Ruhr-University Bochum

Sprachwissenschaftliches Institut

HS Störungen der Sprachproduktion

Wintersemester 2017/18

PLD20 – USER'S MANUAL

Helena Wedig B.A.

Matrikelnr. 108013105330

M.A. Linguistik

helena.wedig@ruhr-uni-bochum.de

Table of Contents

1. System Overview

This manual introduces a tool for producing the phonological Levenshtein distance 20 (PLD20). To examine the connection between the phonological similarity and the lexical access in language production, psycholinguists need to calculate the PLD20 of selected tokens and by doing so get the 20 words with the lowest Levenshtein distance and the mean of all of their distances in regard to a target word. The developed tool enables the user to calculate the phonological Levenshtein distance of any number of words and extract the 20 nearest words of either SUBTLEX or a corpus of choice. The given corpus contains a selection of German words whose graphemes were converted into phonemes via G2P. The Input can either be a list of words or a file containing a list of words that needs to be compared to a target word, a file containing two columns with a pair of words per line which need to be compared with each other or a file that contains a list of words and their phonological Levenshtein distance related to the target word. Apart from processing files the developed graphical user interface enables the user to type in the target and the list manually.

The system developed by the linguistics department of the Ruhr-University Bochum will be freely accessible by any scientist in need of a tool to calculate or use the PLD20.

In the first instance, it was released as an offline tool, but an online version is now available, so the user does not need to install python or the required modules anymore.

This manual will introduce the requirements to use the tool offline and the main functions. A "Getting Started" section explains in short how to install the tool, how to start it and the requirements to use it. The following section describes how to use the system and presents examples. A reporting section explains possible error messages, a “Using the Website” section shows the Usage of the website and an expandability section describes possible future uses.

2. Getting Started

This section will provide a general walkthrough of the system from system configuration through exit. First it will explain which modules need to be installed and how to download them to your system. The next subsection explains how to start the tool and how to use the system menu. The following subsection will introduce the main functions and the handling of the graphical user interface. Finally, the user will get to know how to exit the system.

2.1 System Configuration

To use the tool, python 3.X is required. If the tool is used on Mac or Linux installing is possible via homebrew by typing "brew install python" in your command line or via the download of python from https://www.python.org/downloads/mac-osx for the Mac distribution or from https://www.python.org/downloads/source for the Linux version. The windows distribution is available at the following website https://www.python.org/downloads/windows.

Furthermore, some modules are required. To download these the use of the command line is recommended. Following modules need to be installed:

- xml.etree.ElementTree
- requests
- PyQt5

To install these modules on Mac or Linux, either homebrew or pip via the command line and the command "brew install XXX" or pip3 install XXX" can be used. On windows the command "python -m pip install XXX" will install the required modules.

After ensuring the right environment, the folder "PLD20" and its content has to be put in a location of choice. While using the tool a working internet connection is mandatory.

2.2 Logging On + System Menu

To start the tool the systems direction has to be changed via the command line to the location of the file “PLD20.py” (cd XXX/PLD20). The tool then starts by using the command "python3 PLD20.py". After starting the tool, a window opens as shown in the following figure 1:

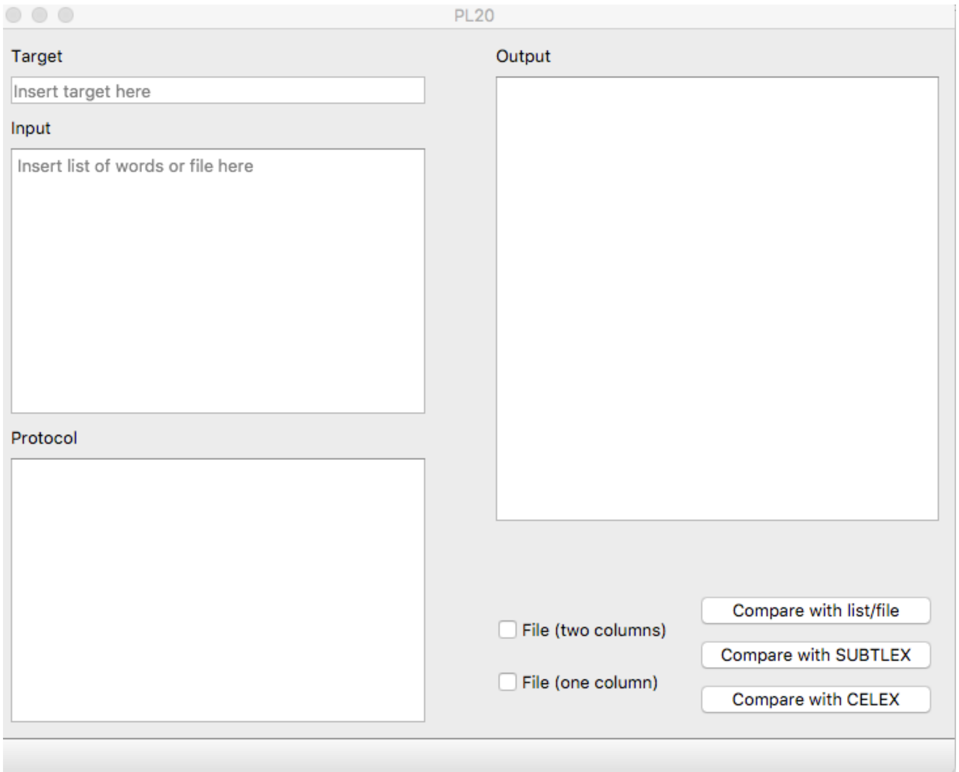


Figure 1. Program window.

2.3 System Functions

This subsection will describe the functions and attributes of the graphical user interface. The actual use of functions will be focused in the section "Using the System".

2.3.1 Target

As shown in the figure 2, the GUI contains a text field named "Target" in the upper left corner of the window. The target word has to be put in here. The stated word will be compared to the given input and be the basis for the calculation of the Levenshtein distance. As the PLD20 will be calculated based on this input, it needs to be a word or fragment containing only alphabetic characters.

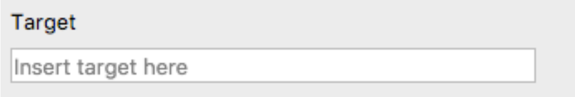


Figure 2. Text field to type in target word.

2.3.2 Input

The text field beneath is named "Input" and is found on the mid left side of the window (figure 3). The data can be inserted by either drag and drop a file that needs to be processed or typing in a list of words. The data will be compared to the target in regards of the Levenshtein distance.

When using a list of words, only one word per line has to be typed in.

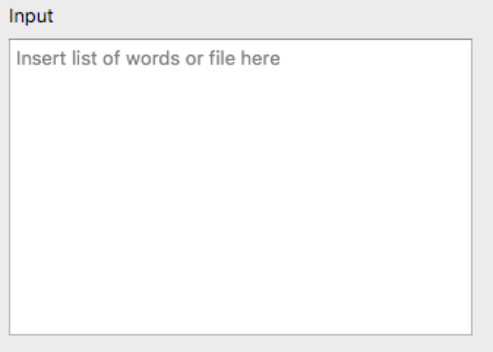


Figure 3. Text field to paste in file or type in list of words

2.3.3 Protocol

The text field "Protocol" is located in the lower left section of the window (figure 4). It is not possible to drop files or write text in here, as it only is used for the protocol. This field lists the steps of the ongoing process and the input of the user. Likewise, error messages are displayed in this text field, if the used data is not appropriate. A detailed list of the possible error messages can be found in the section "Reporting".

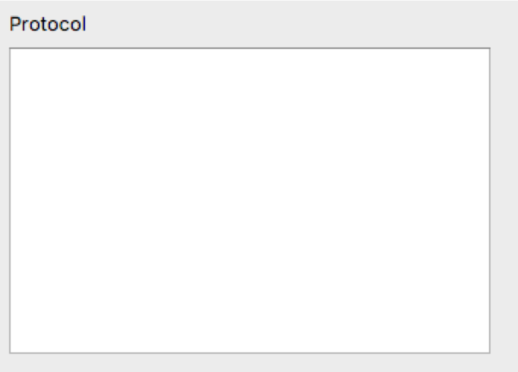


Figure 4. Text field showing the process

2.3.4 Output

After calculating the Levenshtein distance, the results can be seen on the upper right side of the window (figure 5). It is not possible to drop files or write text in here, as it only is used for the selected output. This manual contains an overview of the output in the several subsections of "Using the System".

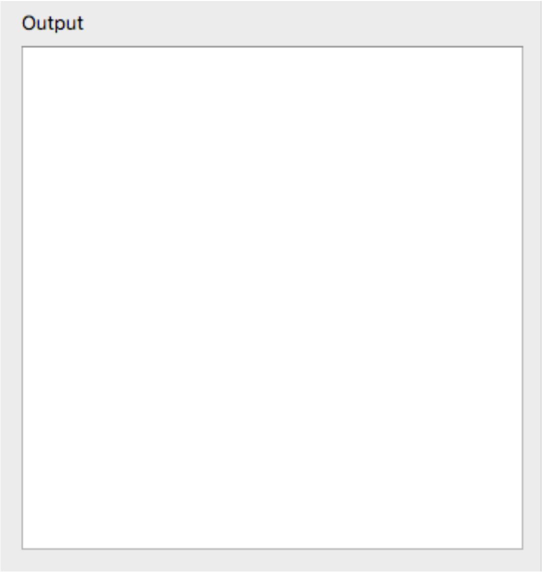


Figure 5. Text field containing the results

2.3.5 Using the Buttons and Checkboxes

To use the tool and start the process, clicking on one of the three Buttons on the right side of the window is necessary (figure 6). If files were dropped as the input, the user can either check "File (two columns)" or "File (one columns)" depending on the given format.

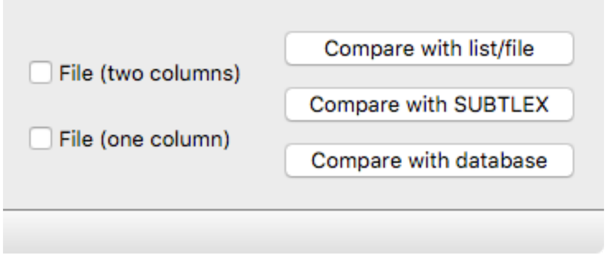


Figure 6. Buttons and check boxes to start the process

2.4 Exit System

To exit the system the user can just close the window.

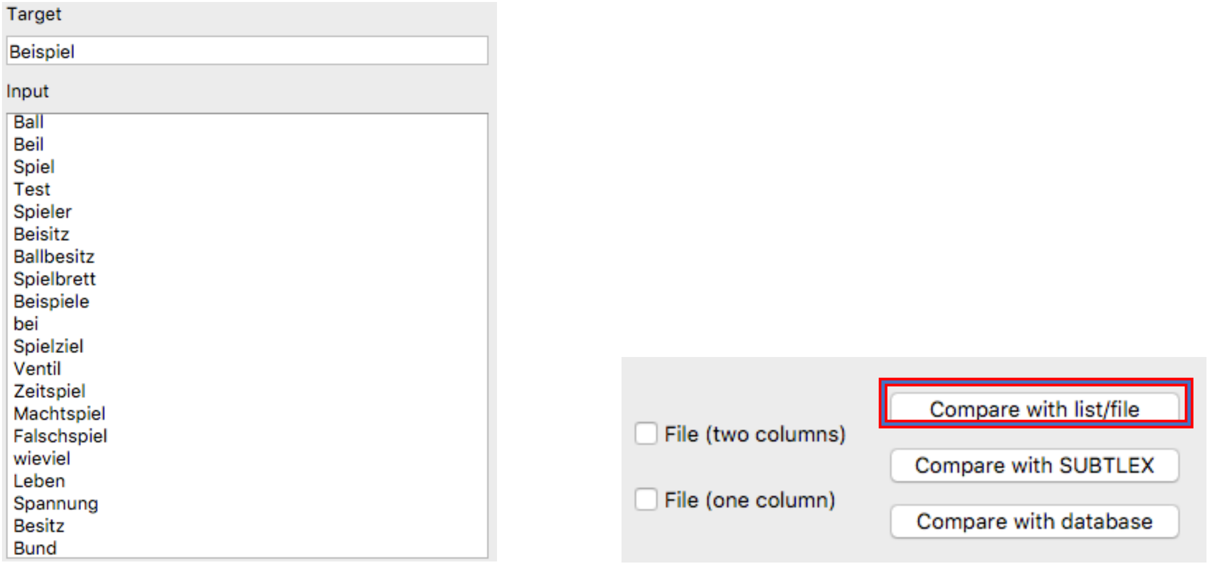
3. Using the System

This section provides a detail description of the tool functions. Each subsection will introduce the required input, the use and the resulting output. In general, all inputs need to be encoded as "utf-8", the output will be encoded likewise.

To transcribe the input and the target word, the tool uses the web application G2P, available at the following website https://clarin.phonetik.uni-muenchen.de/BASWebServices/interface/Grapheme2Phoneme. The output will be in SAMPA format which is an ASCII-based phonetic alphabet (Reichel & Kissler, 2014).

3.1 Compare target and list of words

One of the most important function of this tool is the option to compare a target word with a manually given list of words. For the comparison the target word and a list of words have to be typed in as shown on the left side of figure 7. It is required to type in one word per line. Even though usually a list of 20 words is preferred, the system also enables the user to process more or less than 20 words. To prevent an unintended amount of words the protocol notifies the user that more or less than 20 words will be tested.



Target

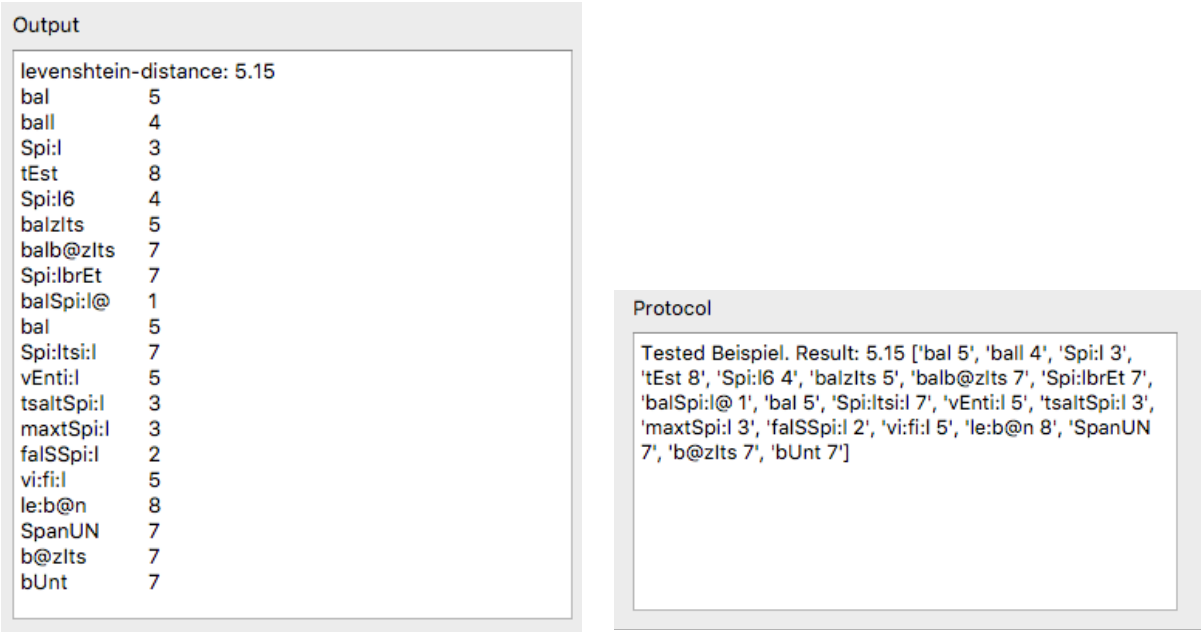
Beispiel

Input

```
Ball
Beil
Spiel
Test
Spieler
Beisitz
Ballbesitz
Spielbrett
Beispiele
bei
Spielziel
Ventil
Zeitspiel
Machtspiel
Falschspiel
wieviel
Leben
Spannung
Besitz
Bund
```

Figure 7. Example showing how to type in target word and list of words on the left side. The right side shows the buttons to start the tool.

When clicking on the button "Compare with list/file" without the use of any checkbox, the tool will begin to request the transcription of the given words via G2P. Afterwards it will calculate the phonological Levenshtein distance between each word and the target as well as their mean. The results will be printed in the text field "Output" as shown on figure 8 below.



Output

```
levenshtein-distance: 5.15
bal         5
ball        4
Spi:l       3
tEst        8
Spi:l6      4
balzIts     5
balb@zIts   7
Spi:lbrEt   7
balSpi:l@   1
bal         5
Spi:ltsi:l  7
vEnti:l     5
tsaltSpi:l  3
maxtSpi:l   3
falSSpi:l   2
vi:fi:l     5
le:b@n      8
SpanUN      7
b@zIts      7
bUnt        7
```

Protocol

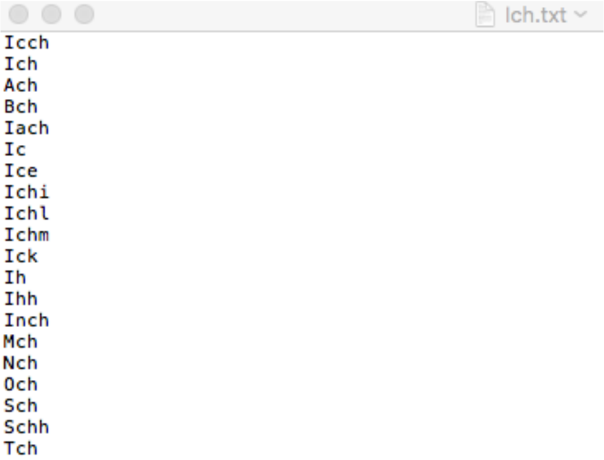
```
Tested Beispiel. Result: 5.15 ['bal 5', 'ball 4', 'Spi:l 3',
'tEst 8', 'Spi:l6 4', 'balzIts 5', 'balb@zIts 7', 'Spi:lbrEt 7',
'balSpi:l@ 1', 'bal 5', 'Spi:ltsi:l 7', 'vEnti:l 5', 'tsaltSpi:l 3',
'maxtSpi:l 3', 'falSSpi:l 2', 'vi:fi:l 5', 'le:b@n 8', 'SpanUN
7', 'b@zIts 7', 'bUnt 7']
```

Figure 8. Example showing the output on the left side and the protocol on the right side.

In addition, the result will be printed into the field "Protocol" and into the corresponding text file so the result will not be lost after testing more than one input.

3.2 Compare target with a one-column file

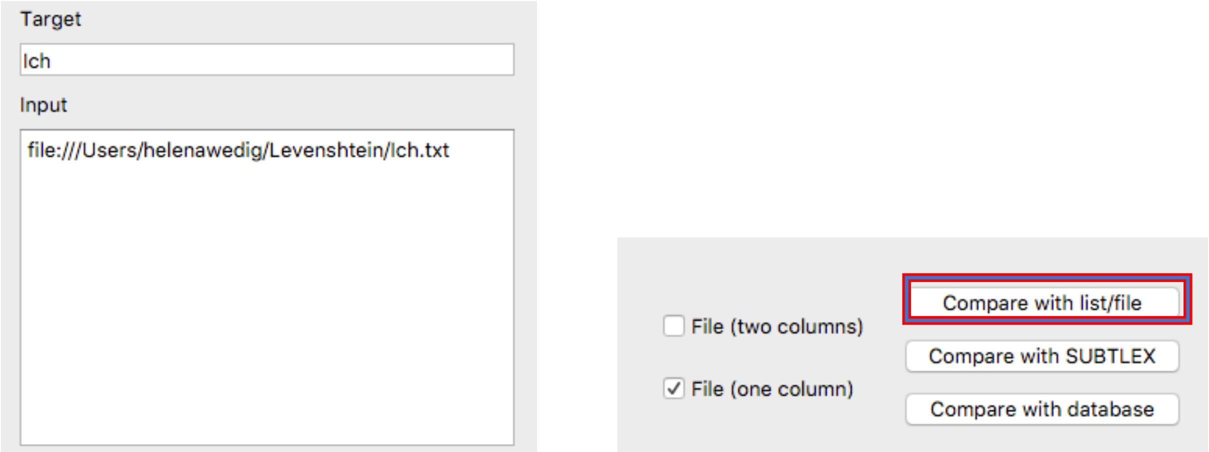
To compare the target word with a list of words, that is not given in the text field "Input", but in a text file, it is possible to drop the text file into the text field "Input". Therefore, the text file has to be encoded as "utf-8" and it has to have one word per line (as seen in figure 9).



```
Icch
Ich
Ach
Bch
Iach
Ic
Ice
Ichi
Ichl
Ichm
Ick
Ih
Ihh
Inch
Mch
Nch
Och
Sch
Schh
Tch
```

Figure 9. Possible input file with one column.

After dropping it there, its path will appear in the text field. As described in chapter 3.1 the target word has to be in the text field "Target". To read the file it is necessary to check "File (One column)" and then click on "Compare with file/list" (figure 10).



Target: Ich

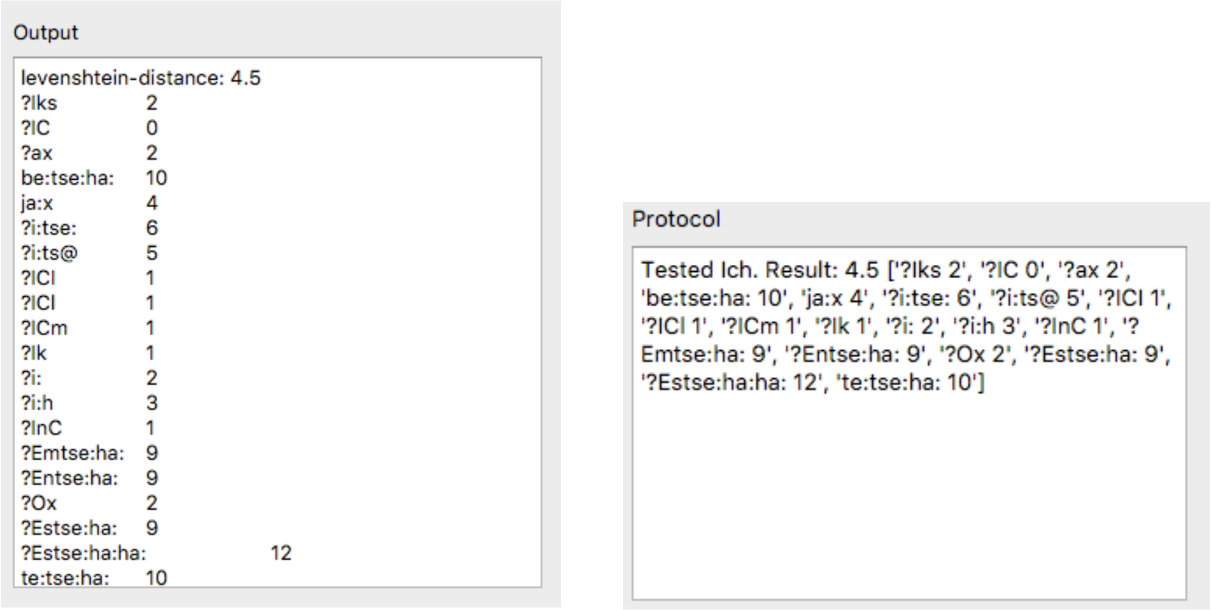
Input: file:///Users/helenawedig/Levenshtein/Ich.txt

□ File (two columns)

☑ File (one column)

Figure 10. Examples showing the input on the left and the checkboxes on the right side.

After processing the file and calculating the phonological Levenshtein distance the result will be printed into the text field "Output", into the field "Protocol" and into the corresponding text file (figure 11).



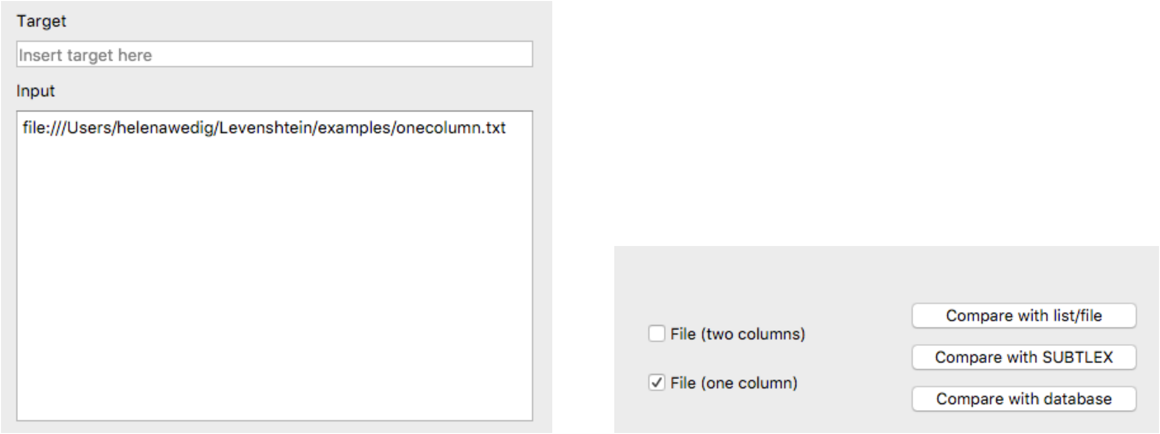
```
levenshtein-distance: 4.5
?Iks          2
?IC           0
?ax           2
be:tse:ha:    10
ja:x          4
?i:tse:       6
?i:ts@        5
?ICl          1
?ICl          1
?ICm          1
?Ik           1
?i:           2
?i:h          3
?InC          1
?Emtse:ha:    9
?Entse:ha:    9
?Ox           2
?Estse:ha:    9
?Estse:ha:ha:               12
te:tse:ha:    10
```

```
Tested Ich. Result: 4.5 ['?Iks 2', '?IC 0', '?ax 2',
'be:tse:ha: 10', 'ja:x 4', '?i:tse: 6', '?i:ts@ 5', '?ICl 1',
'?ICl 1', '?ICm 1', '?Ik 1', '?i: 2', '?i:h 3', '?InC 1', '?
Emtse:ha: 9', '?Entse:ha: 9', '?Ox 2', '?Estse:ha: 9',
'?Estse:ha:ha: 12', 'te:tse:ha: 10']
```

Figure 11. Example showing the output on the left side and the protocol on the right side.

3.3 Cross-compare a one-column file

Given a one-column file with a list of words and no target word, the tool will compare the given words with each other. It will calculate the Levenshtein-distance for each possible pair of words. Again, the user has to drop the file in the text field "input", check "File (One column)" and then click on "Compare with file/list" (figure 12). The following examples use the same text file as shown in figure 9.



Target

Insert target here

Input

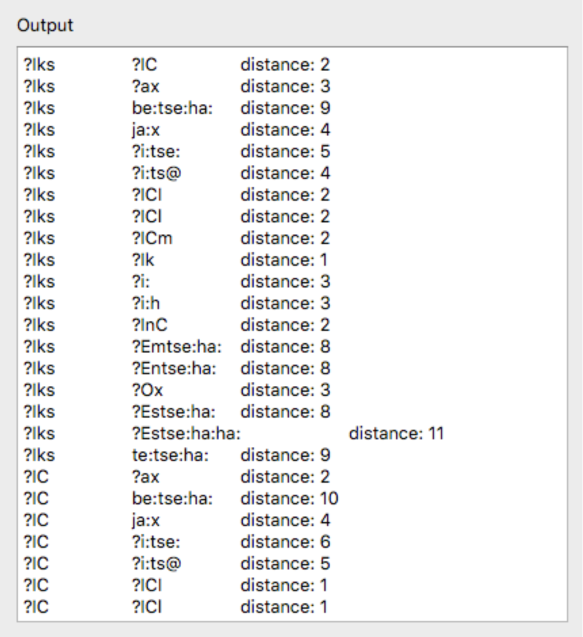
file:///Users/helenawedig/Levenshtein/examples/onecolumn.txt

□ File (two columns)

☑ File (one column)

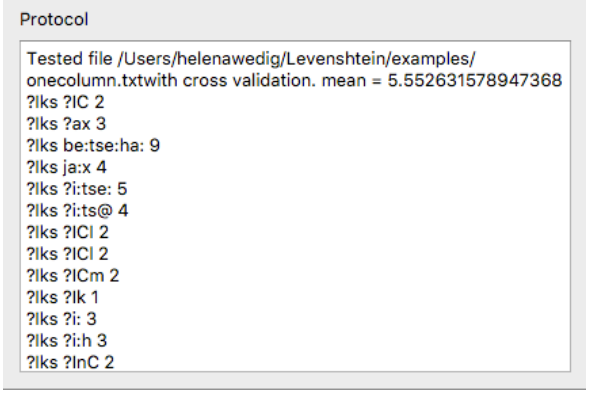
Figure 12. Examples showing the input on the left and the checkboxes on the right side.

Finishing the calculation, the tool will present the result in the text fields "output" and "protocol" as well as in the corresponding files. Because the cross-comparing would lead to double entries in the output, the tool ignores pairs that have already been compared (figure 13).



Output

```
?Iks        ?IC          distance: 2
?Iks        ?ax          distance: 3
?Iks        be:tse:ha:   distance: 9
?Iks        ja:x         distance: 4
?Iks        ?i:tse:      distance: 5
?Iks        ?i:ts@       distance: 4
?Iks        ?ICI         distance: 2
?Iks        ?ICI         distance: 2
?Iks        ?ICm         distance: 2
?Iks        ?Ik          distance: 1
?Iks        ?i:          distance: 3
?Iks        ?i:h         distance: 3
?Iks        ?InC         distance: 2
?Iks        ?Emtse:ha:   distance: 8
?Iks        ?Entse:ha:   distance: 8
?Iks        ?Ox          distance: 3
?Iks        ?Estse:ha:   distance: 8
?Iks        ?Estse:ha:ha:          distance: 11
?Iks        te:tse:ha:   distance: 9
?IC         ?ax          distance: 2
?IC         be:tse:ha:   distance: 10
?IC         ja:x         distance: 4
?IC         ?i:tse:      distance: 6
?IC         ?i:ts@       distance: 5
?IC         ?ICI         distance: 1
?IC         ?ICI         distance: 1
```



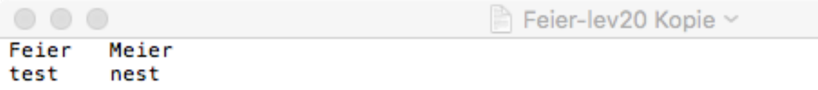
Protocol

```
Tested file /Users/helenawedig/Levenshtein/examples/
onecolumn.txtwith cross validation. mean = 5.552631578947368
?Iks ?IC 2
?Iks ?ax 3
?Iks be:tse:ha: 9
?Iks ja:x 4
?Iks ?i:tse: 5
?Iks ?i:ts@ 4
?Iks ?ICI 2
?Iks ?ICI 2
?Iks ?ICm 2
?Iks ?Ik 1
?Iks ?i: 3
?Iks ?i:h 3
?Iks ?InC 2
```

Figure 13. Examples showing the input on the left and the checkboxes on the right side.

3.4 Compare target within a two-column file

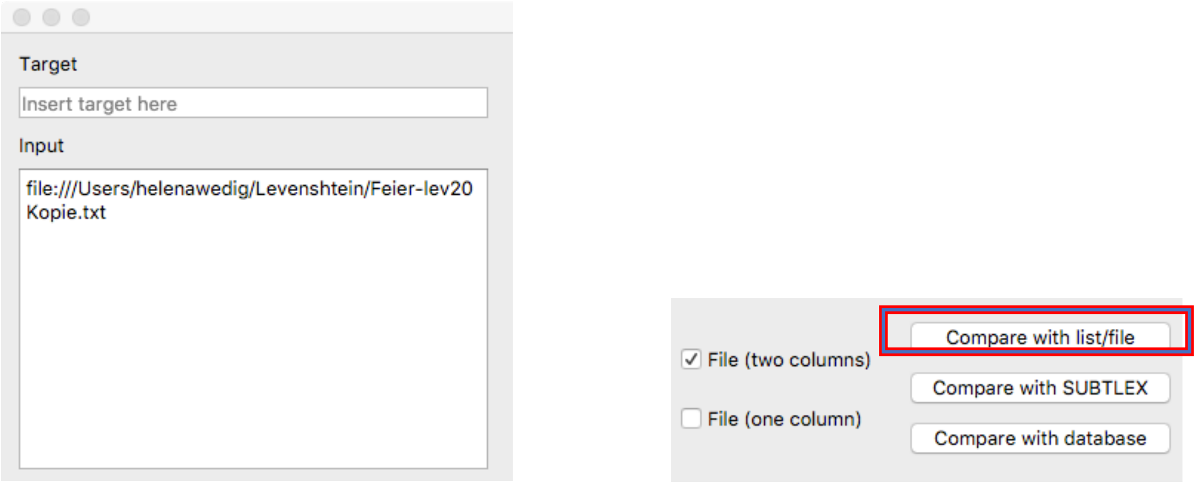
The tool enables the user to compare words of a two-column file. The first column has to include the target word, the second column the corresponding input (figure 14). Processing the "utf-8"-encoded file, the result will be the Levenshtein distance between the token in the first and the token in the second column.



```
Feier   Meier
test    nest
```

Figure 14. Possible input file with two columns.

To start the process, it is required to drop the corresponding file in the text field "Input" and check the box "File (Two Columns)" (figure 15). It is not necessary to put a target word into the text field "target".



Target

Insert target here

Input

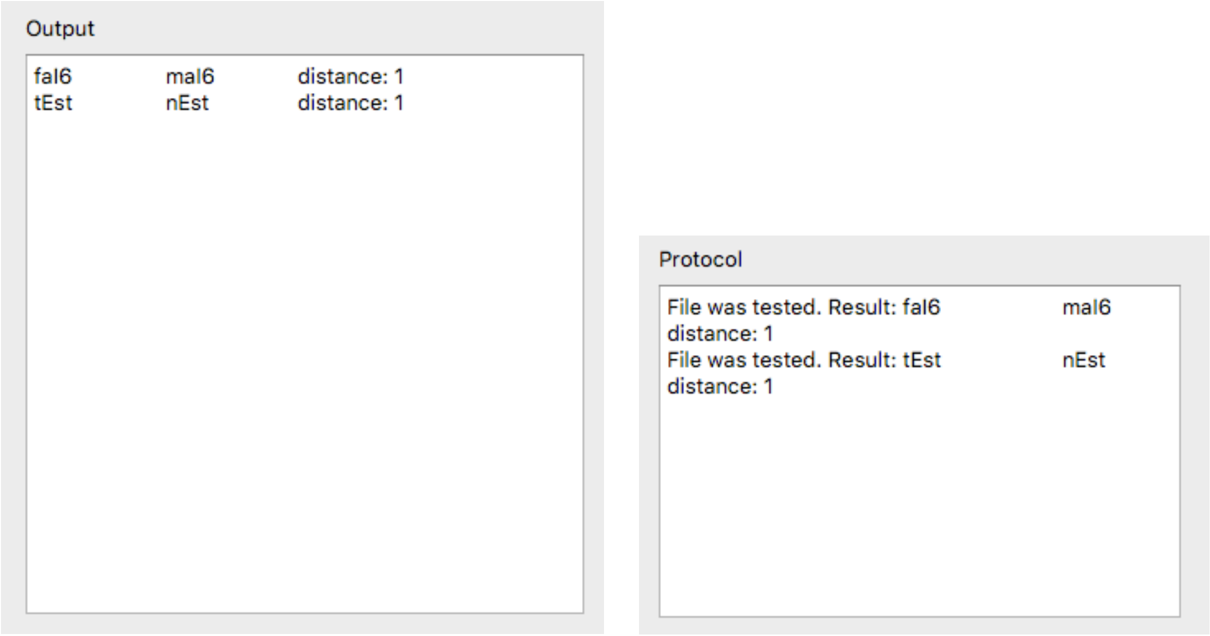
```
file:///Users/helenawedig/Levenshtein/Feier-lev20
Kopie.txt
```

☑ File (two columns)

□ File (one column)

Figure 15. Examples showing the input on the left and the checkboxes on the right side.

After dropping the text file and checking the box, a click on the button "Compare with file/list" will start to calculate the Levenshtein distance. Afterwards the result will be shown in the text fields "Output and "protocol" as well as saved in the corresponding text files (figure 16).



Output

```
fal6          mal6          distance: 1
tEst          nEst          distance: 1
```

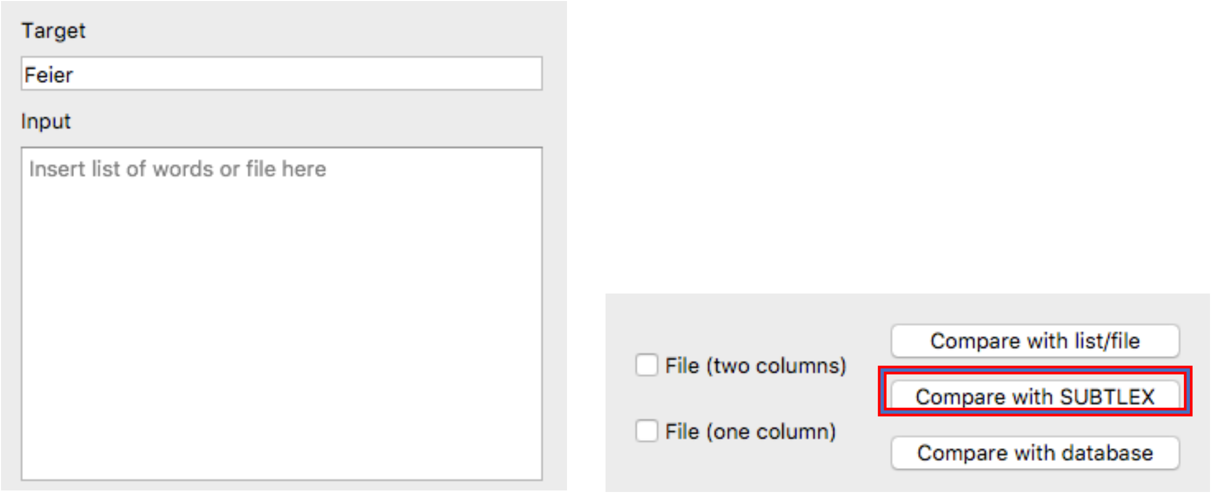
Protocol

```
File was tested. Result: fal6                mal6
distance: 1
File was tested. Result: tEst                nEst
distance: 1
```

Figure 16. Example showing the output on the left side and the protocol on the right side.

3.5 Compare target with SUBTLEX

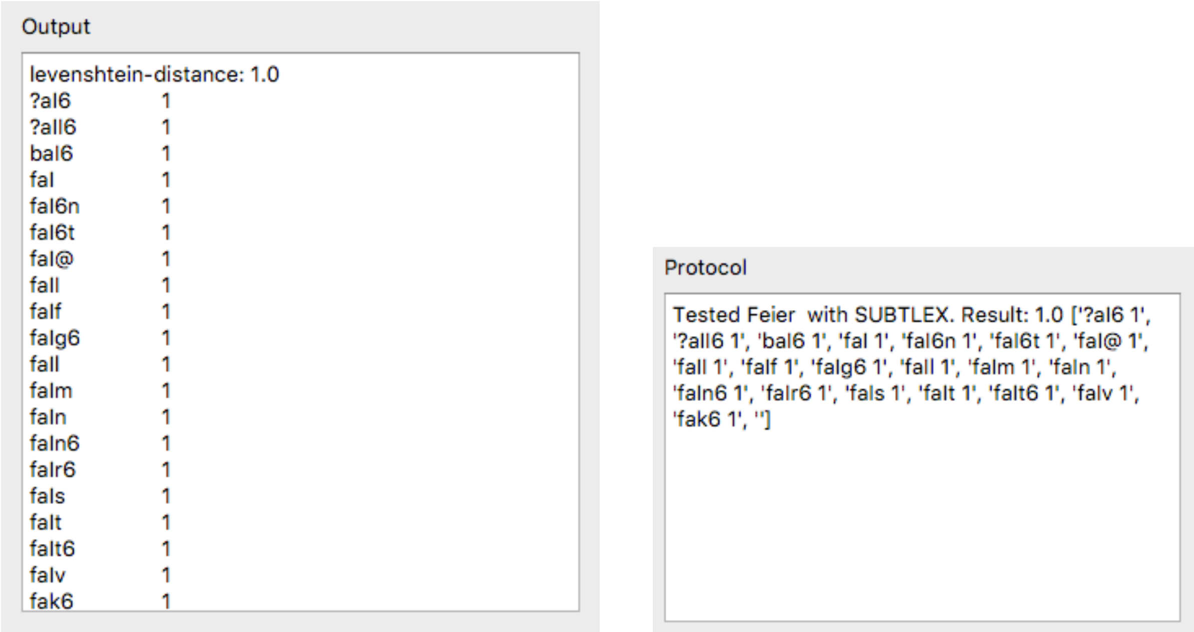
Another main function of this tool is the ability to search for the 20 nearest neighbors in a given corpus. In this case the SUBTLEX corpus is already transcribed. The tool compares the transcribed target word with the processed SUBTLEX and states the 20 lowest Levenshtein distances. To use this function, it is only required to type in a target word and click on the button "Compare with SUBTLEX" (figure 17).



Target: Feier

Figure 17. Examples showing the input on the left and the checkboxes on the right side.

The process will start immediately. After finding the 20 lowest Levenshtein distances, the tool will create a text file, which contains the 20 neighbors and its Levenshtein distance in regard to the target word. Furthermore, the mean of the distances and the individual results will be printed into the text fields "Output" and "Protocol" as well as into the corresponding text file (figure 18).



Output

```
levenshtein-distance: 1.0
?al6        1
?all6       1
bal6        1
fal         1
fal6n       1
fal6t       1
fal@        1
fall        1
falf        1
falg6       1
fall        1
falm        1
faln        1
faln6       1
falr6       1
fals        1
falt        1
falt6       1
falv        1
fak6        1
```

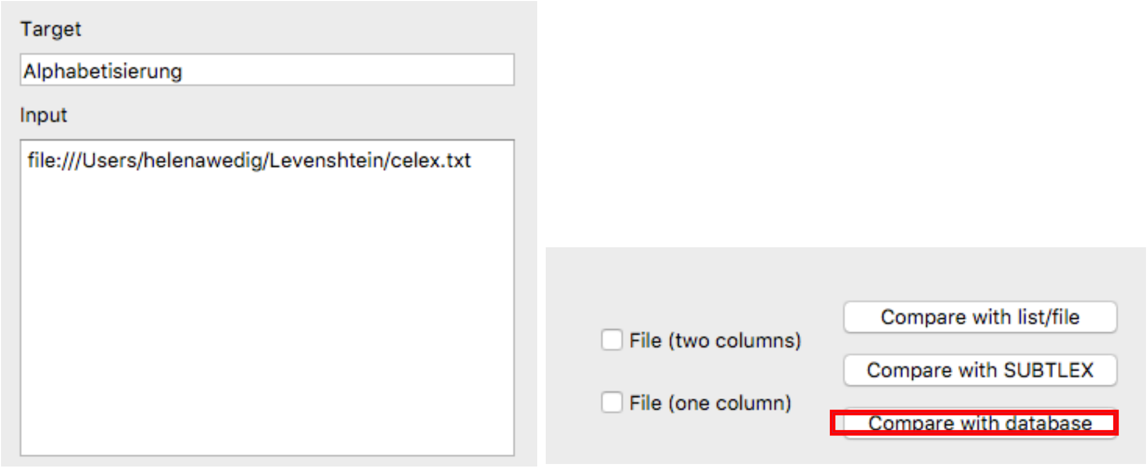
Protocol

```
Tested Feier  with SUBTLEX. Result: 1.0 ['?al6 1',
'?all6 1', 'bal6 1', 'fal 1', 'fal6n 1', 'fal6t 1', 'fal@ 1',
'fall 1', 'falf 1', 'falg6 1', 'fall 1', 'falm 1', 'faln 1',
'faln6 1', 'falr6 1', 'fals 1', 'falt 1', 'falt6 1', 'falv 1',
'fak6 1', '']
```

Figure 18. Example showing the output on the left side and the protocol on the right side.

3.6 Compare target with an individual corpus

As stated in the subsection "Compare target with SUBTLEX", it is also possible to compare a target word with a corpus of choice. To use this function the tool needs a transcribed corpus. Every corpus can be transcribed automatically via G2P as explained in the following subsection 3.6 and then be pasted into the text field "input".

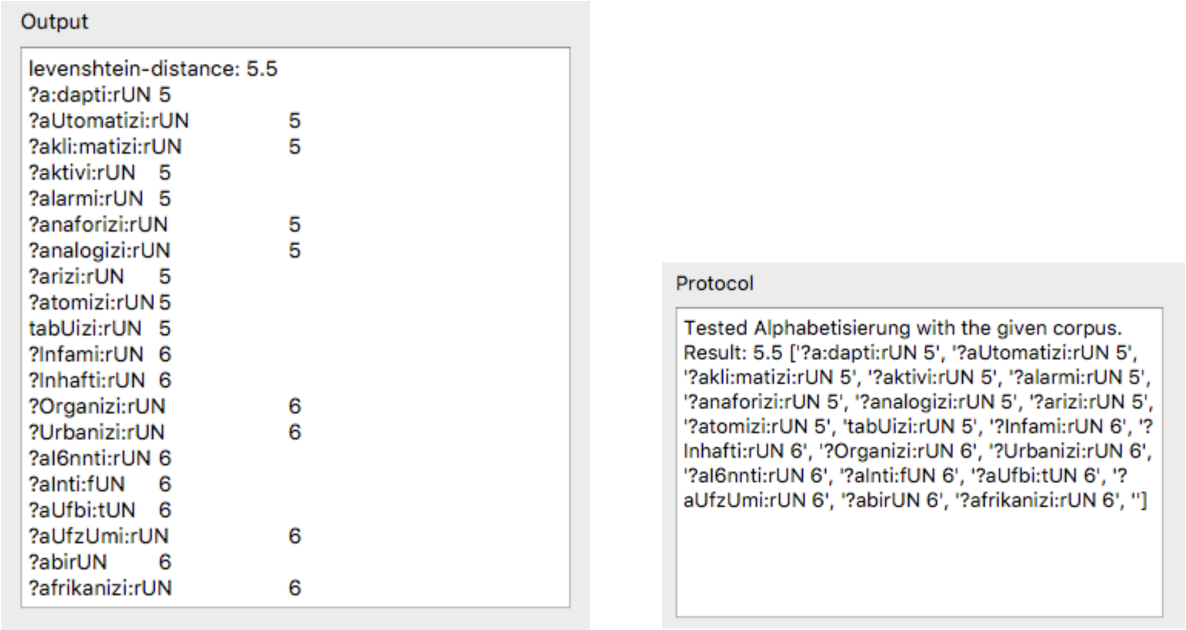


Target: Alphabetisierung

Input: file:///Users/helenawedig/Levenshtein/celex.txt

Figure 19. Examples showing the input on the left and the checkboxes on the right side.

After typing in the target word in the text field "Target" and clicking on the button "Compare with Corpus" (figure 19), the tool will calculate the 20 nearest Levenshtein distances and print the result into the text fields "Output" and "Protocol" in addition to the corresponding text file (figure 20).



```
levenshtein-distance: 5.5
?a:dapti:rUN 5
?aUtomatizi:rUN          5
?akli:matizi:rUN         5
?aktivi:rUN   5
?alarmi:rUN  5
?anaforizi:rUN           5
?analogizi:rUN           5
?arizi:rUN     5
?atomizi:rUN 5
tabUizi:rUN  5
?Infami:rUN  6
?Inhafti:rUN  6
?Organizi:rUN            6
?Urbanizi:rUN            6
?al6nnti:rUN 6
?alnti:fUN     6
?aUfbi:tUN   6
?aUfzUmi:rUN             6
?abirUN       6
?afrikanizi:rUN          6
```

```
Tested Alphabetisierung with the given corpus.
Result: 5.5 ['?a:dapti:rUN 5', '?aUtomatizi:rUN 5',
'?akli:matizi:rUN 5', '?aktivi:rUN 5', '?alarmi:rUN 5',
'?anaforizi:rUN 5', '?analogizi:rUN 5', '?arizi:rUN 5',
'?atomizi:rUN 5', 'tabUizi:rUN 5', '?Infami:rUN 6', '?
Inhafti:rUN 6', '?Organizi:rUN 6', '?Urbanizi:rUN 6',
'?al6nnti:rUN 6', '?alnti:fUN 6', '?aUfbi:tUN 6', '?
aUfzUmi:rUN 6', '?abirUN 6', '?afrikanizi:rUN 6', '']
```

Figure 20. Example showing the output on the left side and the protocol on the right side.

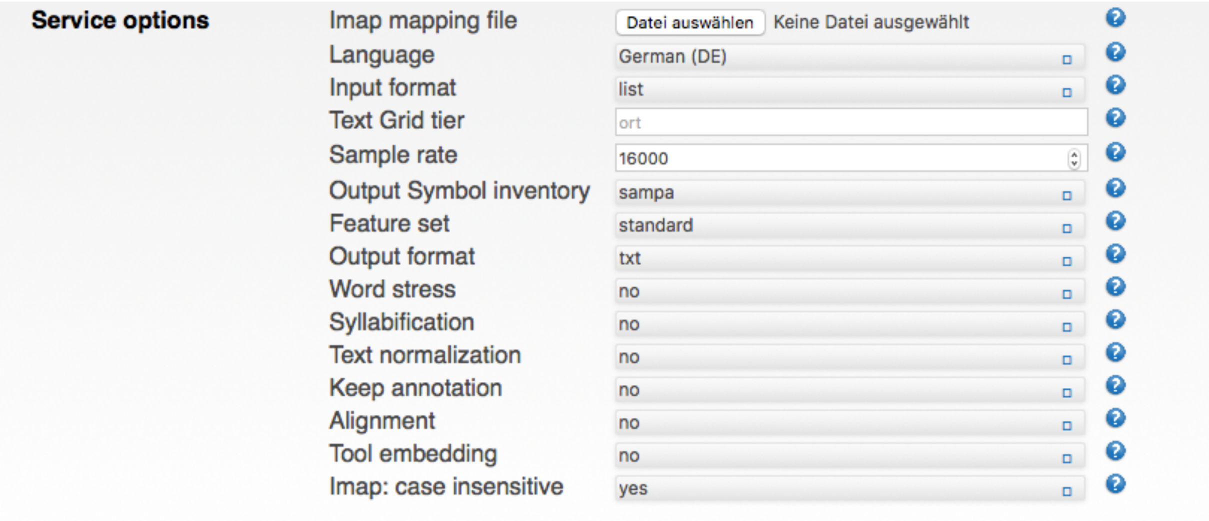
The next subsection explains how to modify a corpus file, so that it can be used by the developed tool.

3.7 Excursus: Modifying a corpus

To use any corpus, it is necessary to adjust its format. The corpus file has to have one word, a semicolon and its SAMPA version per line. To produce such a file, it has to be transcribed by G2P[1]. G2P requires a file that has a maximum of 100.000 tokens and

[1] available at https://clarin.phonetik.uni-muenchen.de/BASWebServices/interface/Grapheme2Phoneme

one token per line. It can be used via a web application. The following figure (21) shows the options that need to be chosen:

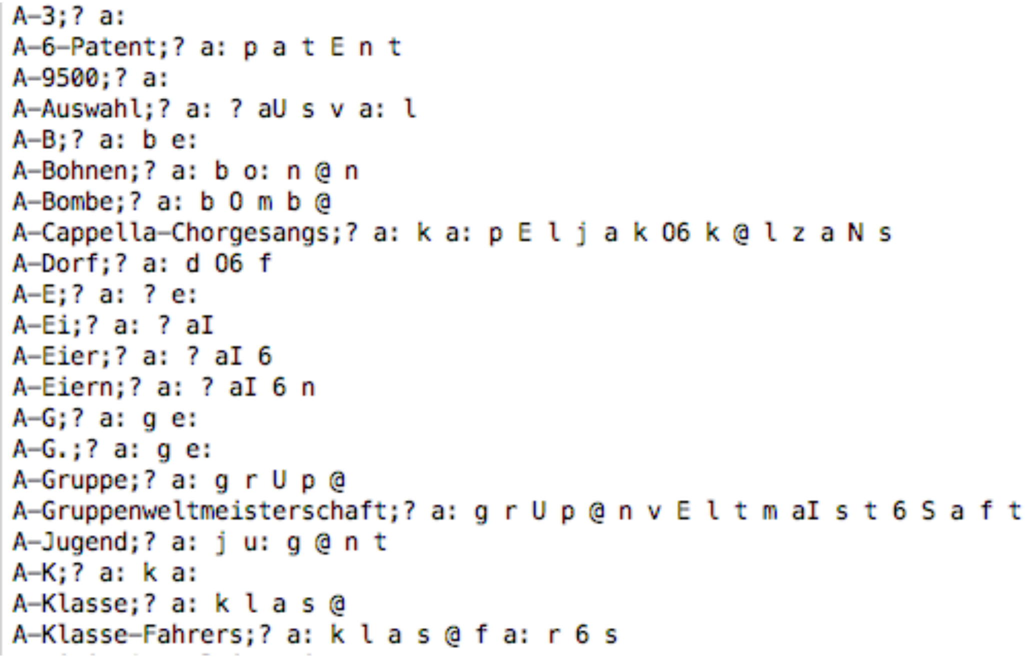


Service options

Imap mapping file	Datei auswählen Keine Datei ausgewählt
Language	German (DE)
Input format	list
Text Grid tier	ort
Sample rate	16000
Output Symbol inventory	sampa
Feature set	standard
Output format	txt
Word stress	no
Syllabification	no
Text normalization	no
Keep annotation	no
Alignment	no
Tool embedding	no
Imap: case insensitive	yes

Figure 21. Necessary options to use G2P via the web application.

After using G2P the file has the following format and can be used to calculate the 20 nearest Levenshtein distances (figure 22):



```
A-3;? a:
A-6-Patent;? a: p a t E n t
A-9500;? a:
A-Auswahl;? a: ? aU s v a: l
A-B;? a: b e:
A-Bohnen;? a: b o: n @ n
A-Bombe;? a: b O m b @
A-Cappella-Chorgesangs;? a: k a: p E l j a k O6 k @ l z a N s
A-Dorf;? a: d O6 f
A-E;? a: ? e:
A-Ei;? a: ? aI
A-Eier;? a: ? aI 6
A-Eiern;? a: ? aI 6 n
A-G;? a: g e:
A-G.;? a: g e:
A-Gruppe;? a: g r U p @
A-Gruppenweltmeisterschaft;? a: g r U p @ n v E l t m aI s t 6 S a f t
A-Jugend;? a: j u: g @ n t
A-K;? a: k a:
A-Klasse;? a: k l a s @
A-Klasse-Fahrers;? a: k l a s @ f a: r 6 s
```

Figure 22. Formatted corpus.

4. Producing the protocol

This section describes and depicts all error messages and results that can be generated by the system. Each protocol and result file will be saved in the folder "protocols" or "results" with the name "protocol" or "result" and an additional timestamp.

4.1 Error messages

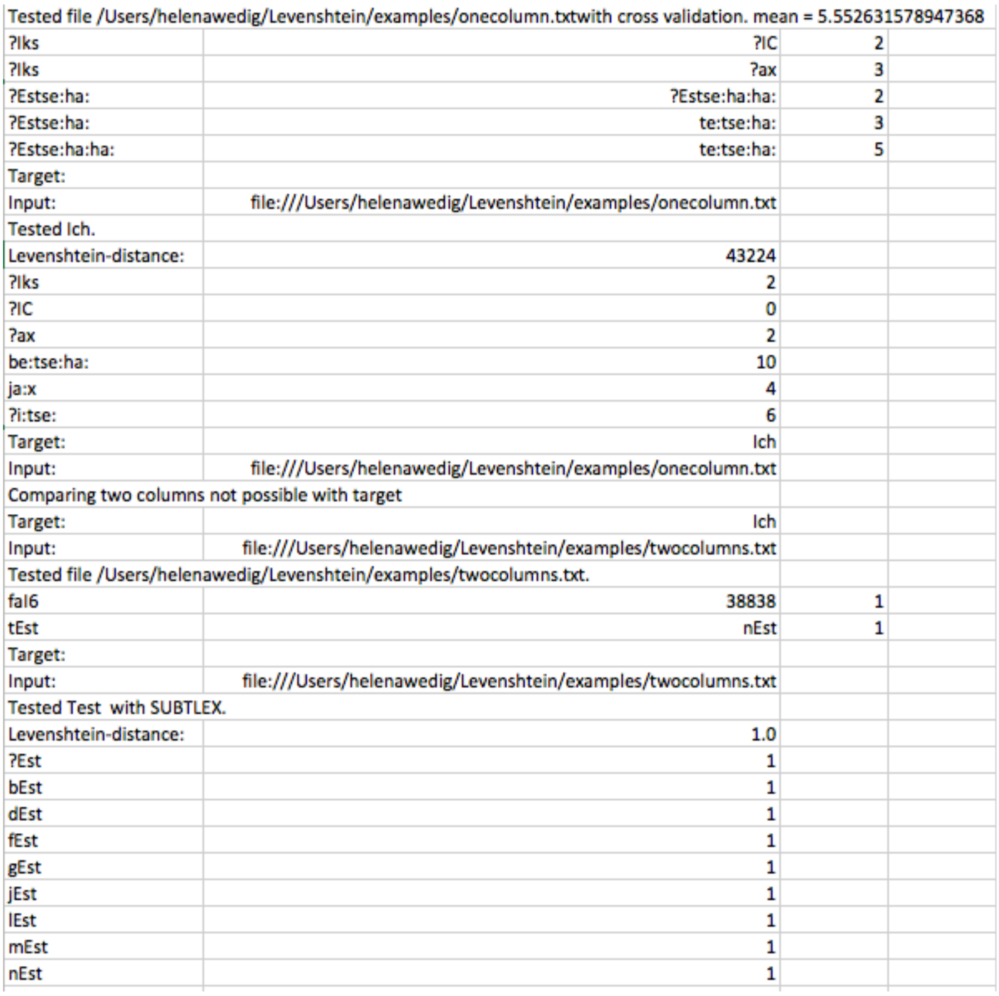
To keep the tool as easy to use as possible several messages notify the user of possible sources of error. The following table gives an overview of them and their corresponding solution.

error message	origin	solution
Cannot read file TITLE	The file does not exist in the given path.	Check if the file exists in the given directory.
Comparing two columns is not possible with given target	The user stated a target word, but wants to compare a file with two columns.	Delete the target word.
Found list, expected file.	The box "file (one column)" is checked but the user stated a list of words.	Either uncheck the box or paste in a file.
Found file, expected list.	The user did not tick a box or the button "Compare with Corpus" but did give a file.	Either check a box, click on the button "Compare with Corpus" or paste in a list of words.
No data: target.	The text field "target" is empty.	Type in a target word.

No data: input.	The text field "input" is empty.	Type in a list of words or a file or use "Compare with SUBTLEX".
Expected file with two columns but found one column.	The user checked in "File (two columns)" but the file has only one.	Either check in "file (one column)" and give a target word or paste in a file with two columns.
No corpus found.	The given file does not have the right format.	Please refer to chapter 3.5

4.2. Printing the protocol

The shown error messages are a part of the protocol that will be produced in the process. The error messages and the source of them in form of the input and the target word will be saved in a protocol file with one error and its source per line. The file also includes the result of a correct calculation with one calculation, its target and its input per line. The following figure 23 shows an example of this protocol file which is in csv-format.

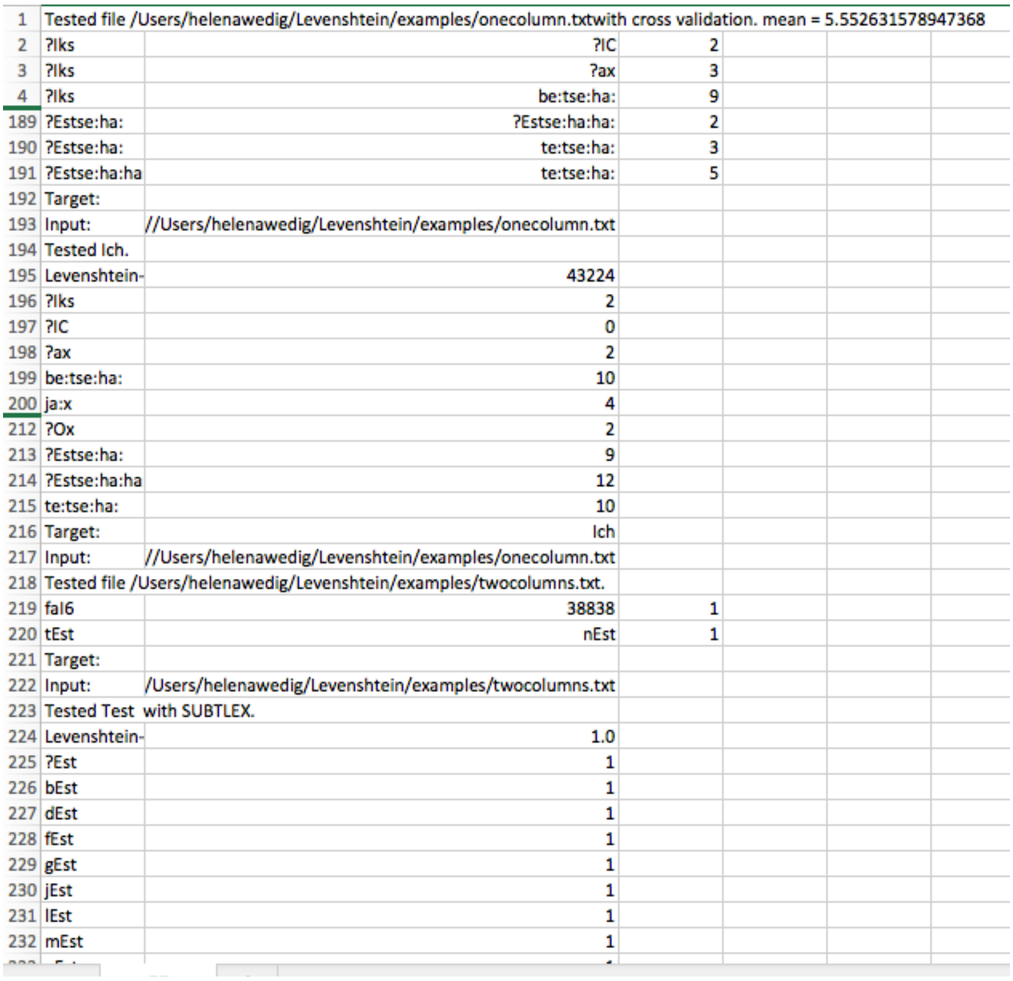


Tested file /Users/helenawedig/Levenshtein/examples/onecolumn.txtwith cross validation. mean = 5.552631578947368			
?Iks	?IC	2	
?Iks	?ax	3	
?Estse:ha:	?Estse:ha:ha:	2	
?Estse:ha:	te:tse:ha:	3	
?Estse:ha:ha:	te:tse:ha:	5	
Target:			
Input:	file:///Users/helenawedig/Levenshtein/examples/onecolumn.txt		
Tested Ich.			
Levenshtein-distance:	43224		
?Iks	2		
?IC	0		
?ax	2		
be:tse:ha:	10		
ja:x	4		
?i:tse:	6		
Target:	Ich		
Input:	file:///Users/helenawedig/Levenshtein/examples/onecolumn.txt		
Comparing two columns not possible with target			
Target:	Ich		
Input:	file:///Users/helenawedig/Levenshtein/examples/twocolumns.txt		
Tested file /Users/helenawedig/Levenshtein/examples/twocolumns.txt.			
fal6	38838	1	
tEst	nEst	1	
Target:			
Input:	file:///Users/helenawedig/Levenshtein/examples/twocolumns.txt		
Tested Test with SUBTLEX.			
Levenshtein-distance:	1.0		
?Est	1		
bEst	1		
dEst	1		
fEst	1		
gEst	1		
jEst	1		
lEst	1		
mEst	1		
nEst	1		

Figure 23. Protocol file that includes the error messages and the correct calculated results. To show as much output as possible some results were hidden.

4.3. Printing the results

To separate the error messages and the results given by the system, the results will be printed in both, the result csv-file and the protocol csv-file. Formatted as shown in this example (figure 24):



1	Tested file /Users/helenawedig/Levenshtein/examples/onecolumn.txtwith cross validation. mean = 5.552631578947368		
2	?Iks	?IC	2
3	?Iks	?ax	3
4	?Iks	be:tse:ha:	9
189	?Estse:ha:	?Estse:ha:ha:	2
190	?Estse:ha:	te:tse:ha:	3
191	?Estse:ha:ha	te:tse:ha:	5
192	Target:		
193	Input:	//Users/helenawedig/Levenshtein/examples/onecolumn.txt	
194	Tested Ich.		
195	Levenshtein-	43224	
196	?Iks	2	
197	?IC	0	
198	?ax	2	
199	be:tse:ha:	10	
200	ja:x	4	
212	?Ox	2	
213	?Estse:ha:	9	
214	?Estse:ha:ha	12	
215	te:tse:ha:	10	
216	Target:	Ich	
217	Input:	//Users/helenawedig/Levenshtein/examples/onecolumn.txt	
218	Tested file /Users/helenawedig/Levenshtein/examples/twocolumns.txt.		
219	fal6	38838	1
220	tEst	nEst	1
221	Target:		
222	Input:	/Users/helenawedig/Levenshtein/examples/twocolumns.txt	
223	Tested Test	with SUBTLEX.	
224	Levenshtein-	1.0	
225	?Est	1	
226	bEst	1	
227	dEst	1	
228	fEst	1	
229	gEst	1	
230	jEst	1	
231	lEst	1	
232	mEst	1	

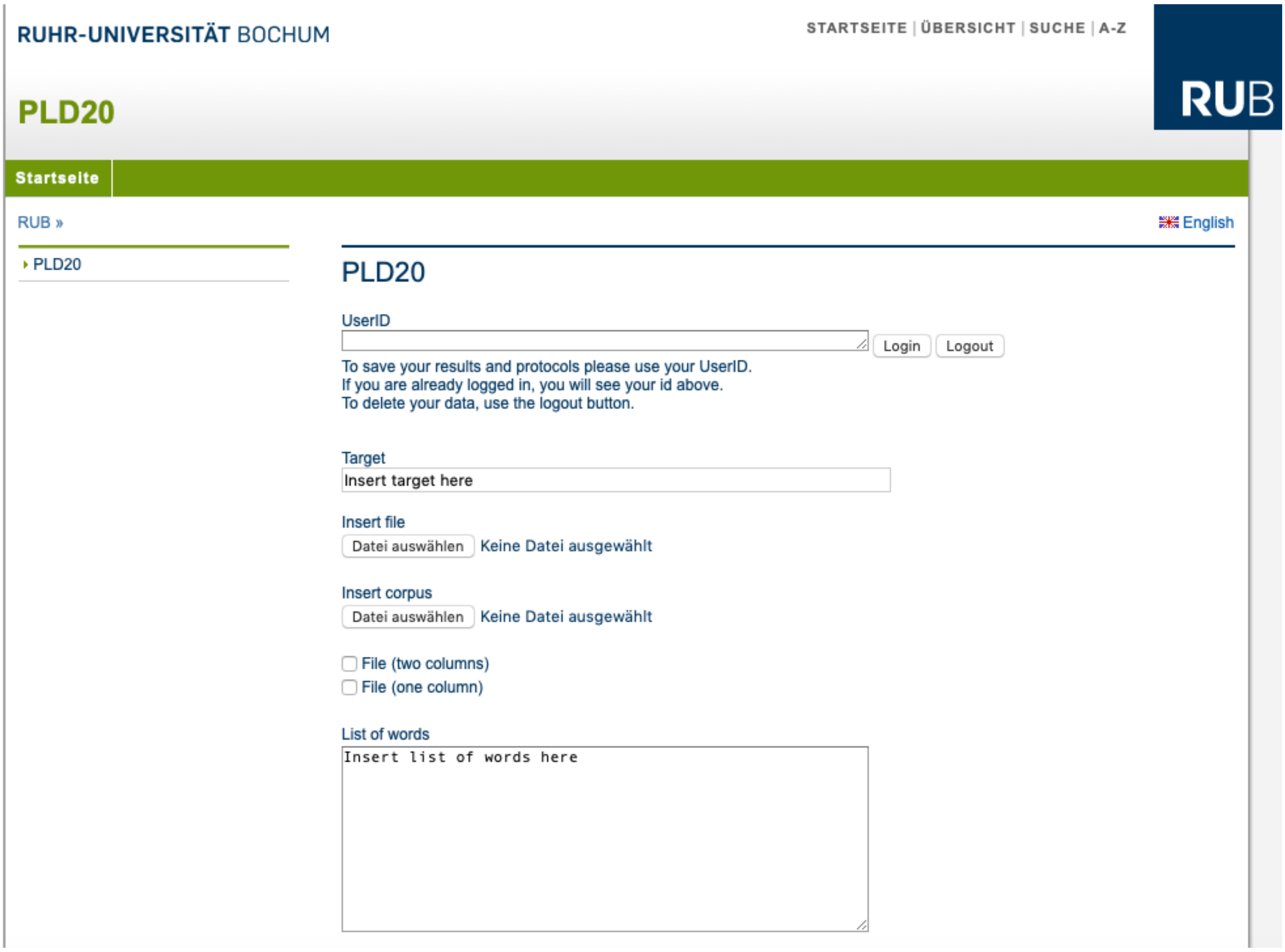
Figure 24. Result file corresponding to the protocol file shown in figure 21. To show as much output as possible some results were hidden.

The result begins with the tested target word, following the mean Levenshtein distance and the tested input words in their phonological form plus their individual Levenshtein distance. To remember the tested input and the target, these will be printed afterwards.

5. Using the Website

As described in the introduction an online tool has just been developed. It is not necessary anymore to install python or download any packages. The user has access to all the functions of the tool via the internet.

As shown below (figure 25 and 26) the website has the same mechanics as the offline tool, so the following section will only describe the usage of the website in short form and then refer to its offline version.



To save your results and protocols please use your UserID.
If you are already logged in, you will see your id above.
To delete your data, use the logout button.

Insert target here

Keine Datei ausgewählt

Keine Datei ausgewählt

□ File (two columns)
□ File (one column)

Insert list of words here

Figure 25. First half of the developed website.

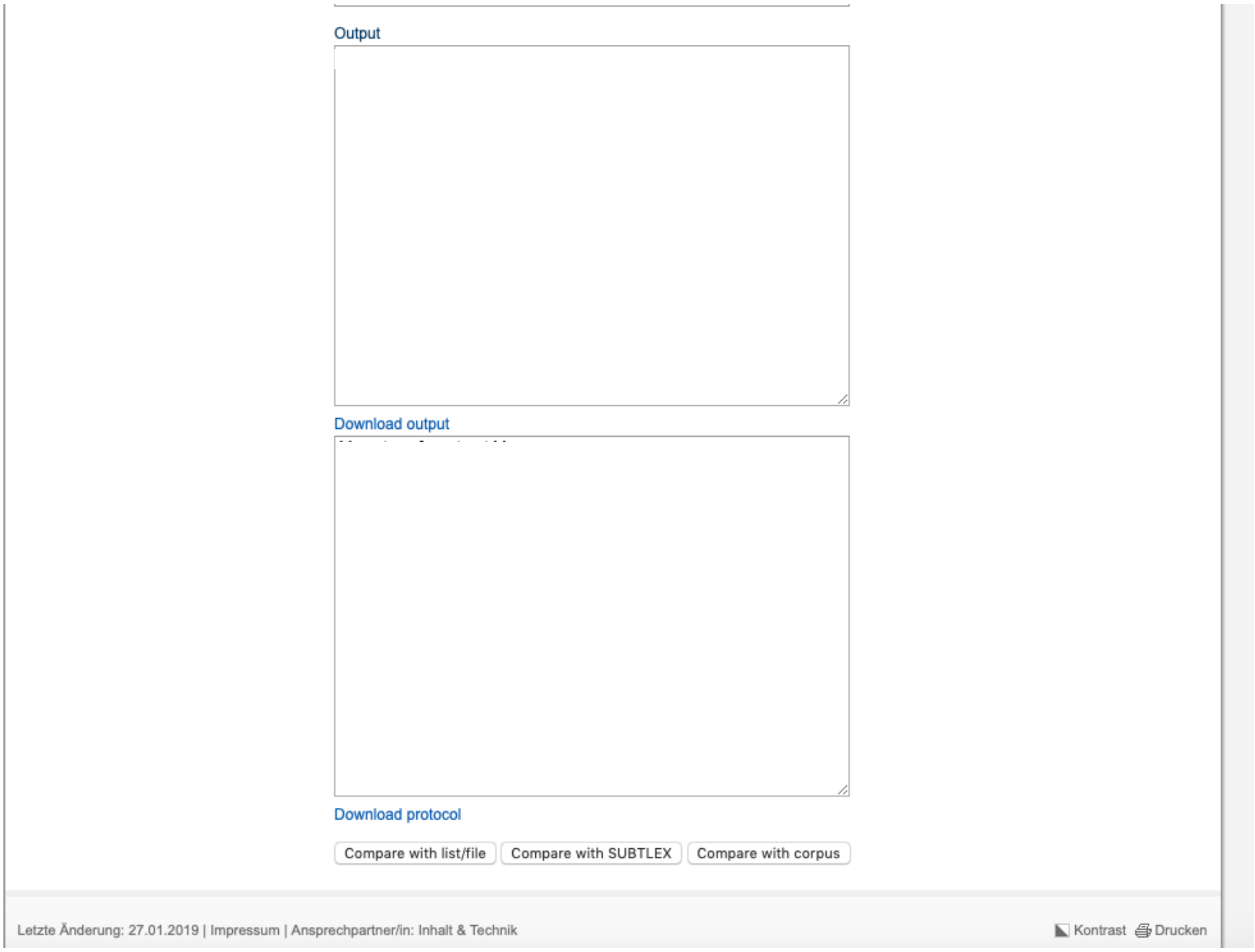


Figure 26. Second half of the developed website.

5.1 UserID – Login and Logout

To enable the same functions as the offline tool, the application has to have the possibility to save the results and protocols the user produced before. To enable this, it is necessary to use an UserID. If logged in, the result wil be saved in a user specific file on the server. It is possible to download these files by clicking “Download results” or “Download protocol” as shown on figure 26.

If it is desired to save the data, users can login by typing in a username and clicking “Login”. The website will set a cookie and thus remember the user’s PC. The button “Logout” is required to stop saving the data and remove the files from the server.

Furthermore, it is possible to use the website without using a UserID. If there is no UserID in the text field or the user has not clicked on “Login” the application will also save a cookie but use the time and date to differentiate the data instead.

5.2 Compare target and list of words or a one-column file

As stated in the sections 3.1 and 3.2 the tool enables the user to compare a given target word with a list of words. The list of words can either be typed in directly using the text field “list of words” or imported from a file by clicking “insert file”. The online version accepts a typed input separated by commas, lines or spaces. After putting in a list of words or a file and a target word, the checkbox “File (one column)” has to be checked and the user has to use the button “Compare with list/file”. The tool will start processing the given input and produce the output. As seen in the offline tool the output will be printed in the “output” text field and the protocol in the “protocol” text field. Please be aware that the newest result will be added to the text field’s

5.3 Cross-compare a one-column file

When only uploading a one-column file and checking “File (one column)”, clicking the button “Compare with list/file” will let the tool cross compare the content of the given file. As seen in section 3.3 the result consists of the Levenshtein-distance given for each pair of words of the input. After processing the data, the results will again be shown in the text field “result” and the protocol in the text field “protocol”.

5.4 Compare target within a two-column file

As seen in section 3.4 the tool allows the user to compare many targets at once. Therefore, a file with two columns is needed. In this case the tool compares a pair of words written in one line with itself. To use this function in the online version the user has to upload a file with two columns, check in the checkbox “File (two columns)” and click on the button “Compare with list/file”. Again, the process starts and prints out the results in text fields “results” and “protocol”.

5.5 Compare target with SUBTLEX or an individual corpus

In addition to the comparison of a target word and words given by the user, it is also possible to compare a target word with a corpus of choice and get the 20 nearest neighbors plus the mean Levenshtein-distance in return. After typing in the target, it is possible to upload an own corpus using the upload button “Insert corpus” and clicking on “Compare with corpus” or use the given SUBTLEX by clicking on “Compare with SUBTLEX”. Again, the output will be printed in the two text fields “result” and “protocol”. Please be aware that each corpus has to be converted to SAMPA before using the tool.

6. Expandability

The first version of this tool focuses on processing German corpora, but as the algorithm to calculate the PLD20 is universal, it is possible to import any corpus of choice this includes also any language of choice. The only important aspects to use the tool with any language is that the given input and given corpus are derived from the same language and the language G2P transcribes is adjusted. The last aspect has to be changed in the algorithm but can be added to GUI.

Furthermore, the algorithm enables the user to not only calculate the phonological but also the orthographical Levenshtein distance. To use this alternative just one line in the code has to be changed. If scientists also urge to calculate the orthographic Levenshtein distance the option will be added to the GUI.

Finally, the format of the output is variable. First it will be pasted in the corresponding output file with one word and its Levenshtein distance per line. If it is necessary to have for example all words in one line and all distances in the next, this can be adjusted by changing one line of the algorithm.

7. References

Reichel, U.D. (2012). PermA and Balloon: Tools for string alignment and text processing. In *Proceedings of Interspeech*.

Reichel, U.D., Kisler, T. (2014). Language-independent grapheme-phoneme conversion and word stress assignment as a web service. In Hoffmann, R. (Ed.): *Elektronische Sprachverarbeitung. Studientexte zur Sprachkommunikation 71* (pp 42-49). Dresden: TUDpress.