

# Inhaltsverzeichnis

- stylo ah online - Handbuch** ..... 2
- Ursprung: stylo ah online - stylo ohne Installation*** ..... 2
- Links zum Thema*** ..... 2
- Implementierte Analyse-Pipeline(s)*** ..... 2
- Funktionsüberblick*** ..... 2
- Benutzung*** ..... 3
- GUI Konzept ..... 3
- Überblick zur Benutzung ..... 4
- 1. Konfiguration des Browsers ..... 4
- 2. Konfigurieren der Analyseschritte ..... 4
- 2.1 Arbeitsschritt: Benennung ..... 4
- 2.2 Arbeitsschritt: Configurations-Datei / Datenbank ..... 5
- 2.3 Arbeitsschritt: Input / Replication ..... 5
- 2.4 Arbeitsschritt: Normalisierung (Normalization) ..... 6
- 2.5 Arbeitsschritt: Zerlegung (features / decomposition / token) ..... 7
- Arbeitsschritt: Zählung ..... 8
- Arbeitsschritt: Maanwendung ..... 8
- Arbeitsschritt: Gruppierung ..... 8
- Arbeitsschritt: Export ..... 8
- 3. Corpus-Auswahl ..... 8
- 4. Erneute Berechnung und Serien ..... 8

# stylo ah online - Handbuch

## Ursprung: stylo ah online - stylo ohne Installation

Im Rahmen unsere Lehrtätigkeit und unserer Forschungstätigkeit haben wir das R-Paket **stylo** eingesetzt. Wir haben dieses mittels R und Python weiterentwickelt. Vornehmlich vier Dinge sind an der R Implementierung schwierig:

1. Die Möglichkeit zur Selbstdokumentation der Experimente ist gering.
2. Die Interaktion zwischen Python und R ist problematisch.
3. Die Möglichkeit eines einfachen Einsatzes von Multiprocessing ist fraglich.
4. Die Software ist teilweise schwierig zu installieren.

Daher wollte wir die Funktionalität von **stylo** nachempfinden; ohne Installation und mit erweiterten Dokumentations- und Vergleichsmöglichkeiten ausstatten.

## Links zum Thema

1. [stylo](#) das Original in R geschrieben.
2. [stylo ah](#) die R / Python Kopie mit zusätzlichen Funktionen für klassische Texte.
3. [stylo ah online](#) die originale Kopie in JavaScript implementiert.

## Implementierte Analyse-Pipeline(s)

Die Implementierte Textanalyse verläuft in sieben Schritten. Diese stellen die Abfolge bzw. die Pipeline der Verarbeitung dar:

1. Auswahl der Files (Corpus) (jedes File repräsentiert einen Text)
2. Normalisierung (Formatanpassung, Zeichenvereinheitlichung, Löschung von Struktur und Metadaten, Maskierung von Wortformen)
3. Zerlegung in Token (Wortformen, Silben, grame, Zeichen)
4. Zählung / Vektorbildung (01-Kodierung, abs. / rel. Häufigkeit, TF-IDF)
5. Auswahl aus den Vektoren (Beschränkung der Häufigkeitsliste, Häufigkeitsfenster, Culling)
6. Anwendung eines Maßes
7. Anwendung einer Cluster-Methode

## Funktionsüberblick

In stylo ah online sind folgende Funktionen verfügbar:

### Normalisierung:

[Normalform](#) [Wortmaskierung \(Stopworte\)](#) [Vereinheitlichen bestimmter Zeichen](#) [UV-Angleich](#) [JI-Angleich](#) [Markup-Löschen](#) [Interpunktion löschen](#) [Zeilenumbrüche löschen](#) [Elision auflösen](#) [Alpha](#)

privativum behandeln Entfernen der Nummerierung Worttrennungen zusammenführen  
Vereinheitlichung des Iota subscriptum und Iota adscriptum Veränderung des Abschluss-Sigmas  
Diakritische Zeichen löschen Ligaturen auflösen Kleinschreibung Klammern entfernen

Normalisierung

### Zerlegung (Token):

Zusätzlich ohne Konsonanten Zusätzlich ohne Vokale Zusätzlich lediglich kleine Wörter Zusätzlich lediglich große Wörter Zerlegung in Wortformen Zerlegung in Buchstaben n-gramme Zerlegung in Buchstaben n-gramme der Wortformen Zerlegung in Wort n-gramme Zerlegung in n-gramme mit Lücken Zerlegung in Pseudo-Silben Zerlegung in Kopf Körper und Coda Zerlegungen in alle Permutationen von Kopf Körper und Coda

Zerlegung

### Zählung:

absolute Häufigkeit relative Häufigkeit 0-1-Codierung TF-IDF Beschränkung/Spannen der  
Frequenzlisten (nach Rang, min-max-Angabe) Culling

Zählung

### Maße:

euclidean, chebyshev, minkowski, manhattan, canberra, soerenen, gower, soergel, lorentzian,  
intersection, wavehedges, motyka, ruzicka, tanimoto, innerproduct, harmonicmean, cosine, kumar  
hasse brook, dice, fidelity, bhattacharya 1, bhattacharya 2, hellinger, jensen, jensen shannon,  
topsoee, kullback divergence, jeffreys, kullback leibler, squared euclidean, pearson chi squared,  
neyman chi squared, squared chi squared, divergence, clarck, additive symmetric chi squared, eder  
simple, burrows delta, argamon linear delta, eders delta, argamons quadratic delta, wasserstein 1d

Maße

### Gruppierung:

hierarchische Clusterung, multidimensional scaling (MDS), tSNR (t-distributed stochastic neighbor embedding)

Gruppierung

## Benutzung

### GUI Konzept

Jeder Verarbeitungsschritt wird in einem Konfigurationsschritt eingestellt. Jeder Konfigurationsschritt besitzt ein zusätzliches Kommentarfeld und eine der Anzeige der Zwischenergebnissen (rechte Spalte). Die GUI empfindet die Schritte der Textanalyse nach. Einzige Ausnahme bildet die Auswahl des Corpus, die ist zwar an oberer Stelle angezeigt, aber von der Benutzung her ist das der letzte Schritt! Jedes GUI-Element ist mit einer Beschriftung versehen. Sofern die Funktion einer weiteren Erklärung bedarf, dann ist zusätzlich eine kursive Erklärung angefügt.

# Überblick zur Benutzung

1. Konfiguration des Browsers
2. Konfigurieren der Analyseschritte
3. Corpus-Auswahl
4. Erneute Berechnung und Serien

## 1. Konfiguration des Browsers

Speicherort: Legen sie einen Ordner an in welchem die Ergebnisse von stylo-ah-online gespeichert werden können. Konfigurieren Sie ihren Browser so, dass er die Downloads in diesem Ordner ablegt. die Anleitung für Firefox:

[https://support.mozilla.org/de/kb/suchen-und-verwalten-heruntergeladener-dateien#w\\_ziel-ordner-der-heruntergeladenen-dateien-andern](https://support.mozilla.org/de/kb/suchen-und-verwalten-heruntergeladener-dateien#w_ziel-ordner-der-heruntergeladenen-dateien-andern)

Datenbanken: Teilen sie dem Browser mit, dass er Datenbanken anlegen und nicht löschen soll. Dazu stellen sie sicher, dass die Chronik angelegt wird:

<https://support.mozilla.org/de/kb/firefox-chronik-zeigt-ihre-besuchten-webseiten>

Web-Konsole: stylo-ah-online ist in JavaScript geschrieben. Die Web-Konsole bietet eine Darstellung von Meldungen zum Programmablauf und Hinweise auf Fehler. Nach einem Fehler können sie die Seite neu laden und den Ablauf erneut starten. Es bietet sich an die Web-Konsole anzuzeigen. Für Firefox aktiviert man diese so:

[https://firefox-source-docs.mozilla.org/devtools-user/tools\\_toolbox/index.html](https://firefox-source-docs.mozilla.org/devtools-user/tools_toolbox/index.html)

## 2. Konfigurieren der Analyseschritte

### 2.1 Arbeitsschritt: Benennung

#### Naming

experiment	<input type="text"/>	Type (Select/give a type, if this is not given by file ending.)
default	<input type="text"/>	Subject (Name the subject of the file.)
unfinished	<input type="text"/>	State (Select/give a state, if this is not given by file ending.)
<input type="text"/>	<input type="text"/>	ID (Give the ID of the file.)
2023-07-14	<input type="text"/>	Date (The actual date, erase to reset.)
<input type="text"/>	<input type="text"/>	Version (Choose a version of the file.)
default	<input type="text"/>	Author name (Fill in the name of the author.)
<input type="text"/>	<input type="text"/>	File ending (Provide a file ending.)
experiment_default_unfinished_2023-07-14_default.styloahonline		
<div style="border: 1px solid gray; height: 40px;"></div>		

(Log entry of naming section.)

Das Benennungsmodell beinhaltet zusammengesetzte Namen für die Experimente. Dabei werden die einzelnen Teile der Benennung durch Unterstriche abgetrennt. Leerzeichen in Einträgen werden durch Bindestriche ersetzt. Die einzelnen Einträge der Benennung erlauben verschiedene

Organisationsmuster für Experimente. Das Datum wird beim ersten Programmablauf automatisch ausgefüllt. Der einmal erstellte Name charakterisiert den Programmablauf und die Einstellungen genau. Änderungen in der Benennung führt auf ein neues Experiment, welches auch separat gespeichert wird. Die Benennung geht in die Config-Datei ein und ein erneuter Aufruf einer Config-Datei stellt alles unter dieser Bezeichnung wieder her. Auf diese Weise können mehrere Konfigurationen unterschieden werden.

## 2.2 Arbeitsschritt: Configurations-Datei / Datenbank

### Configuration

#### Config for text analysis

(This will run the analysis again, if you made changes to the settings below.)

Keine ausgewählt (Choose a existing stylo-online configuration file to set the configuration for stylo-online.)

#### GEN config for SERIAL text analysis

(This will generate config files for each token version (1-3 gram), but leaves the other configuration unchanged.)

(This will generate config files for each counting method, but leaves the other configuration unchanged.)

(This will generate config files for each measure, but leaves the other configuration unchanged.)

#### Config for stylo-ah-online display

Display size of results (Checked: Just show a sample of the results (1000 token/signs). Not checked: Results will be shown in full length.)

#### Delete

(This will delete the configuration.)

(This will delete the stored files and the results of the analysis.)

(This will reset stylo ah online to start an new analysis.)

Im Abschnitt „Configuration“ können alle Einstellungen vorgenommen werden, die sich auf die Config-Datei beziehen. stylo-ah-online erlaubt die Speicherung der Konfiguration und genauso ihren erneuten Aufruf durch Eingabe einer Config-Datei. Für die Erstellung von Serien von Experimenten erlaubt es stylo-ah-online mehrere Config-Dateien zu erstellen und als Serie wieder zu öffnen. Die Ergebnisse werden für jede so gespeicherte Einstellung unter einem extra Namen abgespeichert. Für umfangreichere Corpora ist es nötig die Darstellung von Zwischenergebnissen zu beschränken. Das kann an dieser Stelle mit dem Haken bei „Display size of results“ erzielt werden. stylo-ah-online verfügt über eine Auto-save-Funktion, außerdem werden die eingegebenen Daten gespeichert, um diese bei einem erneuten Analysedurchlauf aus der Browser eigenen Datenbank zu holen. Man kann hier die Einstellungen und die Datenbanken oder beides zurücksetzen.

## 2.3 Arbeitsschritt: Input / Replication

## Input / Replication

Keine ausgewählt *(Just choose the CORPUS FILES, than the selection below will be applied (start analysis). Data in the data base will be overwritten.)*

*(This will RUN the analysis AGAIN, if you made changes to the settings below. Data is taken from the data base.)*

Keine ausgewählt *(Choose MULTIPLE config files to perform multiple analysis on one corpus.)*

### Note

*(Log entry for the input section.)*

Im Abschnitt „input / Replication“ geht es um den Aufruf der Textdateien, die analysiert werden sollen. Diese Handlung löst die erste Analyse aus und speichert Ergebnisse in der Datenbank. Mit „rerun“ kann man eine veränderte Konfiguration auf die vorhandenen Daten anwenden. Sollten Serienexperimente vorbereitet wurden sein, dann kann man die dazugehörigen Config-Files hier eingeben und so die Serienverarbeitung auslösen. Dazu werden die Daten aus der Datenbank verwendet. Der Arbeitsschritt der Dateieingabe muss der Letzte sein, nachdem alle anderen Konfigurationen vorgenommen wurden.

## 2.4 Arbeitsschritt: Normalisierung (Normalization)

## NORMALIZATION

(Please check <http://ecomparatio.net/~khk/NORM-DECOMP-DIST/textnorm.html> to see some examples of how the selection would work.)

### Word masking / stop words

- Use Word masking (Give back the string without stop words.)
- (Check this to apply stop word removal.)
- Keine ausgewählt (Choose a existing stop word file (CSV format, divider: ;).)

### Sign equalization

- Disambiguate diacritica
- Disambiguate dashes
- Text output latin u-v (replaces all u with v)
- Text output latin j-i (replaces all j with i)
- Iota sub to ad (takes greek utf8 string and replaces iota subscriptum with iota ad scriptum)
- Text output tailing sigma uniform (equalize tailing sigma)
- Text output without diacritics (replaces diacritics)
- Text output without some signs (delete some to the programmer unknown signs: †, \*, <, #, §, ‡)
- Text output without ligature (takes a string, return string with ligatures turned to single letters)
- Text output equal case (input a string and get it back with all small case letters)
- Text output no brackets (input string and get it back with no brackets)

### Markup / Format

- Without markup (input a string and get it back with markup (html / xml) removed)
- Delete punctuation (takes string and returns the string without punctuation)
- Without newline (input string and get it back with linebreaks removed)

### Word level conversions

- Elision expansion (elusion it will be expanded)
- Alpha privativum / copulativum (takes utf8 greek and splits the alpha privativum and copulativum from wordforms)
- Text output without numbering (takes string, return string without the edition numbering i.e. [2])
- Text output no hyphenation (removes hyphenation)

### Combinations

(Select one of the combined normalization functions (none of the single steps is used).)

### Translitteration

(Select one of the transliterations.)

### Note

## 2.5 Arbeitsschritt: Zerlegung (features / decomposition / token)

## FEATURES / DECOMPOSITION / TOKEN

(The word level decomposition and the gram decomposition will be combined. Check <http://ecomparatio.net/~khk/NORM-DECOMP-DIST/zerl.html> for some examples to see how decomposition will work.)

### Word level decomposition

- None
- Without consonants (string without consonants)
- Without vowels (string without vowels)
- Small words (string with just small words (stopwords))
- Big words (string with just big words (not stopwords))

### General N-Gram decomposition

Word level (groups of words)  Gram-level

N (gram-size, set to one means for example word statistics)

M (gap-size, for gap n-gram)

Padding (used for sign level of words)

### Note

(An entry for the token section)

## Arbeitsschritt: Zählung

## Arbeitsschritt: Maanwendung

## Arbeitsschritt: Gruppierung

## Arbeitsschritt: Export

## 3. Corpus-Auswahl

1. Aufruf von

## 4. Erneute Berechnung und Serien

From: <http://replicatio.science/dokuwiki/> - documentatio replicationis

Permanent link: <http://replicatio.science/dokuwiki/doku.php/de/styloahonline/handbuch?rev=1699467742>

Last update: 2023-11-08

