Second Language Acquisition: Exploring Common Mistakes

Denis Kapelyushnik*

Abstract

In 2018, a challenge on Second Language Acquisition Modeling was organised by Duolingo AI in conjunction with the 13th BEA Workshop and NAACL-HLT 2018 conference. One of the key findings of the challenge was the fact that a choice of a learning algorithm (for the task) appears to be more important than clever feature engineering. This research paper for the Linguistic Data: Quantitative Analysis and Visualisation course is aimed to explore if any connection between certain available features and mistakes made while acquiring a foreign language exists.

1. Metadata

The dataset used for this paper comes from B. Settles et al. (2018). 7M words produced by more than 6k learners of English, Spanish, and French using Duolingo, an online language-learning app, were collected for the Second Language Acquisition Modeling (SLAM) task. The more detailed task description and results achieved by contestants are available on the official task page[1].

The original data is organized into language pairs: `es_en` — Spanish learners (who already speak English), `fr_en` — French learners (who already speak English), `en_es` English learners (who already speak SPanish). This project is focused on French learners only.

Only `train` splits prepared by Burr Settles (2018) were used in this project. A dataset per language pair was split into two files[2]: `fr_en_metadata.csv` and `fr_en_sessions.csv`.

Both files contain data separated by tabs (no headers):

Table 1: Content of the *_metada.csv files

Column name	Description
user_id	generated during data anonimisation
country	a 2-character country code
days	day of usage (a double)
client	android, ios, or web
session	lesson, practice, or test
format	reverse_translate, reverse_tap, or listen
time	duration of the answer in seconds
session_id	use it to join metadata and sessions
n_tokens	a number of tokens used in the task
n_errors	a number of mistakes a user made
prompt	prompt (no prompt in listening)

*HSE University, dmkapelyushnik@edu.hse.ru, https://github.com/deniskapel/SLAM

[1] http://sharedtask.duolingo.com/2018

[2] To reproduce this paper, follow the instructions specified in the data folder of the project github repository: https://github.com/deniskapel/SLAM/tree/main/data

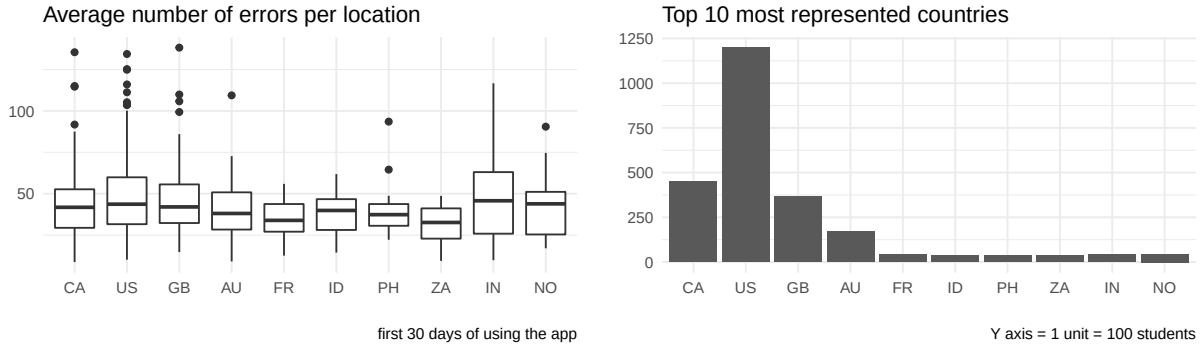
Table 2: Content of the *_sessions.csv files

Column name	Description
session_id	unique ID for a sesssion
task_token_id	location of a token in a task
token	word
POS	part of speech in UD format
morph	morphological features in UD format
ud_edge_label	dependency edge label in UD format
ud_edge_head	dependency edge head in UD format
label	to be predicted (0 or 1): 0 - correct, 1 - wrong

2. Describing the data

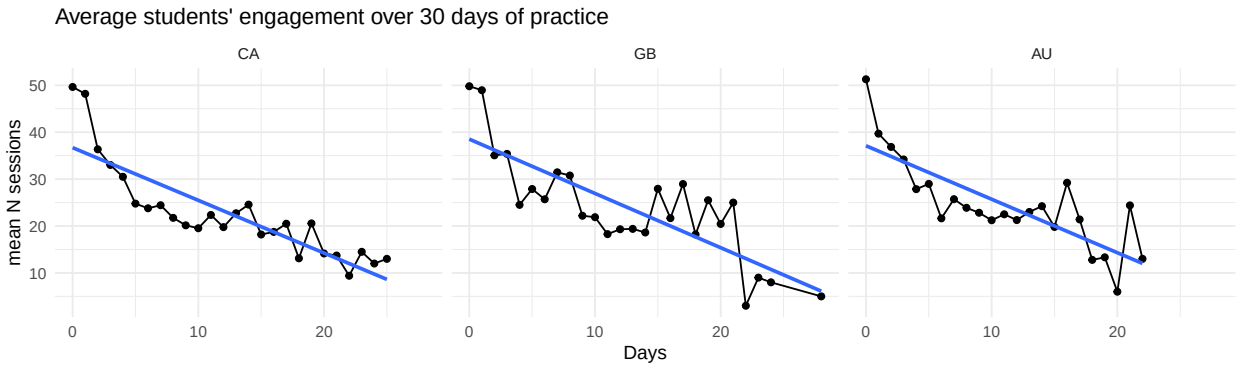
2.1. Countries and users

Overall, there are more than 100 locations where people use the app and the number of users in these countries can differ significantly.

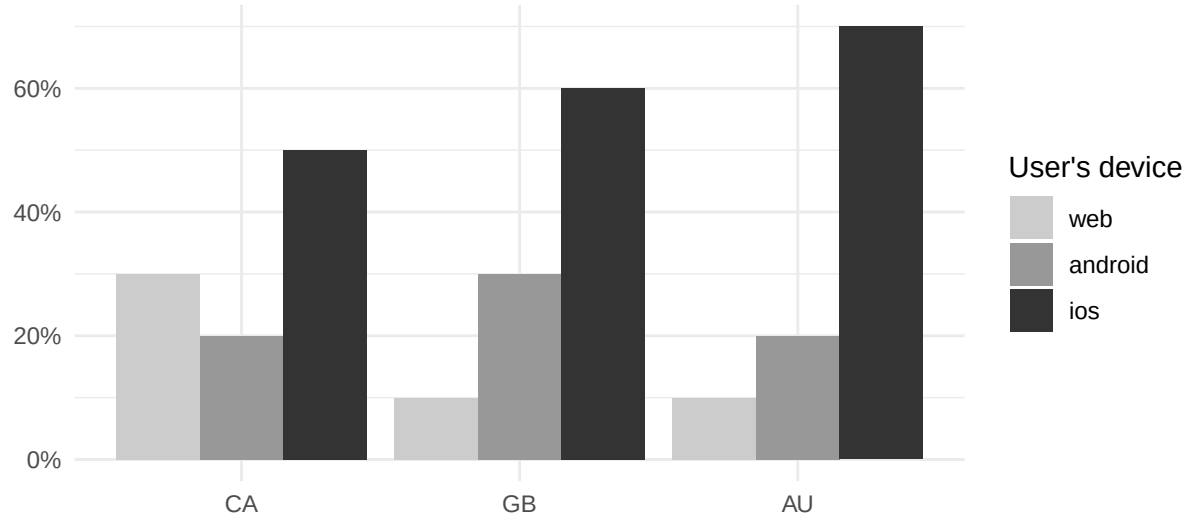


The left side of the graph above demonstrates similarity in a number of mistakes per user (with their mean slightly below 50) in 10 most represented countries for a 30-day period. Based on this, it was decided to limit the data for the project to three countries from Top 4 most represented countries to eliminate any additional factors (e.g. L1) that might have influence on second language acquisition (SLA). Users from `Canada`, `Great Britain` and `Australia` are assumed to be native speakers of English. Additionally, it will be interesting to check if Canada' bilingual status has any influence on SLA. `USA` is removed from the project, mainly to save on computational resources - there are almost as many users from this country as from other three altogether.

Allegedly, all the users are beginners who are taking first steps in acquiring L2. Mostly, they start using the app actively but their engagement decreases over time.

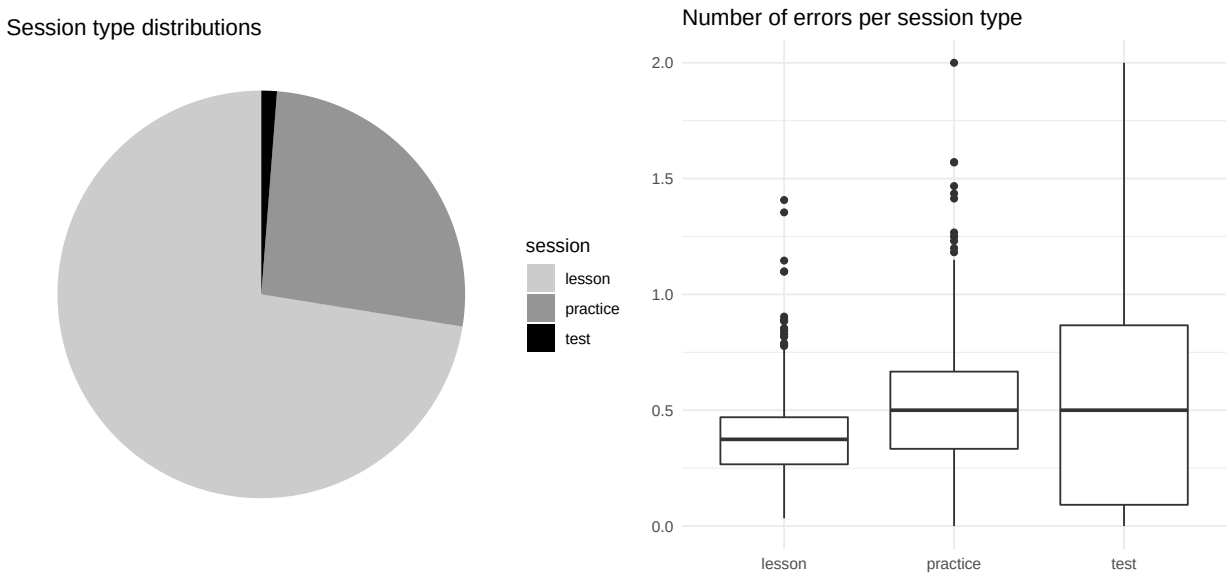


As for users' social status, `client` (users' devices) is the only feature that might be used to describe it (quite indirectly, though). In general, all the users come from high-income countries, and there is no obvious reason to start learning French there (except for `Canada`).



2.2. Types of sessions

There are session types in the dataset: `lesson`, `practice` and `test`. The `lesson` sessions are where new words or concepts are introduced, although lessons also include a lot of previously-learned material (e.g., each exercise tries to introduce only one new word or tense, so all other tokens should have been seen by the student before). The `practice` sessions should contain only previously-seen words and concepts. The `test` sessions allow a student "skip" a particular skill unit of the curriculum (i.e., the student may have never seen this content before in the Duolingo app, but may well have had prior knowledge before starting the course).



It seems that learners are more careful when they see a new word or some unknown grammatical concept (`lesson`) than in situations when all the content is familiar to them (`practise`). In `test` sessions, a wider range in number of errors can be explained that both "experienced" and regular learners can take these tasks. In Section 3, I will concentrate on `practice` sessions to model users' mistakes in a "familiar" background.

2.3. Tasks and common mistakes

The app provides users with three different task formats: `listen` (listen and type a phrase in L2), `reverse_tap` (input L2 tokens in a correct order to translate a phrase) and `reverse_translate` (read and translate a phrase into L2). Only `listen` and `reverse_translate` tasks require typing, hence learners are more prone to make mistakes while taking them[3].

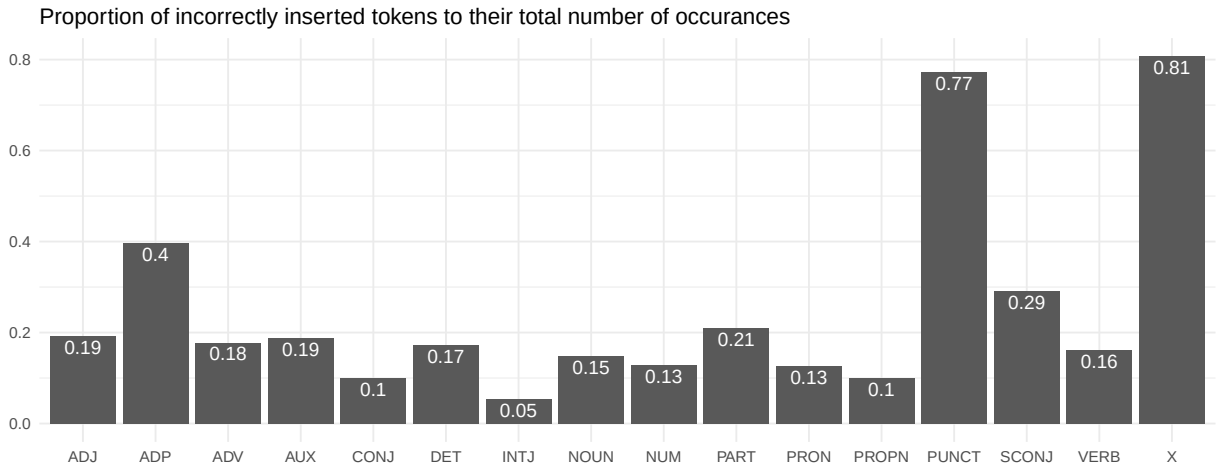
Table 3: Average number of mistake users make per task

Task Type	Value
reverse_translate	0.66
listen	0.60
reverse_tap	0.20

A task can contain from 1 to 14 tokens (depending on the language). Each token has a set of features assigned to it by B. Settles et al. (2018) using the Google SyntaxNet dependency parser and the language-agnostic Universal Dependencies tagset[4].

[3]Indeed, minimal edit distance is used to handle mistyping but it depends on a token, e.g. *you* will not be accepted for *your* even if edit distance is only 1

[4]Parse errors may occur.



Top 3 most "erroneous" tags are `PUNCT`, `X`, `ADP`. The first UD tag refers to `-`, and this character is used in such questions as `Qui sont-ils?` or `Qu'a-t-il?`. The second question includes `t` character as well, which was tagged as `X`. Apparently, both cases, i.e. `PUNCT` and `X`, refer to word order issues as these mistakes happen a lot of the times in `reverse_tap` and `reverse_translate` tasks more often than in the others. Here, students have to input L2 sentences based on L1 prompts. In `listen` tasks, students can compare their input to the correct audio-prompt in L2.

Table 4: Distribution of Errors Tagged as PUNCT by Task Format

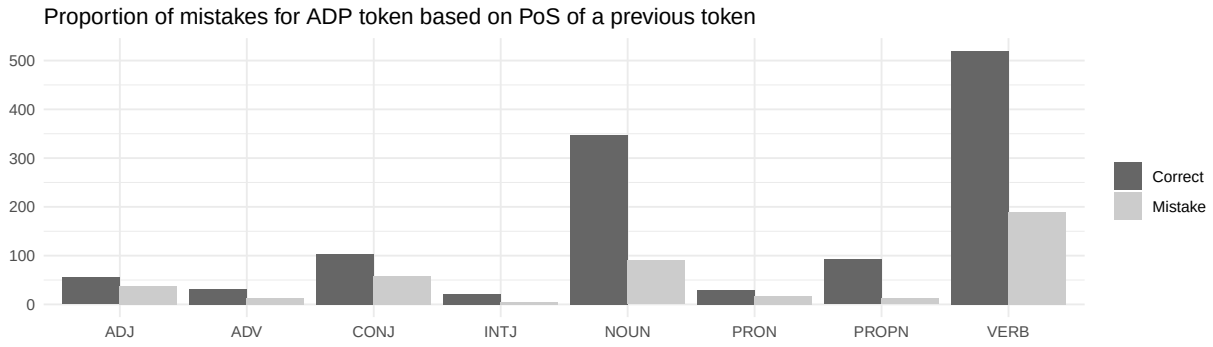
Format	Number of Errors
reverse_translate	487
listen	167
reverse_tap	486

Based on this assumption, there is nothing else to learn about the nature of mistakes with first two "parts of speech". Some additional exploration may be performed for the third one, though. For example, the most "erroneous" word tagged as `ADP` is `de`. There are three variants of this preposition's spelling in Top 5 most common mistakes.

Table 5: Most common mistakes for the ADP tag

Token	D'	de	à	De	en	d'	avec	Dans	Tu	À	comme
Quantity	1086	194	149	70	56	50	47	35	33	20	20

It is quite unexpected, that `De` and `D'` are in the top of the list. It might refer to such phrases as `D'accord` (OK) and `De rien` (Not at all). The problem is that, without seeing actual users' input, it is difficult to understand if a wrong word was used "deliberately" or the users submitted their answer by accident. To avoid any bias, it was decided to explore only the prepositions `de`, `à`, `avec`, `en`, `comme` in more details.



The graph above compares the number of correctly and incorrectly inserted prepositions if they are preceded by a certain part of speech. While exploring previous tokens , I found certain annotation errors that would impact SLAM if these morphological features were used as one of the variables.

The example of these annotation errors may be seen if a preposition is preceded by a token with an `ADJ` tag.

Table 6: Examples of sentences which contain a preposition after an adjective

1	2	3	4	5	6
Elle	est	francaise	de	naissance	
Difficile	a	dire			
C’	est	difficile	de	choisir	

Tag `ADJ` has the following feature distribution. Below are the example of prepositions preceded by `ADJ`.

Table 7: Feature distribution of ADJ

	3	ADJ++	Fem	Fin	Ind	Masc	Plur	Pres	Sing	VERB++
fPOS	0	39	0	0	0	0	0	0	0	53
Gender	0	0	1	0	0	38	0	0	0	0
Mood	0	0	0	0	53	0	0	0	0	0
Number	0	0	0	0	0	0	1	0	91	0
Person	53	0	0	0	0	0	0	0	0	0
Tense	0	0	0	0	0	0	0	53	0	0
VerbForm	0	0	0	53	0	0	0	0	0	0

There is only one binary feature with a large number of examples that we can use, for example, in a Chi-squared test: `fPOS`. The problem is that it is a a `fake` adjective feature. Its values are `ADJ++` and `VERB++`. The second group are the verbs, indeed. In fact, it is only one verb - `manger` (to eat) - in its `3^rd person singular` form.

1	2	3	4	5	6	7	8
Votre	grand	-	pere	mange	de	la	soupe
La	fille	mange	de	la	soupe		
La	fille	mange	de	la	soupe		
L’	homme	mange	de	la	viande		

As there are more `fake` adjectives than `real` ones, and such proportion of mistakes might cause problems for modeling. I decided not to use morphological features for Second Language Acquisition Modeling performed in Chapter 3.

3. Second Language Acqusition Modelling

As it was stated in Chapter 2.2, I am reducing the dataset to the `practice` session format to make sure learners are already familiar with all the vocabulary and grammatical concepts.

A quick summary of the features:

1. On average, there are much more mistakes in `reverse_translate` and `listen` tasks than in `reverse_tap`.
2. Users have more problems with some tokens than with the others (though, it is not always clear why).
3. Previous and following tokens might be an extra feature but using UD tags seems to be unreliable.

A few additional ideas that might be tested as factors for formulas in mixed-effect models:

- task taken in the last decated of a 30-day period indicate that a user is commited to learn and might make less mistakes.
- It is easier to do some tasks (e.g. `reverse_tap`) using mobile platforms than browsers.
- It is easier to learn languages for some users than for the others.

To test for mix-effects models, I will use generalised mixed-effects modelling function from `lme4` package and a join of sessions' metadata and features of each token.

Table 9: SLAM features

Column name	Description
session_id	uniqe session id
user_id	generated during data anonimisation
country	a 2-character country code
days	day of usage (a double)
client	android, ios, or web
session	lesson, practice, or test
format	reverse_translate, reverse_tap, or listen
time	duration of the answer in seconds
n_tokens	a number of tokens used in the task
task_token_id	location of a token in a task
token	token itself or the middle word in a trigram
previous_token	the first word in a trigram
following_token	the last word in the trigram
label	to be predicted (0 or 1): 0 - correct, 1 - wrong

In order to save on computational resources, I take only 1% of the data saving the share of `correct` and `incorrect` entries.

I will begin with a few generalized linear mixed effects models assuming there is a random effect from a token `+ (1|token)`, a user `+ (1|user_id)`, a previous token `+ (1|previous)` or the following one `+ (1|following)`, `+ (1|country)`. First, I will test formulas that include numerical variables only: `n_tokens`, `days` and `task_token_id`.

- `Models 0-4`: label ~ n_tokens * days * task_token_id: the idea here is that the longer the sentence is, the more possibilities for mistake there are + further down the process of acquisition it is, the more committed the user is. If a token comes first, some mistakes happen by accident.

```
lmer0 <- glmer(
  label~days*n_tokens*task_token_id + (1|token),
  data=sample_df, family = binomial)

lmer1 <- glmer(
  label~days*n_tokens*task_token_id + (1|user_id),
  data=sample_df, family = binomial)

lmer2 <- glmer(
  label~days*n_tokens*task_token_id + (1|previous),
  data=sample_df, family = binomial)

lmer3 <- glmer(
  label~days*n_tokens*task_token_id + (1|following),
  data=sample_df, family = binomial)

lmer4 <- glmer(
  label~days*n_tokens*task_token_id + (1|country),
  data=sample_df, family = binomial)
```

All of these models fail to converge as they are too complex. Further tests will use more simple models.

```
lmer4 <- glmer(
  label~days + (1|user_id),
  data=sample_df, family = binomial)

lmer5 <- glmer(
  label~days + (1|token),
  data=sample_df, family = binomial)

lmer6 <- glmer(
  label~days + (1|country),
  data=sample_df, family = binomial)
```

```
## Data: sample_df
## Models:
## lmer4: label ~ days + (1 | user_id)
## lmer5: label ~ days + (1 | token)
## lmer6: label ~ days + (1 | country)
##       npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## lmer4    3 820.35 834.38 -407.17   814.35
## lmer5    3 788.29 802.31 -391.14   782.29 32.062  0
## lmer6    3 821.74 835.77 -407.87   815.74  0.000  0
```

All the models are very close and inefficient. Some improvement comes from a random effect by token itself but not enough.

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
```

```
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: label ~ days + (1 | token)
##    Data: sample_df
##
##      AIC      BIC   logLik deviance df.resid
##    788.3    802.3   -391.1    782.3      790
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.9681 -0.4575 -0.3657 -0.2783  3.4893
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  token  (Intercept) 1.278    1.131
## Number of obs: 793, groups:  token, 252
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.21787    0.21160  -5.756 8.63e-09 ***
## days        -0.02116    0.01939  -1.091    0.275
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## days -0.738
```

```
lmer7 <- glmer(
  label~n_tokens + (1|token),
  data=sample_df, family = binomial)

lmer8 <- glmer(
  label~task_token_id + (1|token),
  data=sample_df, family = binomial)

anova(lmer5, lmer7, lmer8)
```

```
## Data: sample_df
## Models:
## lmer5: label ~ days + (1 | token)
## lmer7: label ~ n_tokens + (1 | token)
## lmer8: label ~ task_token_id + (1 | token)
##       npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## lmer5    3 788.29 802.31 -391.14   782.29
## lmer7    3 786.22 800.25 -390.11   780.22 2.0685  0
## lmer8    3 787.87 801.90 -390.93   781.87 0.0000  0
```

Same thing for other fixed factor, still inefficient. It is now time to start testing categorical data with the same set of random effects.

```
lmer9 <- glmer(
  label~format + format:client + (1|token),
```

```
  data=sample_df, family = binomial)

lmer10 <- glmer(
  label~format + format:client + (1|user_id),
  data=sample_df, family = binomial)

lmer11 <- glmer(
  label~format + format:client + (1|country),
  data=sample_df, family = binomial)

anova(lmer5, lmer9, lmer10, lmer11)
```

```
## Data: sample_df
## Models:
## lmer5: label ~ days + (1 | token)
## lmer9: label ~ format + format:client + (1 | token)
## lmer10: label ~ format + format:client + (1 | user_id)
## lmer11: label ~ format + format:client + (1 | country)
##        npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## lmer5     3 788.29 802.31 -391.14   782.29
## lmer9     9 774.72 816.81 -378.36   756.72 25.563  6  0.0002685 ***
## lmer10    9 803.46 845.54 -392.73   785.46  0.000  0
## lmer11    9 806.54 848.62 -394.27   788.54  0.000  0
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model 9 `label ~ format + format:client + (1 | token)` looks more promising, yet it is still inefficient.

```
summary(lmer9)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: label ~ format + format:client + (1 | token)
##    Data: sample_df
##
##      AIC      BIC   logLik deviance df.resid
##    774.7    816.8   -378.4    756.7      784
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.1653 -0.4706 -0.3305 -0.1985  4.1782
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  token  (Intercept) 1.256    1.121
## Number of obs: 793, groups:  token, 252
##
## Fixed effects:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -1.2892     0.2940  -4.384 1.16e-05 ***
## formatreverse_tap                 -0.7297     0.5726  -1.274    0.203
```

```
## formatlisten                            -0.1466     0.4229  -0.347    0.729
## formatreverse_translate:clientios        0.1330     0.3519   0.378    0.705
## formatreverse_tap:clientios             -0.2765     0.5597  -0.494    0.621
## formatlisten:clientios                   0.5402     0.4167   1.296    0.195
## formatreverse_translate:clientandroid    0.4639     0.4104   1.130    0.258
## formatlisten:clientandroid               0.8552     0.6790   1.260    0.208
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##                       (Intr) frmtr_ frmtls frmtrvrs_trnslt:clnts frmtrvrs_tp:
## frmtrvrs_tp           -0.454
## formatlistn           -0.614  0.321
## frmtrvrs_trnslt:clnts -0.734  0.375  0.516
## frmtrvrs_tp:           0.009 -0.782  0.003  0.006
## frmtlstn:clnts        -0.018 -0.011 -0.575 -0.004                -0.002
## frmtrvrs_trnslt:clntn -0.650  0.326  0.449  0.542                 0.004
## frmtlstn:clntn        -0.031 -0.014 -0.357 -0.005                 0.000
##                       frmtlstn:clnts frmtrvrs_trnslt:clntn
## frmtrvrs_tp
## formatlistn
## frmtrvrs_trnslt:clnts
## frmtrvrs_tp:
## frmtlstn:clnts
## frmtrvrs_trnslt:clntn  0.004
## frmtlstn:clntn         0.365          0.003
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00936858 (tol = 0.002, component 1)
```

Probably, some improvement can be extracted from combining categorical and numerical variables.

```
lmer11 <- glmer(
  label ~ days + format + (1 | token),
  data=sample_df, family = binomial)

lmer12 <- glmer(
  label ~ days + format + (1|user_id),
  data=sample_df, family = binomial)

lmer13 <- glmer(
  label ~ days + format + (1|country),
  data=sample_df, family = binomial)

anova(lmer11, lmer12, lmer13)
```

```
## Data: sample_df
## Models:
## lmer11: label ~ days + format + (1 | token)
## lmer12: label ~ days + format + (1 | user_id)
## lmer13: label ~ days + format + (1 | country)
##        npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)
```

```
## lmer11    5 770.10 793.48 -380.05   760.10
## lmer12    5 797.16 820.54 -393.58   787.16     0  0
## lmer13    5 801.65 825.03 -395.83   791.65     0  0
```

This set of feature does not provide any imporvement as well. For now and based on Chapter 2.3, it seems that choosing `correct` when a task format is `reverse_tap` is the most promising approach. Especially if grouped by the position of a token in a sentence.

```
lmer17 <- glmer(
  label ~ format +
    (1 + task_token_id|token),
  data=sample_df, family = binomial)

lmer18 <- glmer(
  label ~ n_tokens +
    (1 + task_token_id|token),
  data=sample_df, family = binomial)

anova(lmer17, lmer18)
```

```
## Data: sample_df
## Models:
## lmer18: label ~ n_tokens + (1 + task_token_id | token)
## lmer17: label ~ format + (1 + task_token_id | token)
##        npar    AIC    BIC  logLik deviance  Chisq Df Pr(>Chisq)
## lmer18    5 789.73 813.10 -389.86   779.73
## lmer17    6 771.55 799.61 -379.78   759.55 20.172  1  7.077e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(lmer17)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: label ~ format + (1 + task_token_id | token)
##    Data: sample_df
##
##      AIC      BIC   logLik deviance df.resid
##    771.6    799.6   -379.8    759.6      787
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.0644 -0.4707 -0.3240 -0.2109  3.5066
##
## Random effects:
##  Groups Name          Variance Std.Dev. Corr
##  token  (Intercept)   1.6584   1.288
##         task_token_id 0.1971   0.444    -0.64
## Number of obs: 793, groups:  token, 252
##
## Fixed effects:
##                   Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)       -1.14177    0.18335  -6.227 4.75e-10 ***
## formatreverse_tap -1.10738    0.27143  -4.080 4.51e-05 ***
## formatlisten       0.04246    0.25055   0.169    0.865
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) frmtr_
## frmtrvrs_tp -0.441
## formatlistn -0.533  0.329
```

This feature might be useful to identify some accidental mistakes (see Chapter 2.3 on prepositions `De` and `D'`) but does not bring much to learning analytics.

Conclusion

The project was aimed to explore if any available features have a stronger effect on `mistake`/ `correct` classification. While analysing the dataset, I removed UD features due to problems with tags. After I described the features, I applied generalised mixed-effects modeling to find out if any features or their combinations can be used to predict the label. The results of the experiments did not result in any meaningful feature set, and perhaps some additional synthesised features, e.g. `ngram frequency`, might be used in further experiments. Data sampling was applied mainly to save on computational resources, and as soon as some feature set is defined, it is possible to test it on larger dataset.

Below, there are features that I attempted to test during the experiments and alternative hypotheses for them:

- `days` - (fixed) the longer users study, the more committed they are (more attentive)
- `days:n_tokens` - Tasks' difficulty gradually increases, and it is possible that shorter sentences become easier with practice.
- `format` - average number of mistakes for `reverse_tap` tasks is three times as small as for the others.
- `format:client` - some tasks might be easier to do using cellphones rather than laptops or computers.

The following random effects were added to the model as well: `by token`, `by user`, `by country`. Their combination makes models too complex.

References

Settles, B., C. Brust, E. Gustafson, M. Hagiwara, and N. Madnani. 2018. "Second Language Acquisition Modeling." In *Proceedings of the Naacl-Hlt Workshop on Innovative Use of Nlp for Building Educational Applications (Bea)*. ACL. https://doi.org/10.7910/DVN/8SWHNO.

Settles, Burr. 2018. "Data for the 2018 Duolingo Shared Task on Second Language Acquisition Modeling (SLAM)." Harvard Dataverse. https://doi.org/10.7910/DVN/8SWHNO.