Supplementary Information for *3SG is the most conservative subject marker across languages: An exploratory study of rate of change*

Peter Dekker*[1], Sonja Gipper[2], and Bart de Boer[1]

*Corresponding Author: peter.dekker@ai.vub.ac.be
[1]AI Lab, Vrije Universiteit Brussel, Brussels, Belgium
[2]Institut für Linguistik, Philosophische Fakultät, Universität zu Köln, Cologne, Germany

1. Code

All code of this paper can be found in `https://zenodo.org/doi/10.5281/zenodo.10722183` and the GitHub repository `https://github.com/peterdekker/changesubjectmarkers`.

2. Data

The structure of the data from Seržant and Moroz (2022) (data publication: Seržant, 2021, version v5), which we used for our analysis, is given in SI Table 1. Every row is one person-number entry for a certain language, which contains its modern form, proto-language and proto-form. The column `source` (not in excerpt SI Table 1) gives the source that was used for the information about this language. The original data, before removing languages during preprocessing, consists of 383 languages from 53 families. The dataset consists of about 10-50 languages per family.

3. Preprocessing

We used Python, using the `pandas` (McKinney, 2010; The pandas development team, 2020) library, for filtering and processing of the data. First, we removed all rows where either the modern form or the proto-form is NA: this means data that is missing (it does not mean a form with length 0). There was only one entry for which the modern form was NA. Removing the NA proto-forms in practice fully removes all languages where no proto-language is linked (hence there are no proto-forms). Only in one case it removes a part of the entries for a language. After removing entries with empty modern forms and proto-forms, we have 1815 entries for 310 languages, associated with 15 proto-languages.

In order to calculate the Levenshtein distance between modern and proto-forms, we perform more processing of the strings (but no more filtering). The `,` is used to split alternative full forms, whereas the `/` is used to signify alternative

Table 1.: Excerpt of the data structure of Seržant and Moroz (2022) (data publication: Seržant (2021). Shown are the first three languages, and a limited number of columns.

	language	proto_language	person_number	person	number	modern_form	proto_form	clade3
0	Lithuanian	Proto-Indo-European	1sg	first	sg	u	ō, oh2	Indo-European
1	Lithuanian	Proto-Indo-European	2sg	second	sg	i	e-s-i	Indo-European
2	Lithuanian	Proto-Indo-European	3sg	third	sg	a	e-t-i	Indo-European
3	Lithuanian	Proto-Indo-European	1pl	first	pl	ame	o-m-e/os(i)	Indo-European
4	Lithuanian	Proto-Indo-European	2pl	second	pl	ate	e-th2-e	Indo-European
5	Lithuanian	Proto-Indo-European	3pl	third	pl	a	o-nt-i	Indo-European
6	Latvian	Proto-Indo-European	1sg	first	sg	u	ō, oh2	Indo-European
7	Latvian	Proto-Indo-European	2sg	second	sg	0	e-s-i	Indo-European
8	Latvian	Proto-Indo-European	3sg	third	sg	0	e-t-i	Indo-European
9	Latvian	Proto-Indo-European	1pl	first	pl	am	o-m-e/os(i)	Indo-European
10	Latvian	Proto-Indo-European	2pl	second	pl	at	e-th2-e	Indo-European
11	Latvian	Proto-Indo-European	3pl	third	pl	0	o-nt-i	Indo-European
12	Mgreek	Proto-Indo-European	1sg	first	sg	o	ō, oh2	Indo-European
13	Mgreek	Proto-Indo-European	2sg	second	sg	is	e-s-i	Indo-European
14	Mgreek	Proto-Indo-European	3sg	third	sg	i	e-t-i	Indo-European
15	Mgreek	Proto-Indo-European	1pl	first	pl	ume	o-m-e/os(i)	Indo-European
16	Mgreek	Proto-Indo-European	2pl	second	pl	ete	e-th2-e	Indo-European
17	Mgreek	Proto-Indo-European	3pl	third	pl	un	o-nt-i	Indo-European

morphemes. We split the forms on `,` and `/`, to get all the alternative forms, and we only use the first form, as this is the most common form, also used for the precalculated lengths in the dataset. Ideally, one would take into account the variation in forms, but using multiple forms brings in new complexities, where some languages will have multiple datapoints per grammatical person, whereas others have 1. Subsequently, because the forms are not purely phonetic forms, but also dictionary or other notations, we remove all the symbols where the symbol does not directly represent a sound. We remove the morpheme marker `-`, the symbol `2`, which is part of the PIE reconstructed laryngeal h_2 in proto-forms (leaving only the `h`), the notations `0` and `∅` for an empty person marker (leaving an empty string), the `*`, signifying a reconstruction, and segments between brackets.

Also, `...`, signalling a gap in nonconcatenative morphology, is removed. The `:`, lengthening a vowel, is removed. Lastly, `´`, `'` and `#`, which are not counted in the precalculated lengths in the dataset, are removed. We kept `V`, signalling a vowel, as it represents a sound and can in some cases be compared between proto-form and modern form. The resulting form was then run through the `unidecode` method[1], a crude way to remove some diacritics from the characters, to make them more comparable.

4. Levenshtein metric

To calculate unnormalised Levenshtein distance, the modern form and proto-form (processed as described above) are compared using Levenshtein distance (Heeringa, 2004; Levenshtein, 1966), from the `editdistance`[2] package in Python. For the normalised Levenshtein distance, the unnormalised Levenshtein distance is divided by the length of the longest form (either modern or proto form),

[1]From library `unidecode`: `https://github.com/avian2/unidecode`.

[2]`https://github.com/roy-ht/editdistance`

which gives a value between 0 and 1.

5. Statistical model

Mixed linear models were implemented in the lme4 package (Bates, Mächler, Bolker, & Walker, 2015) in R, using the rpy2 wrapper[3] to run R code in Python, as we used Python for all our preprocessing.

The R formula for the model is:

```
proto_levenshtein ~ person*number + (1|clade3)
```

We use the column *clade3* in the dataset as a random effect (random intercept). *clade3* often corresponds to the highest-level language family, only in two cases, the authors of the dataset decided to split up a family, and assign the subfamilies to *clade3*: they did this for highest-level families Nuclear Trans New Guinea and Afroasiatic. In nearly all cases, *clade3* corresponds the column *proto_language*, only in Proto-Tibeto-Burman, *clade3* is more fine-grained.

From this fitted model model, predictions are made for the different grammatical persons using the ggpredict function from the ggeffects package, which serve as the basis for the predictions plots in the main article. Using the mixed function from the afex package (Singmann, Bolker, Westfall, Aust, & Ben-Shachar, 2022) we perform ANOVA likelihood ratio tests for all the fixed effects.

5.1. *Results: Unnormalised Levenshtein distance*

The mixed linear model, fitted with restricted maximum likelihood (REML) gave the following output:

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: proto_levenshtein ~ person * number + (1 | clade3)
   Data: df

REML criterion at convergence: 4989.2

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.4272 -0.6693  0.0473  0.6212  5.0027

Random effects:
 Groups   Name        Variance Std.Dev.
 clade3   (Intercept) 0.1343   0.3665
 Residual             0.8876   0.9421
Number of obs: 1814, groups:  clade3, 16
```

[3] https://rpy2.github.io

```
Fixed effects:
                         Estimate Std. Error         df t value Pr(>|t|)    
(Intercept)               1.35824    0.10755   23.62934  12.628 5.39e-12 ***
personsecond              0.44384    0.07617 1797.28213   5.827 6.67e-09 ***
personthird              -0.50024    0.07617 1797.28213  -6.568 6.67e-11 ***
numberpl                  0.36452    0.07567 1793.33331   4.817 1.58e-06 ***
personsecond:numberpl     0.01706    0.10755 1793.33331   0.159    0.874    
personthird:numberpl      0.79271    0.10880 1794.12071   7.286 4.76e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) prsnsc prsnth nmbrpl prsns:
personsecnd -0.344                            
personthird -0.344  0.497                     
numberpl    -0.352  0.497  0.497              
prsnscnd:nm  0.248 -0.706 -0.350 -0.704       
prsnthrd:nm  0.245 -0.346 -0.698 -0.696  0.489
```

Predictions, using `ggpredict`:

```
# number = sg

person | Predicted |       95% CI
---------------------------------
first  |      1.36 | [1.15, 1.57]
second |      1.80 | [1.59, 2.01]
third  |      0.86 | [0.65, 1.07]

# number = pl

person | Predicted |       95% CI
---------------------------------
first  |      1.72 | [1.51, 1.93]
second |      2.18 | [1.97, 2.40]
third  |      2.02 | [1.80, 2.23]
```

According to the ANOVA likelihood ratio tests, the fixed effects person, number and the interaction between person and number are significant:

```
Mixed Model Anova Table (Type 3 tests, LRT-method)

Model: proto_levenshtein ~ person * number + (1 | clade3)
Data: df
Df full model: 8
         Effect df      Chisq p.value
1        person  2 115.01 ***   <.001
2        number  1 194.47 ***   <.001
3 person:number  2  67.21 ***   <.001
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

5.2. *Results: Normalised Levenshtein distance*

Output of mixed linear model (restricted maximum likelihood):

```
Linear mixed model fit by REML. t-tests use Satterthwaite's method [
lmerModLmerTest]
Formula: proto_levenshtein ~ person * number + (1 | clade3)
   Data: df

REML criterion at convergence: 1251.8

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.7837 -0.6264  0.1670  0.8673  2.0389

Random effects:
 Groups   Name        Variance Std.Dev.
 clade3   (Intercept) 0.01651  0.1285
 Residual             0.11234  0.3352
Number of obs: 1814, groups:  clade3, 16

Fixed effects:
                        Estimate Std. Error         df t value Pr(>|t|)
(Intercept)              0.64892    0.03786   23.54196  17.139 8.52e-15 ***
personsecond             0.05925    0.02710 1797.17321   2.187   0.0289 *
personthird             -0.16635    0.02710 1797.17321  -6.139 1.02e-09 ***
numberpl                -0.04860    0.02692 1793.12696  -1.805   0.0712 .
personsecond:numberpl    0.02329    0.03826 1793.12696   0.609   0.5428
personthird:numberpl     0.28820    0.03871 1793.94781   7.446 1.49e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Correlation of Fixed Effects:
            (Intr) prsnsc prsnth nmbrpl prsns:
personsecnd -0.348
personthird -0.348  0.497
numberpl    -0.356  0.497  0.497
prsnscnd:nm  0.250 -0.706 -0.350 -0.704
prsnthrd:nm  0.247 -0.346 -0.698 -0.696  0.489
```

Predictions, using `ggpredict`:

```
# number = sg

person | Predicted |       95% CI
---------------------------------
first  |      0.65 | [0.57, 0.72]
second |      0.71 | [0.63, 0.78]
third  |      0.48 | [0.41, 0.56]

# number = pl

person | Predicted |       95% CI
---------------------------------
first  |      0.60 | [0.53, 0.67]
second |      0.68 | [0.61, 0.76]
third  |      0.72 | [0.65, 0.80]
```

```
Adjusted for:
* clade3 = 0 (population-level)
```

According to the ANOVA likelihood ratio tests, the fixed effects person, number and the interaction between person and number are significant:

```
Mixed Model Anova Table (Type 3 tests, LRT-method)

Model: proto_levenshtein ~ person * number + (1 | clade3)
Data: df
Df full model: 8
         Effect df     Chisq p.value
1        person  2 25.17 ***   <.001
2        number  1 12.25 ***   <.001
3 person:number  2 66.45 ***   <.001
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

References

Bates, D., Mächler, M., Bolker, B., & Walker, S. (2015). Fitting linear mixed-effects models using lme4. *Journal of Statistical Software*, *67*(1), 1–48.

Heeringa, W. J. (2004). *Measuring dialect pronunciation differences using Levenshtein distance.* Unpublished doctoral dissertation, Rijksuniv. Groningen.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady* (Vol. 10, pp. 707–710).

McKinney, W. (2010). Data Structures for Statistical Computing in Python. In S. van der Walt & Millman, Jarrod (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61).

Seržant, I. (2021). *Dataset for the paper "Universal attractors in language evolution provide evidence for the kinds of efficiency pressures involved".* https://zenodo.org/records/7641119: Zenodo.

Seržant, I. A., & Moroz, G. (2022). Universal attractors in language evolution provide evidence for the kinds of efficiency pressures involved. *Humanities and Social Sciences Communications*, *9*(1), 58.

Singmann, H., Bolker, B., Westfall, J., Aust, F., & Ben-Shachar, M. S. (2022). *Afex: Analysis of factorial experiments* [Manual].

The pandas development team. (2020). *Pandas-dev/pandas: Pandas.* Zenodo.