08 CSI online typing APPENDIX: Automatic answer classification with alternative metrices

Kirsten Stark

3/30/2021

Load packages

```
rm(list = ls())

library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(stringr)
library(stringdist)
```

```
##
## Attaching package: 'stringdist'

## The following object is masked from 'package:tidyr':
##
##     extract
```

```
options( "encoding" = "UTF-8" )
set.seed(99)
```

Load data

```r
# input
input = "data_long_anonymous.csv"

# input synonym/alternative naming list
alternatives = "naming_alternatives.csv"

# load data
df <- read.csv(here::here("data", input))

# load alternatives
alternatives <- read.csv(here::here("data", "supplementary_info", alternatives),
                         sep = ";")
```

Load functions

Functions from https://github.com/KirstenStark/stringmatch_typed_naming (Stark,2021)

```r
source("automatic_preprocessing_functions.R")
```

Preprocess data, applying functions

1) Clean word ending

By deleting the last character(s) of typed words if those are space or enter keys. (Alternatively, the function also takes custom endings that should be deleted.)
As entries, the delete_ending function takes the column with the word entries and, optionally, a custom ending. We can repeat applying this function if we want to keep deleting if Enter or space is repeated several times at the end of the word. The while loops stops as soon as none of the words has a space or Enter (or custom ending) at the end. (In our case, this changes only the ending of three words)

```r
isnotequal <- 1
df$word.c = currentupdate = df$word
while (isnotequal > 0) {
  df <- df %>% mutate(word.c = delete_ending(df$word.c))
  isnotequal <- sum(currentupdate != df$word.c, na.rm = TRUE)
  currentupdate <- df$word.c
}
```

2) Replace special characters

Special characters such as Enter and Backspace are written as entire words. We want to replace these with identifiable numbers.

```r
oldnames <- c("Enter", "CapsLock", "Shift", "ArrowLeft", "ArrowRight", "Backspace", "Control")
newnames <- c("1", "2", "3", "4", "5", "6", "7")
df$word.c <- replace_special_chars(input = df$word.c, oldnames = oldnames, newnames = newnames)
```

```
## [1] "The pattern Enter has been replaced by the pattern 1."
## [1] "The pattern CapsLock has been replaced by the pattern 2."
## [1] "The pattern Shift has been replaced by the pattern 3."
## [1] "The pattern ArrowLeft has been replaced by the pattern 4."
## [1] "The pattern ArrowRight has been replaced by the pattern 5."
## [1] "The pattern Backspace has been replaced by the pattern 6."
## [1] "The pattern Control has been replaced by the pattern 7."
```

3) Compute finally submitted words by applying all backspaces

Function takes as input the word entries and, optionally, the backspace identifier.

```
df$word.c <- replace_backspace(df$word.c, backspace = "6")
```

4) Compute stringdist between word entries and items/alternatives

a) Compute Jaro distance, the metrics we used for the main analyses

```
tictoc::tic()
output <- calculate_stringdist(word = df$word.c, stims = df$item,
                               alternatives = alternatives,
                               method = "jw", p = 0,
                               firstlettercorrect = TRUE)
tictoc::toc()
```

```
## 1.079 sec elapsed
```

```
df$jaro <- output[,1]
df$bestmatch_jaro <- output[,2]
```

b) Compute Jaro-Winkler distance with p = 0.1

```
tictoc::tic()
output <- calculate_stringdist(word = df$word.c, stims = df$item,
                               alternatives = alternatives,
                               method = "jw", p = 0.1,
                               firstlettercorrect = TRUE)
tictoc::toc()
```

```
## 1.046 sec elapsed
```

```
df$jw <- output[,1]
df$bestmatch_jw <- output[,2]
```

c) Compute Levenshtein distance with equally weighted operations

```
tictoc::tic()
output <- calculate_stringdist(word = df$word.c, stims = df$item,
                               alternatives = alternatives,
                               method = "lv", weight = c(1,1,1),
                               firstlettercorrect = TRUE)
df$lv <- output[,1]
df$bestmatch_lv <- output[,2]
tictoc::toc()
```

```
## 1.09 sec elapsed
```

d) Compute optimal string alignment (restricted Damerau Levenshtein distance) with equally weighted operations

```
tictoc::tic()
output <- calculate_stringdist(word = df$word.c, stims = df$item,
                               alternatives = alternatives,
                               method = "osa", weight = c(1,1,1,1),
                               firstlettercorrect = TRUE)
df$osa <- output[,1]
df$bestmatch_osa <- output[,2]
tictoc::toc()
```

```
## 1.068 sec elapsed
```

e) Compute distance based on Bi-gram (Jaccard)

```
tictoc::tic()
output <- calculate_stringdist(word = df$word.c, stims = df$item,
                               alternatives = alternatives,
                               method = "jaccard", q = 2,
                               firstlettercorrect = TRUE)
df$jaccard <- output[,1]
df$bestmatch_jaccard <- output[,2]
tictoc::toc()
```

```
## 1.034 sec elapsed
```

5) Classify word entries using different answercodes

```
df <- df %>%
  mutate(answer_auto_jaro = case_character_type(word, item,
          word.c, jaro, bestmatch_jaro, d = 0.3)) %>%
  mutate(answer_auto_jw = case_character_type(word, item,
          word.c, jw, bestmatch_jw, d = 0.3)) %>%
  mutate(answer_auto_lv = case_character_type(word, item,
          word.c, osa, bestmatch_lv, d = 3)) %>%
  mutate(answer_auto_osa = case_character_type(word, item,
          word.c, osa, bestmatch_osa, d = 4)) %>%
  mutate(answer_auto_jaccard = case_character_type(word, item,
          word.c, jaccard, bestmatch_jaccard, d = 0.8))

# The different classifications are
levels(as.factor(df$answer_auto_jaro))
```

```
##  [1] "alternative_corrected" "approx_alternative"    "approx_correct"
##  [4] "backspace_space_enter" "correct"               "correctedtocorrect"
##  [7] "distance_based_error"  "first_letter_error"    "isna"
## [10] "not_correct"           "shift_start"
```

Classify answers as correct or incorrect based on the answercodes.
correct = 1, incorrect = 0

```r
df <- df %>%
  mutate(correct_auto_jaro = case_when(
    answer_auto_jaro == "correct" ~ 1,
    answer_auto_jaro == "correctedtocorrect" ~ 1,
    answer_auto_jaro == "approx_correct" ~ 1,
    answer_auto_jaro == "alternative" ~ 1,
    answer_auto_jaro == "alternative_corrected" ~ 1,
    answer_auto_jaro == "approx_alternative" ~ 1,
    TRUE ~ 0)) %>%
  mutate(correct_auto_jw = case_when(
    answer_auto_jw == "correct" ~ 1,
    answer_auto_jw == "correctedtocorrect" ~ 1,
    answer_auto_jw == "approx_correct" ~ 1,
    answer_auto_jw == "alternative" ~ 1,
    answer_auto_jw == "alternative_corrected" ~ 1,
    answer_auto_jw == "approx_alternative" ~ 1,
    TRUE ~ 0)) %>%
  mutate(correct_auto_lv = case_when(
    answer_auto_lv == "correct" ~ 1,
    answer_auto_lv == "correctedtocorrect" ~ 1,
    answer_auto_lv == "approx_correct" ~ 1,
    answer_auto_lv == "alternative" ~ 1,
    answer_auto_lv == "alternative_corrected" ~ 1,
    answer_auto_lv == "approx_alternative" ~ 1,
    TRUE ~ 0)) %>%
  mutate(correct_auto_osa = case_when(
    answer_auto_osa == "correct" ~ 1,
    answer_auto_osa == "correctedtocorrect" ~ 1,
    answer_auto_osa == "approx_correct" ~ 1,
    answer_auto_osa == "alternative" ~ 1,
    answer_auto_osa == "alternative_corrected" ~ 1,
    answer_auto_osa == "approx_alternative" ~ 1,
    TRUE ~ 0)) %>%
  mutate(correct_auto_jaccard = case_when(
    answer_auto_jaccard == "correct" ~ 1,
    answer_auto_jaccard == "correctedtocorrect" ~ 1,
    answer_auto_jaccard == "approx_correct" ~ 1,
    answer_auto_jaccard == "alternative" ~ 1,
    answer_auto_jaccard == "alternative_corrected" ~ 1,
    answer_auto_jaccard == "approx_alternative" ~ 1,
    TRUE ~ 0)) %>%
  mutate(correct_manual = case_when(correct == 1 ~ 1,
                                    is.na(correct) ~ 0))
```

Inspect results

Create data frame

```
overview <- data.frame(name = rep(NA, times=5), correlation=rep(NA, times=5),
                       newcorrect =rep(NA, times=5),
                        newcorrect_partialname=rep(NA, times=5),newcorrect_orthosim=rep(NA, times=5),
                        newcorrect_losely_related=rep(NA, times=5),
                       newincorrect = rep(NA, times=5),
                        newincorrect_firstletter_backspace=rep(NA, times=5),
                        newincorrect_phon_firstletter=rep(NA, times=5),
                        newincorrect_distance_based=rep(NA, times=5),
                        newincorrect_other=rep(NA, times=5))
overview$name <- c("Jaro", "Jaro-Winkler", "Levenshtein", "Optimal string alignment", "Bi-Gram (Jaccard
```

Jaro distance Correlation with manual classification

```
(cor_jaro <- cor.test(df$correct_manual, df$correct_auto_jaro))
```

```
##
##  Pearson’s product-moment correlation
##
## data:  df$correct_manual and df$correct_auto_jaro
## t = 269.55, df = 4798, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9667295 0.9702372
## sample estimates:
##       cor
## 0.9685314
```

```
overview$correlation[overview$name=="Jaro"]<- cor_jaro$estimate
```

“New correct” classifications: partialname: when they typed only parts of the picture name orthosim: when they typed orthographically similar word losely_related: losely_related, non-accepted alternative with orthographical similarities

```
(new_correct <- df %>%
  filter(correct_manual == 0 &
           correct_auto_jaro == 1) %>%
  dplyr::select(item, word, word.c, bestmatch_jaro,answer_auto_jaro, answercode))
```

```
##             item
## 1        flasche
## 2          kelle
## 3          geige
## 4      schublade
## 5         u-boot
## 6 geschirrspüler
## 7    daunenweste
## 8 kaffeemaschine
##                                                                                                      W
## 1                                                                                              GLASE
## 2                                                                                            KESSELE
## 3 GITARRBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceGE
## 4                                                                                             SCHUBE
## 5
## 6                                                                                              GESCH
## 7                                                                                       DAUNENJACKEE
## 8                                                                                            KAFFEEE
##        word.c bestmatch_jaro   answer_auto_jaro             answercode
## 1        GLAS    GLASFLASCHE approx_alternative      semantic_relation
## 2      KESSEL          kelle     approx_correct        unrelated_other
## 3       GEIGE          geige correctedtocorrect      semantic_relation
## 4       SCHUB      schublade     approx_correct        unrelated_other
## 5           U         u-boot     approx_correct        unrelated_other
## 6    GESCHIRR geschirrspüler     approx_correct      semantic_relation
## 7 DAUNENJACKE    daunenweste     approx_correct first_letter_incorrect
## 8      KAFFEE   KAFFEEKOCHER approx_alternative      semantic_relation
```

```
overview$newcorrect[overview$name=="Jaro"]<- nrow(new_correct)
overview$newcorrect_partialname[overview$name=="Jaro"]<- 5
overview$newcorrect_orthosim[overview$name=="Jaro"]<- 2
overview$newcorrect_losely_related[overview$name=="Jaro"]<- 1
```

“New incorrect” classifications: Firstletter_backspace: when participants backspace-corrected an accepted alternative, changing the first character of the word entry
Phon_firstletter:when they misspelled the beginning of a word with a phonologically similar phoneme
Distance-based: Distance greater cut-off

```
(new_incorrect <- df %>%
  filter(correct_manual == 1 &
           correct_auto_jaro == 0) %>%
  dplyr::select(item, word, word.c, bestmatch_jaro,answer_auto_jaro, answercode))
```

```
##           item
## 1  schornstein
## 2  daunenweste
## 3   pelzmantel
## 4    goldfisch
## 5   pelzmantel
## 6        feile
## 7        feile
## 8        couch
## 9    goldfisch
## 10        burg
## 11  pelzmantel
## 12     schloss
## 13       feile
## 14  luftballon
## 15   zigarette
## 16      kuchen
## 17   goldfisch
## 18       feile
## 19       feile
## 20        fuss
## 21      glocke
```

```
##
## 1
## 2                                                               WESTEBackspaceBackspaceBackspaceBacks
## 3
## 4                                                          FISCBackspaceBackspaceBackspaceBackspaceG
## 5                                                          MANTBackspaceBackspaceBackspaceBackspaceE
## 6
## 7
## 8                                                                                              SOFBa
## 9
## 10                                                                                              SCHB
## 11 MANTELBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBa
## 12                                                                                           BURBacks
## 13
## 14                                                                                       BALLBackspa
## 15                                                                               ZOIGBackspaceBackspa
## 16
## 17
## 18
## 19
## 20                                                 FPBackspaceUDeadDeadBackspaceBackspace?=Backspace
## 21
##            word.c bestmatch_jaro     answer_auto_jaro    answercode
## 1     SCHORNSTEIN    schornstein   first_letter_error almostcorrect
## 2     DAUNENWESTE    daunenweste   first_letter_error almostcorrect
## 3      PELZMANTEL     pelzmantel   first_letter_error almostcorrect
## 4          6FISCH          FISCH   first_letter_error almostcorrect
## 5      PELZMANTEL     pelzmantel   first_letter_error almostcorrect
## 6         PFEILER          feile   first_letter_error almostcorrect
## 7           FEILE          feile   first_letter_error almostcorrect
## 8          6COUCH          couch   first_letter_error almostcorrect
## 9      6GOLDFISCH      goldfisch   first_letter_error almostcorrect
## 10          6BURG           burg   first_letter_error almostcorrect
## 11           PELZ           PELZ   first_letter_error almostcorrect
## 12       6SCHLOSS        schloss   first_letter_error almostcorrect
## 13         PFEILE          feile   first_letter_error almostcorrect
## 14     LUFTBALLON     luftballon   first_letter_error almostcorrect
## 15    6ZIGARETTE      zigarette   first_letter_error almostcorrect
## 16          TORTE          TORTE   first_letter_error almostcorrect
## 17      GOLDFISCH      goldfisch   first_letter_error almostcorrect
## 18         PFEILE          feile   first_letter_error almostcorrect
## 19         PFEILE          feile   first_letter_error almostcorrect
## 20 FUDeadDeDeadSS            FUß distance_based_error almostcorrect
## 21         CLOCKE         glocke   first_letter_error almostcorrect
```

```
overview$newincorrect[overview$name=="Jaro"]<- nrow(new_incorrect)
overview$newincorrect_firstletter_backspace[overview$name=="Jaro"]<- 13
overview$newincorrect_phon_firstletter[overview$name=="Jaro"]<- 6
overview$newincorrect_distance_based[overview$name=="Jaro"]<- 1
overview$newincorrect_other[overview$name=="Jaro"]<- 1
```

Jaro-Winkler distance Correlation with manual classification

```
(cor_jw <- cor.test(df$correct_manual, df$correct_auto_jw))
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$correct_manual and df$correct_auto_jw
## t = 243.14, df = 4798, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9595515 0.9638021
## sample estimates:
##       cor
## 0.9617346
```

```
overview$correlation[overview$name=="Jaro-Winkler"]<- cor_jw$estimate
```

"New correct" classifications: partialname: when they typed only parts of the picture name orthosim: when they typed orthographically similar word losely_related: losely_related, non-accepted alternative with orthographical similarities

```
(new_correct <- df %>%
  filter(correct_manual == 0 &
           correct_auto_jw == 1) %>%
  dplyr::select(item, word, word.c, bestmatch_jw,answer_auto_jw, answercode))
```

```
##              item
## 1         flasche
## 2          hocker
## 3           kelle
## 4          hocker
## 5           geige
## 6          hocker
## 7          hocker
## 8       schublade
## 9          u-boot
## 10         hocker
## 11 geschirrspüler
## 12    daunenweste
## 13 kaffeemaschine
## 14         hocker
##
## 1                                                                                               GLAS
## 2                                                                                              STUHL
## 3                                                                                             KESSEL
## 4                                                                                                   S
## 5  GITARRBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceG
## 6                                                                                                   S
## 7                                                                                                   s
## 8                                                                                              SCHUB
## 9
## 10                                                                                                  S
## 11                                                                                              GESC
## 12                                                                                         DAUNENJACKE
## 13                                                                                              KAFFEE
## 14                                                                                               STUHL
##         word.c   bestmatch_jw     answer_auto_jw             answercode
## 1         GLAS    GLASFLASCHE approx_alternative      semantic_relation
## 2        STUHL        SCHEMEL approx_alternative      semantic_relation
## 3       KESSEL          kelle     approx_correct        unrelated_other
## 4        STUHL        SCHEMEL approx_alternative      semantic_relation
## 5        GEIGE          geige correctedtocorrect      semantic_relation
## 6        STUHL        SCHEMEL approx_alternative      semantic_relation
## 7        stuhl        SCHEMEL approx_alternative      semantic_relation
## 8        SCHUB      schublade     approx_correct        unrelated_other
## 9            U         u-boot     approx_correct        unrelated_other
## 10       STUHL        SCHEMEL approx_alternative      semantic_relation
## 11    GESCHIRR geschirrspüler     approx_correct      semantic_relation
## 12 DAUNENJACKE    daunenweste     approx_correct first_letter_incorrect
## 13      KAFFEE   KAFFEEKOCHER approx_alternative      semantic_relation
## 14       STUHL        SCHEMEL approx_alternative      semantic_relation
```

```
overview$newcorrect[overview$name=="Jaro-Winkler"]<- nrow(new_correct)
overview$newcorrect_partialname[overview$name=="Jaro-Winkler"]<- 5
overview$newcorrect_orthosim[overview$name=="Jaro-Winkler"]<- 2
overview$newcorrect_losely_related[overview$name=="Jaro-Winkler"]<- 6
```

"New incorrect" classifications: Firstletter_backspace: when participants backspace-corrected an accepted alternative, changing the first character of the word entry
Phon_firstletter:when they misspelled the beginning of a word with a phonologically similar phoneme
Distance-based: Distance greater cut-off

```
(new_incorrect <- df %>%
  filter(correct_manual == 1 &
           correct_auto_jw == 0) %>%
  dplyr::select(item, word, word.c, bestmatch_jw,answer_auto_jw, answercode))
```

```
##           item
## 1  schornstein
## 2  daunenweste
## 3   pelzmantel
## 4    goldfisch
## 5   pelzmantel
## 6        feile
## 7        feile
## 8        couch
## 9    goldfisch
## 10        burg
## 11  pelzmantel
## 12     schloss
## 13       feile
## 14  luftballon
## 15   zigarette
## 16      kuchen
## 17   goldfisch
## 18       feile
## 19       feile
## 20        fuss
## 21      glocke
##
## 1
## 2                                                                WESTEBackspaceBackspaceBackspaceBacks
## 3
## 4                                                           FISCBackspaceBackspaceBackspaceBackspaceG
## 5                                                           MANTBackspaceBackspaceBackspaceBackspaceE
## 6
## 7
## 8                                                                                               SOFBa
## 9
## 10                                                                                              SCHBa
## 11 MANTELBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBa
## 12                                                                                           BURBacks
## 13
## 14                                                                                       BALLBackspa
## 15                                                                               ZOIGBackspaceBackspa
## 16
## 17
## 18
## 19
## 20                                                 FPBackspaceUDeadDeadBackspaceBackspace?=Backspace
## 21
##            word.c bestmatch_jw       answer_auto_jw    answercode
## 1     SCHORNSTEIN  schornstein   first_letter_error almostcorrect
## 2     DAUNENWESTE  daunenweste   first_letter_error almostcorrect
## 3      PELZMANTEL   pelzmantel   first_letter_error almostcorrect
## 4          6FISCH        FISCH   first_letter_error almostcorrect
## 5      PELZMANTEL   pelzmantel   first_letter_error almostcorrect
## 6         PFEILER        feile   first_letter_error almostcorrect
## 7           FEILE        feile   first_letter_error almostcorrect
## 8          6COUCH        couch   first_letter_error almostcorrect
## 9      6GOLDFISCH    goldfisch   first_letter_error almostcorrect
## 10          6BURG         burg   first_letter_error almostcorrect
## 11           PELZ         PELZ   first_letter_error almostcorrect
## 12       6SCHLOSS      schloss   first_letter_error almostcorrect
## 13         PFEILE        feile   first_letter_error almostcorrect
## 14     LUFTBALLON   luftballon   first_letter_error almostcorrect
## 15     6ZIGARETTE    zigarette   first_letter_error almostcorrect
## 16          TORTE        TORTE   first_letter_error almostcorrect
## 17      GOLDFISCH    goldfisch   first_letter_error almostcorrect
## 18         PFEILE        feile   first_letter_error almostcorrect
## 19         PFEILE        feile   first_letter_error almostcorrect
## 20 FUDeadDeDeadSS          FUß distance_based_error almostcorrect
## 21         CLOCKE       glocke   first_letter_error almostcorrect
```

```
overview$newincorrect[overview$name=="Jaro-Winkler"]<- nrow(new_incorrect)
overview$newincorrect_firstletter_backspace[overview$name=="Jaro-Winkler"]<- 14
overview$newincorrect_phon_firstletter[overview$name=="Jaro-Winkler"]<- 6
overview$newincorrect_distance_based[overview$name=="Jaro-Winkler"]<- 1
overview$newincorrect_other[overview$name=="Jaro-Winkler"]<- 0
```

Levenshtein distance Correlation with manual classification

```
(cor_lv <- cor.test(df$correct_manual, df$correct_auto_lv))
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$correct_manual and df$correct_auto_lv
## t = 282.67, df = 4798, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9696153 0.9728230
## sample estimates:
##       cor
## 0.9712632
```

```
overview$correlation[overview$name=="Levenshtein"]<- cor_lv$estimate
```

"New correct" classifications: partialname: when they typed only parts of the picture name orthosim: when they typed orthographically similar word losely_related: losely_related, non-accepted alternative with orthographical similarities

```
(new_correct <- df %>%
  filter(correct_manual == 0 &
           correct_auto_lv == 1) %>%
  dplyr::select(item, word, word.c, bestmatch_lv,answer_auto_lv, answercode))
```

```
##    item
## 1 geige
##                                                                                                      w
## 1 GITARRBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceGE
##   word.c bestmatch_lv     answer_auto_lv        answercode
## 1  GEIGE        geige correctedtocorrect semantic_relation
```

```
overview$newcorrect[overview$name=="Levenshtein"]<- nrow(new_correct)
overview$newcorrect_partialname[overview$name=="Levenshtein"]<- 0
overview$newcorrect_orthosim[overview$name=="Levenshtein"]<- 0
overview$newcorrect_losely_related[overview$name=="Levenshtein"]<- 1
```

"New incorrect" classifications: Firstletter_backspace: when participants backspace-corrected an accepted alternative, changing the first character of the word entry
Phon_firstletter:when they misspelled the beginning of a word with a phonologically similar phoneme
Distance-based: Distance greater cut-off

```
(new_incorrect <- df %>%
  filter(correct_manual == 1 &
           correct_auto_lv == 0) %>%
  dplyr::select(item, word, word.c, bestmatch_lv,answer_auto_lv, answercode))
```

```
##              item
## 1     schornstein
## 2     daunenweste
## 3      pelzmantel
## 4       goldfisch
## 5      pelzmantel
## 6           feile
## 7           feile
## 8  geschirrspüler
## 9          bürste
## 10          couch
## 11      goldfisch
## 12           burg
## 13     pelzmantel
## 14        schloss
## 15          feile
## 16     luftballon
## 17    lippenstift
## 18      zigarette
## 19        teppich
## 20         kuchen
## 21            bär
## 22      goldfisch
## 23          feile
## 24          feile
## 25 kaffeemaschine
## 26         glocke
##
## 1
## 2                                                               WESTEBackspaceBackspaceBackspaceBacks
## 3
## 4                                                          FISCBackspaceBackspaceBackspaceBackspaceG
## 5                                                          MANTBackspaceBackspaceBackspaceBackspaceE
## 6
## 7
## 8                                                                                      GESCHIRRWASCBa
## 9
## 10                                                                                             SOFBac
## 11
## 12                                                                                              SCHBa
## 13 MANTELBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBa
## 14                                                                                           BURBacks
## 15
## 16                                                                                       BALLBackspac
## 17                                                                                          LIPPENTI
## 18                                                                               ZOIGBackspaceBackspa
## 19                                                                                     TPPICHArrowLe
## 20
## 21                                                               BÄREnterEnter BackspaceEnterBackspac
## 22
## 23
## 24
## 25                                                              KAFEMASCHINEArrowLeftArrowLeftArrowLe
## 26
##                  word.c         bestmatch_lv       answer_auto_lv    answercode
## 1           SCHORNSTEIN          schornstein   first_letter_error almostcorrect
## 2          DAUNENWESTE          daunenweste   first_letter_error almostcorrect
## 3           PELZMANTEL           pelzmantel   first_letter_error almostcorrect
## 4               6FISCH                FISCH   first_letter_error almostcorrect
## 5           PELZMANTEL           pelzmantel   first_letter_error almostcorrect
## 6              PFEILER                feile   first_letter_error almostcorrect
## 7                FEILE                feile   first_letter_error almostcorrect
## 8     GESCHIRRMASCHINE GESCHIRRSPÜLMASCHINE distance_based_error almostcorrect
## 9        BBÜRSTEBÜRSTE           HAARBÜRSTE          not_correct almostcorrect
## 10              6COUCH                couch   first_letter_error almostcorrect
## 11          6GOLDFISCH            goldfisch   first_letter_error almostcorrect
## 12               6BURG                 burg   first_letter_error almostcorrect
## 13                PELZ                 PELZ   first_letter_error almostcorrect
## 14            6SCHLOSS              schloss   first_letter_error almostcorrect
## 15              PFEILE                feile   first_letter_error almostcorrect
## 16          LUFTBALLON           luftballon   first_letter_error almostcorrect
## 17     LIPPENTIFT4444S          lippenstift distance_based_error almostcorrect
## 18          6ZIGARETTE            zigarette   first_letter_error almostcorrect
## 19        TPPICH44444E              teppich distance_based_error almostcorrect
## 20               TORTE                TORTE   first_letter_error almostcorrect
## 21              BÄRBÄR                  bär distance_based_error almostcorrect
## 22           GOLDFISCH            goldfisch   first_letter_error almostcorrect
## 23              PFEILE                feile   first_letter_error almostcorrect
## 24              PFEILE                feile   first_letter_error almostcorrect
## 25 KAFEMASCHINE4444444       kaffeemaschine distance_based_error almostcorrect
## 26              CLOCKE               glocke   first_letter_error almostcorrect
```

```
overview$newincorrect[overview$name=="Levenshtein"]<- nrow(new_incorrect)
overview$newincorrect_firstletter_backspace[overview$name=="Levenshtein"]<- 14
overview$newincorrect_phon_firstletter[overview$name=="Levenshtein"]<- 6
overview$newincorrect_distance_based[overview$name=="Levenshtein"]<- 6
overview$newincorrect_other[overview$name=="Levenshtein"]<- 0
```

Optimal string alignment (restricted Damereau-Levenshtein) Correlation with manual classification

```
(cor_osa <- cor.test(df$correct_manual, df$correct_auto_osa))
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$correct_manual and df$correct_auto_osa
## t = 282.14, df = 4798, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9695069 0.9727258
## sample estimates:
##       cor
## 0.9711606
```

```
overview$correlation[overview$name=="Optimal string alignment"]<- cor_osa$estimate
```

"New correct" classifications: partialname: when they typed only parts of the picture name orthosim: when they typed orthographically similar word losely_related: losely_related, non-accepted alternative with orthographical similarities

```
(new_correct <- df %>%
  filter(correct_manual == 0 &
           correct_auto_osa == 1) %>%
  dplyr::select(item, word, word.c, bestmatch_osa,answer_auto_osa, answercode))
```

```
##    item
## 1 kelle
## 2 geige
##                                                                                                      w
## 1                                                                                             KESSELE
## 2 GITARRBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceGE
##   word.c bestmatch_osa    answer_auto_osa        answercode
## 1 KESSEL         kelle     approx_correct   unrelated_other
## 2  GEIGE         geige correctedtocorrect semantic_relation
```

```
overview$newcorrect[overview$name=="Optimal string alignment"]<- nrow(new_correct)
overview$newcorrect_partialname[overview$name=="Optimal string alignment"]<- 0
overview$newcorrect_orthosim[overview$name=="Optimal string alignment"]<- 1
overview$newcorrect_losely_related[overview$name=="Optimal string alignment"]<- 1
```

"New incorrect" classifications: Firstletter_backspace: when participants backspace-corrected an accepted alternative, changing the first character of the word entry
Phon_firstletter:when they misspelled the beginning of a word with a phonologically similar phoneme
Distance-based: Distance greater cut-off

```
(new_incorrect <- df %>%
  filter(correct_manual == 1 &
           correct_auto_osa == 0) %>%
  dplyr::select(item, word, word.c, bestmatch_osa,answer_auto_osa, answercode))
```

```
##              item
## 1     schornstein
## 2     daunenweste
## 3      pelzmantel
## 4       goldfisch
## 5      pelzmantel
## 6           feile
## 7           feile
## 8  geschirrspüler
## 9          bürste
## 10          couch
## 11      goldfisch
## 12           burg
## 13     pelzmantel
## 14        schloss
## 15          feile
## 16     luftballon
## 17    lippenstift
## 18      zigarette
## 19        teppich
## 20         kuchen
## 21      goldfisch
## 22          feile
## 23          feile
## 24 kaffeemaschine
## 25         glocke
##
## 1
## 2                                                               WESTEBackspaceBackspaceBackspaceBacks
## 3
## 4                                                          FISCBackspaceBackspaceBackspaceBackspaceG
## 5                                                          MANTBackspaceBackspaceBackspaceBackspaceE
## 6
## 7
## 8                                                                                      GESCHIRRWASCBa
## 9
## 10                                                                                             SOFBac
## 11
## 12                                                                                              SCHBa
## 13 MANTELBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBa
## 14                                                                                           BURBacks
## 15
## 16                                                                                       BALLBackspac
## 17                                                                                          LIPPENTI
## 18                                                                               ZOIGBackspaceBackspa
## 19                                                                                     TPPICHArrowLe
## 20
## 21
## 22
## 23
## 24                                                              KAFEMASCHINEArrowLeftArrowLeftArrowL
## 25
##                  word.c        bestmatch_osa      answer_auto_osa    answercode
## 1          SCHORNSTEIN          schornstein   first_letter_error almostcorrect
## 2          DAUNENWESTE          daunenweste   first_letter_error almostcorrect
## 3           PELZMANTEL           pelzmantel   first_letter_error almostcorrect
## 4               6FISCH                FISCH   first_letter_error almostcorrect
## 5           PELZMANTEL           pelzmantel   first_letter_error almostcorrect
## 6              PFEILER                feile   first_letter_error almostcorrect
## 7                FEILE                feile   first_letter_error almostcorrect
## 8     GESCHIRRMASCHINE GESCHIRRSPÜLMASCHINE distance_based_error almostcorrect
## 9        BBÜRSTEBÜRSTE           HAARBÜRSTE          not_correct almostcorrect
## 10              6COUCH                couch   first_letter_error almostcorrect
## 11          6GOLDFISCH            goldfisch   first_letter_error almostcorrect
## 12               6BURG                 burg   first_letter_error almostcorrect
## 13                PELZ                 PELZ   first_letter_error almostcorrect
## 14            6SCHLOSS              schloss   first_letter_error almostcorrect
## 15              PFEILE                feile   first_letter_error almostcorrect
## 16          LUFTBALLON           luftballon   first_letter_error almostcorrect
## 17     LIPPENTIFT4444S          lippenstift distance_based_error almostcorrect
## 18          6ZIGARETTE            zigarette   first_letter_error almostcorrect
## 19        TPPICH44444E              teppich distance_based_error almostcorrect
## 20               TORTE                TORTE   first_letter_error almostcorrect
## 21           GOLDFISCH            goldfisch   first_letter_error almostcorrect
## 22              PFEILE                feile   first_letter_error almostcorrect
## 23              PFEILE                feile   first_letter_error almostcorrect
## 24 KAFEMASCHINE4444444       kaffeemaschine distance_based_error almostcorrect
## 25              CLOCKE               glocke   first_letter_error almostcorrect
```

```
overview$newincorrect[overview$name=="Optimal string alignment"]<- nrow(new_incorrect)
overview$newincorrect_firstletter_backspace[overview$name=="Optimal string alignment"]<- 14
overview$newincorrect_phon_firstletter[overview$name=="Optimal string alignment"]<- 6
overview$newincorrect_distance_based[overview$name=="Optimal string alignment"]<- 5
overview$newincorrect_other[overview$name=="Optimal string alignment"]<- 0
```

Jaccard Bi-gram frequency Correlation with manual classification

```
(cor_jaccard <- cor.test(df$correct_manual, df$correct_auto_jaccard))
```

```
##
##  Pearson's product-moment correlation
##
## data:  df$correct_manual and df$correct_auto_jaccard
## t = 249.08, df = 4798, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9613508 0.9654156
## sample estimates:
##       cor
## 0.9634386
```

```
overview$correlation[overview$name=="Bi-Gram (Jaccard)"]<- cor_jaccard$estimate
```

"New correct" classifications: partialname: when they typed only parts of the picture name orthosim: when they typed orthographically similar word losely_related: losely_related, non-accepted alternative with orthographical similarities

```
(new_correct_jaccard <- df %>%
  filter(correct_manual == 0 &
           correct_auto_jaccard == 1) %>%
  dplyr::select(item, word, word.c, bestmatch_jaccard,answer_auto_jaccard, answercode))
```

```
##             item
## 1        flasche
## 2          kelle
## 3          geige
## 4      schublade
## 5 geschirrspüler
## 6    daunenweste
## 7 kaffeemaschine
##                                                                                                      w
## 1                                                                                              GLASEr
## 2                                                                                            KESSELEr
## 3 GITARRBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceGE
## 4                                                                                             SCHUBE
## 5                                                                                              GESCH
## 6                                                                                       DAUNENJACKEE
## 7                                                                                            KAFFEEE
##         word.c bestmatch_jaccard answer_auto_jaccard             answercode
## 1         GLAS       GLASFLASCHE  approx_alternative      semantic_relation
## 2       KESSEL             kelle      approx_correct        unrelated_other
## 3        GEIGE             geige  correctedtocorrect      semantic_relation
## 4        SCHUB         schublade      approx_correct        unrelated_other
## 5     GESCHIRR    geschirrspüler      approx_correct      semantic_relation
## 6 DAUNENJACKE       daunenweste      approx_correct first_letter_incorrect
## 7       KAFFEE      KAFFEEKOCHER  approx_alternative      semantic_relation
```

```
overview$newcorrect[overview$name=="Bi-Gram (Jaccard)"]<- nrow(new_correct)
overview$newcorrect_partialname[overview$name=="Bi-Gram (Jaccard)"]<- 4
overview$newcorrect_orthosim[overview$name=="Bi-Gram (Jaccard)"]<- 2
overview$newcorrect_losely_related[overview$name=="Bi-Gram (Jaccard)"]<- 1
```

"New incorrect" classifications: Firstletter_backspace: when participants backspace-corrected an accepted alternative, changing the first character of the word entry
Phon_firstletter:when they misspelled the beginning of a word with a phonologically similar phoneme
Distance-based: Distance greater cut-off

```
(new_incorrect <- df %>%
  filter(correct_manual == 1 &
           correct_auto_jaccard == 0) %>%
  dplyr::select(item, word, word.c, bestmatch_jaccard,answer_auto_jaccard, answercode))
```

```
##           item
## 1  schornstein
## 2  daunenweste
## 3   pelzmantel
## 4    goldfisch
## 5   pelzmantel
## 6        feile
## 7        feile
## 8          hai
## 9         safe
## 10       couch
## 11   goldfisch
## 12        burg
## 13  pelzmantel
## 14     schloss
## 15       feile
## 16   bleistift
## 17        glas
## 18     brosche
## 19  luftballon
## 20   zigarette
## 21      kuchen
## 22   goldfisch
## 23       feile
## 24      u-boot
## 25       feile
## 26        fuss
## 27      glocke
##
## 1
## 2                                                                   WESTEBackspaceBackspaceBackspaceBacks
## 3
## 4                                                              FISCBackspaceBackspaceBackspaceBackspaceG
## 5                                                              MANTBackspaceBackspaceBackspaceBackspaceE
## 6
## 7
## 8
## 9
## 10                                                                                                  SOFBac
## 11
## 12                                                                                                   SCHBa
## 13 MANTELBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBackspaceBa
## 14                                                                                                BURBacks
## 15
## 16
## 17
## 18
## 19                                                                                            BALLBackspac
## 20                                                                                    ZOIGBackspaceBackspa
## 21
## 22
## 23
## 24
## 25
## 26                                                      FPBackspaceUDeadDeadBackspaceBackspace?=BackspaceB
## 27
##           word.c bestmatch_jaccard  answer_auto_jaccard    answercode
## 1     SCHORNSTEIN       schornstein   first_letter_error almostcorrect
## 2     DAUNENWESTE       daunenweste   first_letter_error almostcorrect
## 3      PELZMANTEL        pelzmantel   first_letter_error almostcorrect
## 4          6FISCH             FISCH   first_letter_error almostcorrect
## 5      PELZMANTEL        pelzmantel   first_letter_error almostcorrect
## 6         PFEILER             feile   first_letter_error almostcorrect
## 7           FEILE             feile   first_letter_error almostcorrect
## 8             HEI               hai distance_based_error almostcorrect
## 9            SAGE              safe distance_based_error almostcorrect
## 10         6COUCH             couch   first_letter_error almostcorrect
## 11     6GOLDFISCH         goldfisch   first_letter_error almostcorrect
## 12          6BURG              burg   first_letter_error almostcorrect
## 13           PELZ              PELZ   first_letter_error almostcorrect
## 14       6SCHLOSS           schloss   first_letter_error almostcorrect
## 15         PFEILE             feile   first_letter_error almostcorrect
## 16      BELISTIFT             STIFT   first_letter_error almostcorrect
## 17           gals              glas distance_based_error almostcorrect
## 18    JU7ASCHMUCK           SCHMUCK   first_letter_error almostcorrect
## 19     LUFTBALLON        luftballon   first_letter_error almostcorrect
## 20     6ZIGARETTE        zigarette   first_letter_error almostcorrect
## 21          TORTE             TORTE   first_letter_error almostcorrect
## 22      GOLDFISCH         goldfisch   first_letter_error almostcorrect
## 23         PFEILE             feile   first_letter_error almostcorrect
## 24          UBOOT              BOOT   first_letter_error almostcorrect
## 25         PFEILE             feile   first_letter_error almostcorrect
## 26 FUDeadDeDeadSS              fuss distance_based_error almostcorrect
## 27         CLOCKE            glocke   first_letter_error almostcorrect
```

```
overview$newincorrect[overview$name=="Bi-Gram (Jaccard)"]<- nrow(new_incorrect)
overview$newincorrect_firstletter_backspace[overview$name=="Bi-Gram (Jaccard)"]<- 15
overview$newincorrect_phon_firstletter[overview$name=="Bi-Gram (Jaccard)"]<- 6
overview$newincorrect_distance_based[overview$name=="Bi-Gram (Jaccard)"]<- 4
overview$newincorrect_other[overview$name=="Bi-Gram (Jaccard)"]<- 2
```

```


overview
```

Display overview table

```
##                        name correlation newcorrect newcorrect_partialname
## 1                      Jaro   0.9685314          8                      5
## 2              Jaro-Winkler   0.9617346         14                      5
## 3               Levenshtein   0.9712632          1                      0
## 4 Optimal string alignment   0.9711606          2                      0
## 5         Bi-Gram (Jaccard)   0.9634386          2                      4
##   newcorrect_orthosim newcorrect_losely_related newincorrect
## 1                   2                         1           21
## 2                   2                         6           21
## 3                   0                         1           26
## 4                   1                         1           25
## 5                   2                         1           27
##   newincorrect_firstletter_backspace newincorrect_phon_firstletter
## 1                                 13                             6
## 2                                 14                             6
## 3                                 14                             6
## 4                                 14                             6
## 5                                 15                             6
##   newincorrect_distance_based newincorrect_other
## 1                           1                  1
## 2                           1                  0
## 3                           6                  0
## 4                           5                  0
## 5                           4                  2
```