates a word recognition experiment by taking reaction time values from a preexisting visual word processing corpus, namely the Dutch Lexicon Project (DLP) 2. The entire corpus is not included in the project files due to Github's size constraints. Besides, it's always nice to give full credit where it is due: without the work put in by Marc Brysbaert et al., this entire project would not have been possible. You can access the corpus, together with the accompanying article, at http://crr.ugent.be/archives/1796. You will need to download all the corpus files in order to run the code below.

```
# Variables and Items for predictor data
variables <- read_csv("../../DLP/dlp2_variables.csv", show_col_types = FALSE)
items <- read_tsv("../../DLP/dlp2_items.tsv", show_col_types = FALSE)
# Make a big table for all items and variables
data <- inner_join(items, variables, by="spelling")
# Exclude nonwords, filter for only nouns
data <- filter(data, lexicality=="W", PoS=="N")

# Trials for participant response data
t0 <- read_tsv("../../DLP/dlp2_trials0.tsv", show_col_types = FALSE)
t1 <- read_tsv("../../DLP/dlp2_trials1.tsv", show_col_types = FALSE)
trials <- rbind(t0, t1)
# Filter the trials block to exclude repeating trial blocks (0-1 = 60-61)
# Remove the first two blocks for RT modelling to account for learning effects
trials <- filter(trials, block >= 2)
```

2. Set up the diminutive file for value assignment

```
# Set up a value for diminutive type
data$dim_type = "none"
# Assign "undecided" to any wordform ending with "-je"
```

```
data[grepl(".+je$", data$spelling),]$dim_type <- "undecided"

# Use the "undecided" value to filter for only diminutives
diminutives <- filter(data, dim_type=="undecided") # 364 in total
# Set up a few values for manual assignment
diminutives$nmorph <- 0
diminutives$double_checked <- 0
diminutives$dec_criterion <- "unassigned"
# Select only the relevant values to keep the assignment CSV readable
diminutives <- select(diminutives, c(spelling,
                                     dim_type,
                                     nmorph,
                                     double_checked,
                                     dec_criterion))
# Write the CSV for manual assignment
write_csv(diminutives, file="../corpus/dims_unass.csv")
```

3. Diminutive assignment procedure

The project is rooted within the framework of Distributed Morphology and based on the theory of there existing two heads that diminutive morphemes can attach to: Size and Lex. As such, the group of diminutive morphemes is posited to be split along the lines of where each attaches (or is "spelled out" even):

- Inflectional Diminutive: fully productive and compositional, e.g. X + Dim = "small X"; merges above the category head with the category phrase and realises SizeP; responsible for the kind-to-item reading shift.
- Derivational Diminutive: lexical gaps and non-compositional meaning, e.g. X + Dim = Y (new denotation); merges below the category head with the root and realizes LexP; possibly phonologically irregular allomorph selection.

3.1. Guidelines used for value assignment

Each item was assigned a value for the variable "dim_type" and a value for "dec_criterion", alongside a straightforward morpheme count. The guidelines of assignment closely followed the distinguishing features of the two diminutives: an item was assigned "infl" for "Inflectional" if a) its meaning was compositional in nature, usually reflecting the "small X" pattern, or if b) it forced a kind-to-item reading shift; conversely, non-compositional items (either lexicalised diminutives or diminutiva tantum) were assigned "deriv" for "Derivational". An additional test for the derivational nature of the suffixed item was to check for the item exhibiting the predicted suffixal allomorph (see Phonological Allomorphy Generalisations below): thus, bloem-pje "small flower" features the predicted allomorph -pje and is treated as inflected, bloem-etje has a different allomorph from the expected -pje and means "bouquet" instead, hence it being treated as derived.

The value "dec_criterion" stands for "Decisive Criterion", but can as well be read as defining the subtype of diminutive, with values "small_X", "kind_to_item", "lex_dim" and "dim_tant" reflecting the assignment criteria above. Items that had at least one of the inflectional and one of the derivational type readings at the same time, as reflected in native speaker intuitions and attested in the item entry on https://nl.wiktionary.org/, were assigned the value "both" for the variable "dim_type" and a combination of the four subtype values for the variable "dec_criterion".

3.2. Phonological Allomorphy Generalisations:

a. after short stressed vowels followed by a sonorant consonant: <-etje>
b. after final obstruent: <-je>
c. elsewhere <-pje> after /m/, <-kje> after /e/, <-tje> in all other cases

3.3. List of excluded items

- bonje "quarrel": morphological nature uncertain
- flottielje "flotilla": simplex, not diminutive
- franje "fringe": simplex, not diminutive
- kastanje "chestnut": simplex, not diminutive
- plunje "(certain) clothing": etymologically diminutive, too opaque
- rapalje "riffraff": old spelling of "rapaille", simplex, not diminutive

4. Finalise rough preprocessing

```
# Remove dim_type from the noun dataset to avoid duplicates post-merge
data <- select(data, -"dim_type")

# Import the CSV with assigned diminutive values
dim_data <- read_csv("../corpus/dims_assigned.csv", show_col_types = FALSE)

# Merge all datasets
dim_data <- inner_join(dim_data, data, by="spelling")
trialdata <- inner_join(dim_data, trials, by = "spelling")

# Clean up the trial dataset
trialdata <- distinct(trialdata)
# Make sure RT values are intact
trialdata <- filter(trialdata, complete.cases(trialdata$rtC))
# Remove all trials with low mean accuracy scores, exclude error responses too
trialdata <- filter(trialdata, acc.mean>=0.66, is_error==0)

# Filter out all single-value and irrelevant columns
trialdata <- rename(trialdata, "item"="item.x")
trialdata <- select(trialdata,
                    -"lexicality.x", # Single-value column
                    -"nobs", # Irrelevant for analysis
                    -"Acc_dlp2", # Duplicated in column "acc.mean"
                    -"RT_dlp2", # Duplicated in column "rtC_mean"
                    -"PoS", # Single-value column
                    -"list", # Irrelevant for analysis
                    -"block", # Irrelevant for analysis
                    -"subblock", # Irrelevant for analysis
                    -"trial_in_block", # Irrelevant for analysis
                    -"trial_in_subblock", # Irrelevant for analysis
                    -"trial", # Irrelevant for analysis
                    -"observation", # Irrelevant for analysis
                    -"warmup", # Single-value column
                    -"repetition", # Irrelevant for analysis
```

```
                    -"item.y", # Duplicated in column "item"
                    -"pair", # Irrelevant for analysis
                    -"lexicality.y", # Single-value column
                    -"handedness", # Irrelevant for analysis
                    -"xrb", # Irrelevant for analysis
                    -"one", # Single-value column
                    -"rb",# Irrelevant for analysis
                    -"r", # Single-value column
                    -"corr", # Single-value column
                    -"corrn", # Single-value column
                    -"is_missing", # Single-value column
                    -"is_error", # Single-value column
                    -"is_bad_item", # Not sure what this one means
                    -"is_lt_200", # Single-value column
                    -"is_gt_1999", # Single-value column
                    -"is_lt_lower", # Single-value column
                    -"is_gt_upper", # Single-value column
                    -"is_imputed", # Single-value column
                    - "double_checked" # Single-value column
)

# Write the CSV as output for the analysis
write_csv(trialdata, file="../corpus/trialdata.csv")
```

The experimental dataset setup part is now done. “analysis.Rmd” picks us where this left off.