

# Estrategias de comprensión con y para Python

Un acercamiento  
filológico



Martina Kienberger

# Contenidos

1. Python, un lenguaje de programación
2. Empezando a trabajar con Python
3. Mis estrategias de aprendizaje
4. Mi proyecto de investigación
5. Retos futuros

# 1. Python, un lenguaje de programación

# 1. Python, un lenguaje de programación

In [18]: `A2.ID_alumno.value_counts().sort_index()`

# 1. Python, un lenguaje de programación

In [18]: `A2.ID_alumno.value_counts().sort_index()`

variable definida  
anteriormente  
(datos de mi tabla  
Excel)

columna en la  
tabla Excel

**nombres propios**

# 1. Python, un lenguaje de programación

In [18]: `A2.ID_alumno.value_counts().sort_index()`

variable definida  
anteriormente  
(datos de mi tabla  
Excel)

columna en la  
tabla Excel

Instrucciones

**vocabulario**

**nombres propios**

# 1. Python, un lenguaje de programación

In [18]:

**A2.ID\_alumno.value\_counts().sort\_index()**

variable definida  
anteriormente  
(datos de mi tabla  
Excel)

columna en la  
tabla Excel

Instrucciones

**vocabulario**

**nombres propios**

Elementos de conexión y reglas para  
su uso, el orden de las partes

**gramática**

```
In [18]: A2.ID_alumno.value_counts().sort_index()
```

```
Out[18]: 1      27  
         2      13  
         3      10  
         4       4  
         5       5  
         6      20  
         7      19  
         8      19  
        10      10  
        11      10  
        12      20  
        13      13  
        14       7  
        15      14  
        16      16  
        17      11  
        18       9  
        19      15  
        20       9  
        21      17  
        22      10  
        23      13  
        24       9  
        25       5  
        26      20  
        27      14  
        28      12  
        29      20  
Name: ID_alumno, dtype: int64
```



# 1. Python, un lenguaje de programación

```
In [26]: for val in CTA2_2.index:
         for col in CTA2_2.columns:
             if CTA2_2.loc[val,col]!=0:
                 CTA2.loc[val,col] = CTA2.loc[val,col]+ CTA2_2.loc[val,col]
```

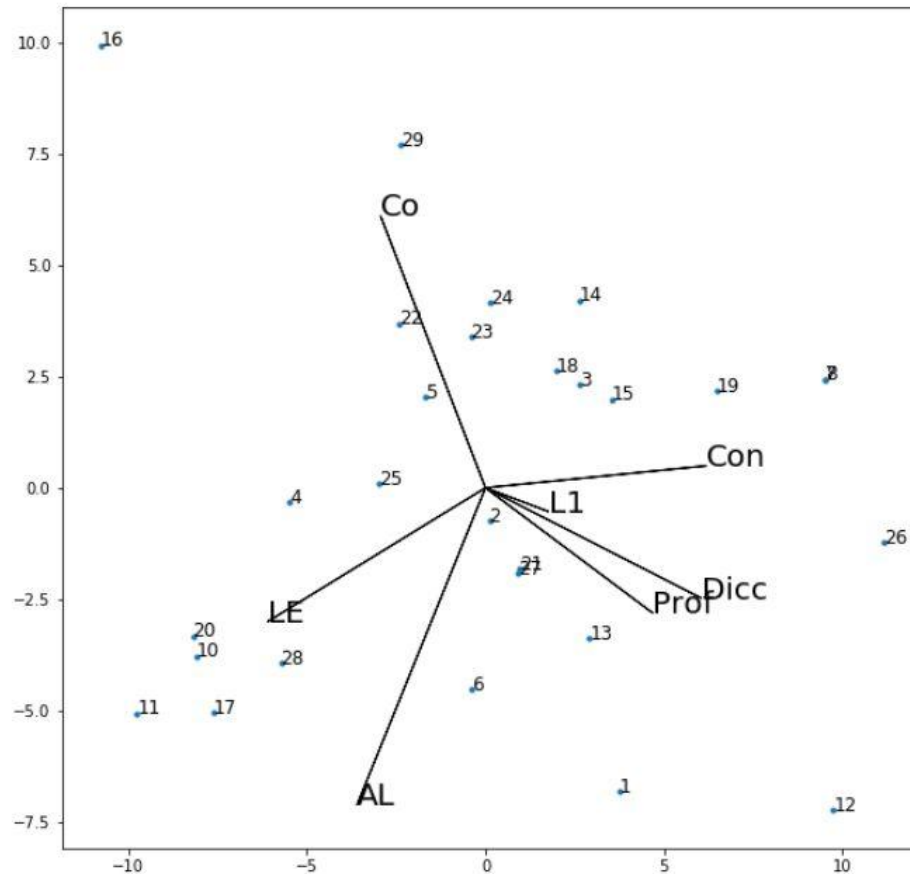
# 1. Python, un lenguaje de programación

```
In [26]: for val in CTA2_2.index:
          for col in CTA2_2.columns:
            if CTA2_2.loc[val,col] != 0:
              CTA2.loc[val,col] = CTA2.loc[val,col] + CTA2_2.loc[val,col]
```

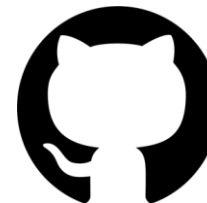
```
In [34]: fig = plt.figure(figsize=(10,10))
ax1 = fig.add_subplot(111)

ax1.scatter(R[:,0],R[:,1], s = 9)
for i in range(R.shape[0]):
    ax1.text(R[i,0],R[i,1], str(CTA2.index[:1][i]), fontsize=12 )
for i in range(C.shape[0]):
    ax1.arrow(0,0,C[i,0],C[i,1])
    #ax1.scatter(C[i,0],C[i,1])
    ax1.text(C[i,0],C[i,1],CTA2.columns[:1][i],fontsize=20)

plt.show()
```



## 2. Empezando a trabajar con Python



## 2. Empezando a trabajar con Python

- Instalación



## 2. Empezando a trabajar con Python

- Instalación



- Un entorno para programar





Logout

Files

Running

Clusters

Select items to perform actions on them.

Upload

New ▼



0



/

Name ▼

Last Modified



Anderes

vor ein paar Sekunden



Code

vor 15 Tagen



Code\_Stan

vor 8 Monaten



Code\_Voc

vor 3 Tagen



Daten

vor 5 Tagen

 jupyter **Untitled** Last Checkpoint: vor einer Minute (unsaved changes)



Logout

File

Edit

View

Insert

Cell

Kernel

Widgets

Help

Trusted

Python 3 



Code



In [1]: 1+1

Out[1]: 2

In [ ]:



jupyter Datenanalyse\_Voc\_Valencia Last Checkpoint: 18.09.2018 (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3

Run Stop Restart Clear All Cell Toolbar

## Vorbereitung - Datenreinigung

```
In [1]: import pandas as pd
import numpy as np
from biofes import biplot
from sklearn.utils.extmath import randomized_svd
from sklearn.cluster import KMeans, AgglomerativeClustering
from adjustText import adjust_text
% pylab inline
```

Populating the interactive namespace from numpy and matplotlib

```
In [2]: Rohdaten = pd.read_excel('D:\Daten\DISS\Daten\Valencia.xlsx')
```

```
In [3]: Rohdaten.head(3)
```

Out[3]:

	Grupo	ID_alumno	Encuesta	Palabra	Estrategia_1	Estrategia_2	Estrategia_Esp_1	Estrategia_Esp_2	Comentario
0	A2	1	1	lebendiger	AL	NaN	Comp	NaN	NaN
1	A2	1	1	langen	Con	NaN	NaN	NaN	NaN
2	A2	1	1	eigene	Dicc	NaN	NaN	NaN	NaN

jupyter Datenanalyse\_Voc\_Valencia Last Checkpoint: 18.09.2018 (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

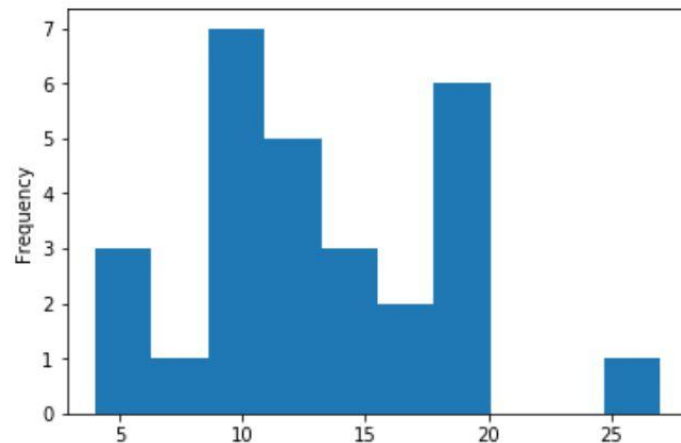
Trusted

Python 3



```
28      12
29      20
Name: ID_alumno, dtype: int64
```

```
In [19]: A2.ID_alumno.value_counts().plot(kind = 'hist',
                                             bins = 10);
```



Das Histogramm zeigt die Frequenz der notierten Wortanzahl pro TN - keine Normalverteilung in diesem Fall.

## 2. Empezando a trabajar con Python

- Instalación



- Un entorno para programar



- Guardar y compartir los resultados





Search or jump to...



Pull requests

Issues

Marketplace

Explore



**Martina  
Kienberger**

[martinakienberger](#)

Teacher and researcher at the  
University of Salamanca, phd student  
at the University of Viena

Edit bio

Salamanca

<http://diarium.usal.es/martinaki...>

Overview

Repositories 2

Stars 0

Followers 1

Following 0

Popular repositories

Customize your pinned repositories

**Pilotstudie\_SP17**

Research project

Jupyter Notebook

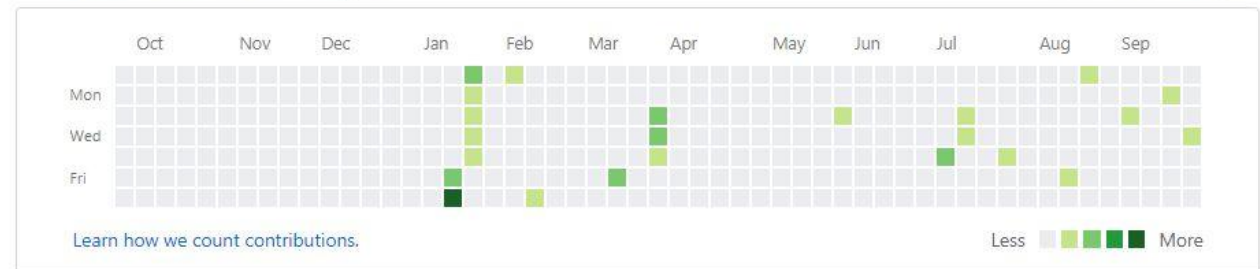
**Strategienanalyse\_Voc**

Research project

Jupyter Notebook

39 contributions in the last year

Contribution settings



Contribution activity

Jump to

2018

September 2018



Created 3 commits in 2 repositories

[martinakienberger/Strategienanalyse\\_Voc](#) 2 commits



Branch: master ▾

Strategienanalyse\_Voc / Datenanalyse\_Voc\_Valencia.ipynb

Find file

Copy path



martinakienberger Wortlisten, Details Granada

6b9f198 3 days ago

1 contributor

4629 lines (4628 sloc) | 596 KB



Raw

Blame

History



## Analyse des Einsatzes von Erschließungsstrategien beim Lesen

Am 2. 3. 2018 wurden an der Universität Valencia in zwei DaF-Gruppen (A2, B2) Daten zum wahrgenommenen Strategieneinsatz der Lernenden bei der Bearbeitung einer Aufgabe zum Lesen und Verstehen des neuen Wortschatzes erhoben. Die Studierenden wurden gebeten, beim Lesen eines Textes aus dem Lehrwerk zunächst alle für sie neuen Wörter zu markieren (unabhängig von deren Verständnis). Nach einem zweiten Lesedurchgang sollten sie diese in eine Liste eintragen und angeben, welche Strategien sie angewendet hätten, um deren Bedeutung zu erkennen. Ein Teil der Lernenden hatte zuvor (freiwillig) an der Befragung zum Thema Erschließungsstrategien für unbekannten Wortschatz im Deutschen durch Lernende an spanischen Universitäten teilgenommen (siehe: <http://diarium.usal.es/martinakienberger/200-2/?lang=de>). Für alle Studierenden wurde kurz das Thema erklärt und Beispiele für Strategien gegeben.

### Vorbereitung - Datenreinigung

```
In [1]: import pandas as pd
import numpy as np
from biofes import biplot
from sklearn.utils.extmath import randomized_svd
from sklearn.cluster import KMeans, AgglomerativeClustering
from adjustText import adjust_text
% pylab inline
```

Populating the interactive namespace from numpy and matplotlib

```
In [2]: Rohdaten = pd.read_excel('D:\Daten\DISS\Daten\Valencia.xlsx')
```

```
In [3]: Rohdaten.head(3)
```

## 2. Empezando a trabajar con Python

- Instalación



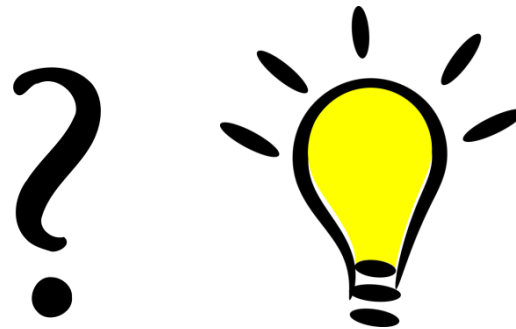
- Un entorno para programar



- Guardar y compartir los resultados



### 3. Mis estrategias de aprendizaje





### 3. Mis estrategias de aprendizaje

- ✓ Aprender lo básico con un libro de introducción
- ✓ Utilizar recursos online:
  - Páginas web de apoyo para principiantes
  - Documentación de las librerías
  - Foros (Stack Overflow)
- ✓ Ver ejemplos
- ✓ Aprender de mis errores



## Vorbereitung - Datenreinigung

```
In [1]: import pandas as pd
import numpy as np
from biofes import biplot
from sklearn.utils.extmath import randomized_svd
from sklearn.cluster import KMeans, AgglomerativeClustering
from adjustText import adjust_text
% pylab inline
```

Populating the interactive namespace from numpy and matplotlib

```
In [2]: Rohdaten = pd.read_excel('D:\Daten\DISS\Daten\Valencia.xlsx')
```

```
In [1]: Rohdaten.head(3)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-1-c9831417dbd0> in <module>()
----> 1 Rohdaten.head(3)

NameError: name 'Rohdaten' is not defined
```

```
In [5]: Rohdaten.head.
```

```
File "<ipython-input-5-e52ee1ca942f>", line 1
    Rohdaten.head.
            ^
SyntaxError: invalid syntax
```

### 3. Mis estrategias de aprendizaje

- ✓ Aprender lo básico con un libro de introducción
- ✓ Utilizar recursos online:
  - Páginas web de apoyo para principiantes
  - Documentación de las librerías
  - Foros (Stack Overflow)
- ✓ Ver ejemplos
- ✓ Aprender de mis errores
- ✓ Empezar con lo básico + variar
- ✓ Buscar ayuda, preguntar

## 4. Mi proyecto de investigación



**Estrategias de comprensión  
para palabras desconocidas  
en alemán**

Descubro el significado o la función de palabras nuevas en alemán gracias a mi lengua materna. (L1)

Descubro el significado o la función de palabras nuevas en alemán con la ayuda de otras lenguas. (LE)

Utilizo mis conocimientos de alemán, p. ej. otras palabras conocidas, clases de palabras o la formación de palabras compuestas. (AL)

# Nach Adelboden zur digitalen Entgiftung

**BERN.** Flugmodus für die Seele: Adelboden Tourismus eröffnet das erste Digital Detox Camp der Schweiz. Vier Tage lang werden die Besucher von Pushs und Whatsapp-Nachrichten abgeschirmt. Anstatt Surfen im Internet stehen Yoga und Entspannungsübungen auf dem Programm. Selfiesticks haben dabei nichts verloren, dafür werden in der Berg-hütte Freelax in Tronegg Bleistifte und Notizbücher verteilt. «Bei uns auf der Alp kann man Antworten auf die tägliche Hektik finden», sagt Tourismusdirektor Urs Pfenninger. Die Teilnehmer sollen in erster Linie

wieder lernen, offline zu sein. Für diesen «kalten Entzug» stehen den Gästen Coaches zur Seite, die Teilnehmer auf eine Sinnesreise mitnehmen.

Doch das Camp vom 7. bis 10. September ist erst ein Experiment. Deshalb sucht Adelboden Tourismus derzeit nach fünf Freiwilligen, die sich dieser Challenge stellen. **MIW**



Yoga statt Whatsapp in den Bergen. DIGITALDETOX

Utilizo información extraída del contexto, p. ej. la posición de una palabra en la frase o el formato del texto. (Con)

Pido ayuda a mi profesor/a. (Prof)

Pido ayuda a mis compañeros/as de clase. (Co)

Consulto un diccionario u otra fuente (en papel o digital). (Dicc)

## 4. Mi proyecto de investigación

### ✓ El estudio:

- 94 estudiantes de 5 cursos de alemán  
de 3 universidades españolas (Valencia, Salamanca, Granada)
- Hojas de trabajo: palabras + estrategias (texto libre)

### ✓ Datos → Excel



2. 3. 2018, Valencia

## Estrategias para comprender palabras desconocidas en alemán - un ejercicio

- Vas a leer un texto.
- Al leerlo por primera vez marca todas las palabras que no conoces aún (independientemente de si las entiendes por el contexto o no).
- Después, lee el texto por segunda vez y busca el significado de las palabras desconocidas. Seguramente lograrás entender algunas de estas palabras nuevas enseguida, otras después de pensar un poco. Si no encuentras el significado sin ayuda, puedes usar un diccionario o preguntar a otra persona.
- A continuación, apunta en esta hoja las palabras nuevas y las estrategias que has usado para averiguar su significado. También puedes indicar elementos del texto que han facilitado la comprensión de una palabra para ti.

[illegible]

	A	B	C	D	E	F	G	H	I
1	Grupo	ID_alumno	Encuesta	Palabra	Estrategia_1	Estrategia_2	Estrategia_Esp_1	Estrategia_Esp_2	Comentario
2	A2	1	1	lebendiger	AL		Comp		
3	A2	1	1	langen	Con				
4	A2	1	1	eigene	Dicc				
5	A2	1	1	Lieder	AL				
6	A2	1	1	verwenden	Dicc				
7	A2	1	1	alltägliches	AL		Comp		
8	A2	1	1	Kommunikationsmittel	AL		Comp		
9	A2	1	1	hört	AL				
10	A2	1	1	besonders	Dicc				
11	A2	1	1	Färbung	Dicc				
12	A2	1	1	meistens	Dicc				
13	A2	1	1	klingt	Con				
14	A2	1	1	sondern	Dicc				
15	A2	1	1	Stadtsparkasse	Dicc				
16	A2	1	1	gegründet	Con				
17	A2	1	1	Geschenk	Dicc				
18	A2	1	1	bietet	Dicc				
19	A2	1	1	liere					
20	A2	1	1	echte	Dicc				
21	A2	1	1	Abschlusstest	Dicc				
22	A2	1	1	Zugezogene	Dicc				
23	A2	1	1	Kölsch-Abitur	Dicc				
24	A2	1	1	Außerdem	AL		Comp		
25	A2	1	1	Bibliothek	L1				
26	A2	1	1	Büchern	AL		Comp		
27	A2	1	1	Sammlung	Dicc				
28	A2	1	1	Bildern	Dicc				
29	A2	2	1	Kölsch	Con				
30	A2	2	1	lebendiger	Dicc				
31	A2	2	1	eigene	Con				

## 4. Mi proyecto de investigación

- ✓ El estudio:
  - 94 estudiantes de alemán de 5 cursos/asignaturas de lengua de 3 universidades españolas
  - Hojas de trabajo: palabras + estrategias (texto libre)
- ✓ Datos → Excel
- ✓ Importar datos en Jupyter
- ✓ Análisis
- ✓ Visualización



jupyter Datenanalyse\_Voc\_Valencia Last Checkpoint: 18.09.2018 (autosaved)



Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted

Python 3

Run Stop Restart Clear All

Markdown

## Vorbereitung - Datenreinigung

```
In [1]: import pandas as pd
import numpy as np
from biofes import biplot
from sklearn.utils.extmath import randomized_svd
from sklearn.cluster import KMeans, AgglomerativeClustering
from adjustText import adjust_text
% pylab inline
```

Populating the interactive namespace from numpy and matplotlib

```
In [2]: Rohdaten = pd.read_excel('D:\Daten\DISS\Daten\Valencia.xlsx')
```

```
In [3]: Rohdaten.head(3)
```

Out[3]:

	Grupo	ID_alumno	Encuesta	Palabra	Estrategia_1	Estrategia_2	Estrategia_Esp_1	Estrategia_Esp_2	Comentario
0	A2	1	1	lebendiger	AL	NaN	Comp	NaN	NaN
1	A2	1	1	langen	Con	NaN	NaN	NaN	NaN
2	A2	1	1	eigene	Dicc	NaN	NaN	NaN	NaN

## Entfernung ungeeigneter Antworten

```
In [4]: Rohdaten.loc[117]
```

```
Out[4]: Grupo                A2  
ID_alumno                  9  
Encuesta                   2  
Palabra                   Imis  
Estrategia_1               Con  
Estrategia_2               NaN  
Estrategia_Esp_1    L1 = Deutsch!  
Estrategia_Esp_2               NaN  
Comentario                 NaN  
Name: 117, dtype: object
```

```
In [5]: Daten = Rohdaten.drop([117,372])
```

Als ungeeignet wurden ID 9 und ID 30 von der Analyse ausgeschlossen: ID 9 war Deutsch-Erstsprecher. ID 30 hatte die Aufgabe falsch verstanden und keine Strategien (sondern Übersetzungen) angegeben.

## Auswahl der für die Analyse relevanten Teile der Erhebung

```
In [6]: cols = ['Grupo', 'ID_alumno', 'Encuesta', 'Palabra', 'Estrategia_1', 'Estrategia_2']
```

```
In [7]: data = Daten[cols]
```

```
In [8]: data.loc[118]
```

```
Out[8]: Grupo                A2  
ID_alumno                  10  
Encuesta                   2  
Palabra    lebendiger  
Estrategia_1          Dicc  
Estrategia_2          NaN  
Name: 118, dtype: object
```

## Analyse des Strategieneinsatzes

### *Unterteilung der beiden Gruppen*

```
In [9]: A2 = data[data.Grupo == 'A2']
```

```
In [10]: B2 = data[data.Grupo == 'B2']
```

### Gruppe A2

```
In [11]: print(r'Teilnehmer: ' + str(len(A2.ID_alumno.unique()))+'\n'+  
              r'TN + Online-Befragung: ' + str(len(A2[A2.Encuesta == 1].ID_alumno.unique()))+'\n'+  
              r'TN - Online-Befragung: ' + str(len(A2[A2.Encuesta == 2].ID_alumno.unique()))+'\n'+  
              r'TN ? Online-Befragung: ' + str(len(A2[A2.Encuesta == 0].ID_alumno.unique())))
```

```
Teilnehmer: 28  
TN + Online-Befragung: 6  
TN - Online-Befragung: 18  
TN ? Online-Befragung: 4
```

Die Gruppe besteht aus 28 Teilnehmern. 6 hatten an der Online-Befragung teilgenommen, 18 nicht, 4 hatten keine Angabe dazu gemacht.

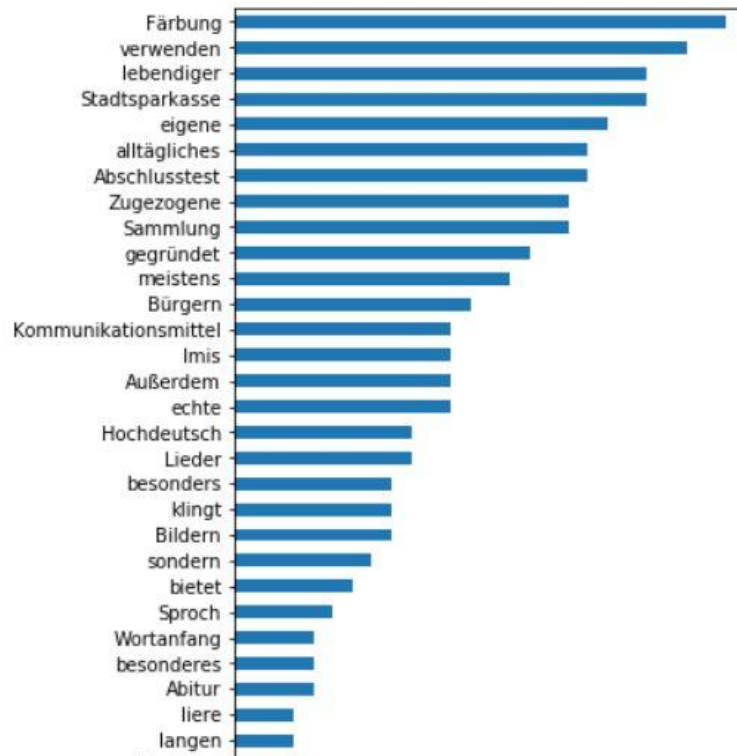
### Als "neu" identifizierte Wörter (A2)

```
In [12]: len(A2.Palabra.unique())
```

```
Out[12]: 47
```

47 unterschiedliche Wörter wurden von den Lernenden als für sie "neu" angegeben. Bei der Datenübertragung von den Aufgabenblättern in Excel wurden orthographische Fehler berichtigt. Die Wörter wurden in der jeweiligen Schreibung des Originaltextes aus dem Lehrbuch übernommen (auch wenn die Lernenden teilweise stattdessen eine andere Form, z.B. Infinitiv, angegeben hatten). Das Wort "bietet ... an" wurde zweimal aufgenommen, einmal als "bietet" und einmal als "bietet an", je nach Angabe der Lernenden, die das Wort offenbar teilweise nicht als trennbares Verb erkannt hatten.

```
In [13]: A2.Palabra.value_counts(ascending = True).plot(kind = 'barh', figsize = (5,12));
```



### Kreuztabelle Studenten - neue Wörter (A2)

```
In [14]: Al_Pal_A2 = pd.crosstab(A2.ID_alumno, A2.Palabra, margins = True)
```

```
In [15]: Al_Pal_A2.T.head()
```

Out[15]:

ID_alumno	1	2	3	4	5	6	7	8	10	11	...	21	22	23	24	25	26	27	28	29	All
Palabra																					
Abitur	0	0	0	0	0	0	1	1	0	0	...	0	0	0	0	0	0	0	0	1	4
Abschlusstest	1	1	1	0	0	1	0	1	1	1	...	1	1	1	0	0	1	1	1	1	18
Akademie	0	0	0	0	0	0	0	0	0	0	...	0	0	0	1	0	0	0	0	0	1
Außerdem	1	0	0	0	0	1	0	1	0	0	...	0	1	1	0	0	0	0	0	1	11
Bibliothek	1	0	1	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	2

5 rows × 29 columns

```
In [16]: print(r'Durchschnittlich als "neu" identifizierte Wortanzahl: '+str(Al_Pal_A2['All'][: -1].mean()))
```

Durchschnittlich als "neu" identifizierte Wortanzahl: 13.25

```
In [17]: print(r'Minimum der als "neu" identifizierten Wörter: '+str(A2.ID_alumno.value_counts().min()) + '\n' +
            r'Maximum der als "neu" identifizierten Wörter: '+str(A2.ID_alumno.value_counts().max()))
```

Minimum der als "neu" identifizierten Wörter: 4

Maximum der als "neu" identifizierten Wörter: 27

Durchschnittlich wurden ca. 13 Wörter als "neu" identifiziert. Große Schwankungsbreite: zwischen 4 und 27 (siehe oben).

## Einsatz von Erschließungsstrategien (A2) [...]

```
In [20]: v1 = A2.Estrategia_1.value_counts()
v2 = A2.Estrategia_2.value_counts()
```

```
In [21]: v = v1 + v2
v['L1'] = v1['L1']
v['LE'] = v1['LE']
v['Co'] = v1['Co']
v
```

```
Out[21]: AL      75.0
Co       31.0
Con      62.0
Dicc     177.0
L1       13.0
LE       11.0
Prof      5.0
dtype: float64
```

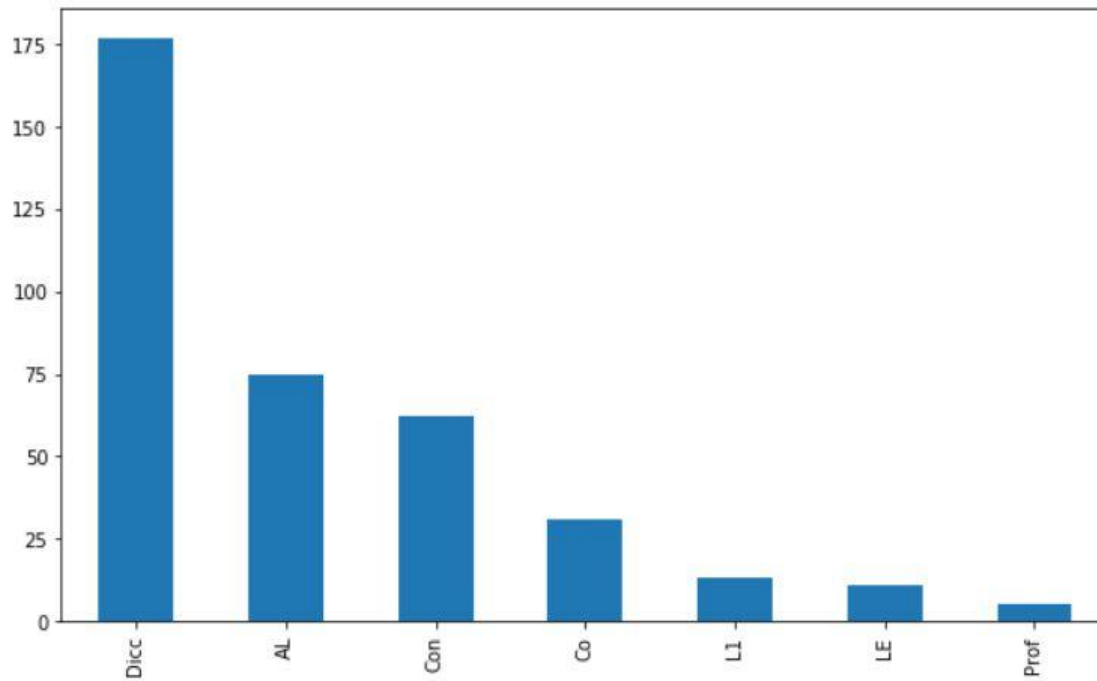
```
In [22]: v.sort_values(ascending = False)
```

```
Out[22]: Dicc     177.0
AL       75.0
Con      62.0
Co       31.0
L1       13.0
LE       11.0
Prof      5.0
dtype: float64
```

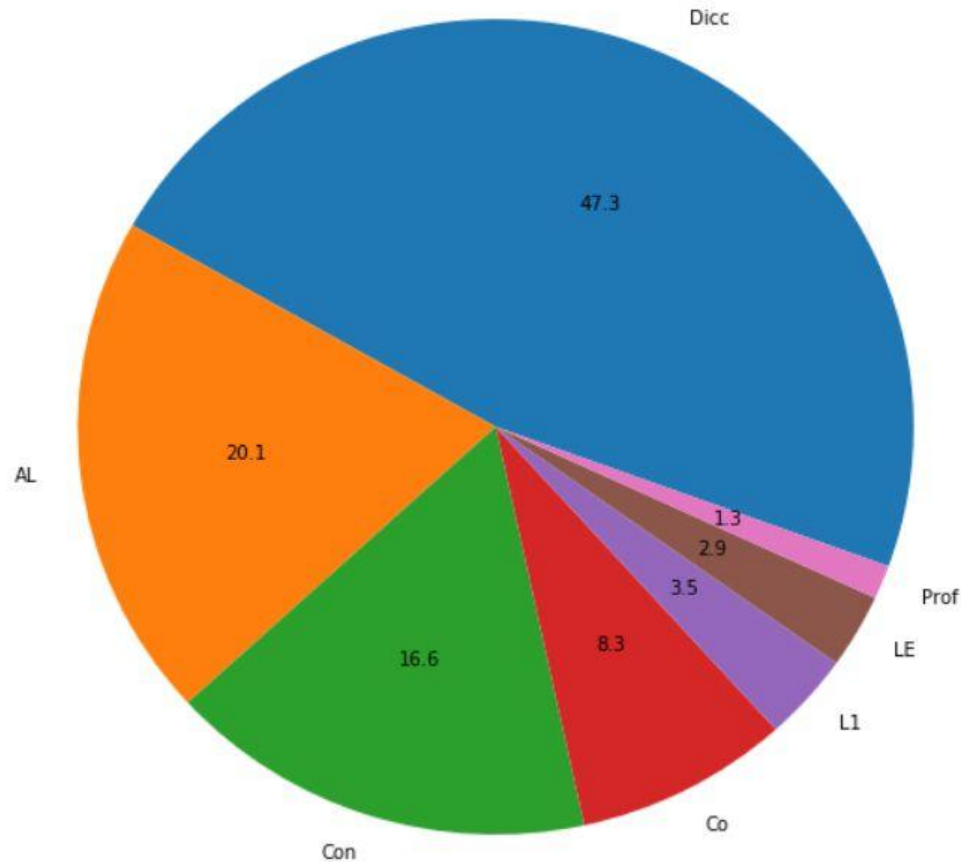
Da die Angaben einiger Studierender zwei Strategien zugeordnet werden können, müssen die Nennungen für die Gesamtauswertung zusammengezählt werden.



```
In [23]: v.sort_values(ascending = False).plot(kind = 'bar',  
            #title = 'Angaben zu eingesetzten Strategien',  
            figsize = (10,6));
```



```
In [24]: ax = v.sort_values(ascending = False).plot(kind = 'pie',  
#title = 'Angaben zu eingesetzten Strategien',  
figsize = (10,10),  
autopct='%.1f', startangle = -20);  
  
ax.set_ylabel("");
```





### Kreuztabelle Studenten - verwendete Strategien (A2)

```
In [25]: CTA2 = pd.crosstab(A2.ID_alumno, A2.Estrategia_1, margins = True)

CTA2_2 = pd.crosstab(A2.ID_alumno, A2.Estrategia_2, margins = True)
```

```
In [26]: for val in CTA2_2.index:
          for col in CTA2_2.columns:
            if CTA2_2.loc[val,col]!=0:
              CTA2.loc[val,col] = CTA2.loc[val,col]+ CTA2_2.loc[val,col]
```

```
In [27]: CTA2
```

```
Out[27]:
```

Estrategia_1	AL	Co	Con	Dicc	L1	LE	Prof	All
ID_alumno								
1	7	0	3	15	1	0	0	26
2	3	0	6	2	1	1	0	13
3	1	0	3	5	1	0	0	10
4	4	0	0	0	0	0	0	4
5	2	0	2	1	0	0	0	5
6	6	1	0	11	2	0	0	20
7	0	0	6	13	0	0	0	19
8	0	0	6	13	0	0	0	19
10	4	0	2	3	0	2	0	11
11	5	0	1	3	0	2	0	11
12	5	0	4	10	0	0	2	21
13	4	0	2	6	0	0	1	13
14	0	0	4	3	0	0	0	7

## Korrespondenzanalyse und Biplot Studenten - verwendete Strategien (A2)

```
In [28]: CA_A2L = biplot.CA(CTA2.values[:-1,:-1], 7, method = 1)
```

Mit den Daten der Kreuztabelle kann eine Korrespondenzanalyse durchgeführt werden, um die Beziehungen zwischen Elementen und Variablen grafisch repräsentieren zu können.

Eine Alternative stellt der Biplot dar, der besser geeignet ist, um Dimensionen der Darstellung zu reduzieren.

```
In [29]: Biplot_A2L = biplot.Classic(CTA2.values[:-1,:-1], 7, method = 1)
```

```
In [30]: R = Biplot_A2L.RowCoord
C = Biplot_A2L.ColCoord
cr = Biplot_A2L.RowCont
cc = Biplot_A2L.ColCont
```

## Qualität der Repräsentation der Elemente und Variablen auf den Achsen des Biplots

```
In [31]: Biplot_A2L.Inert
```

```
Out[31]: array([ 27.4862713 ,  19.30682653,  16.16280832,  13.94912583,
                9.54467983,   7.89842503,   5.65186318])
```

Überblick über Informationsgehalt der einzelnen Achsen des Biplot (hier 7)

```
In [32]: pd.DataFrame(cr, columns = ['Axis_'+str(el+1) for el in range(7)], index = CTA2.index[:-1])
```

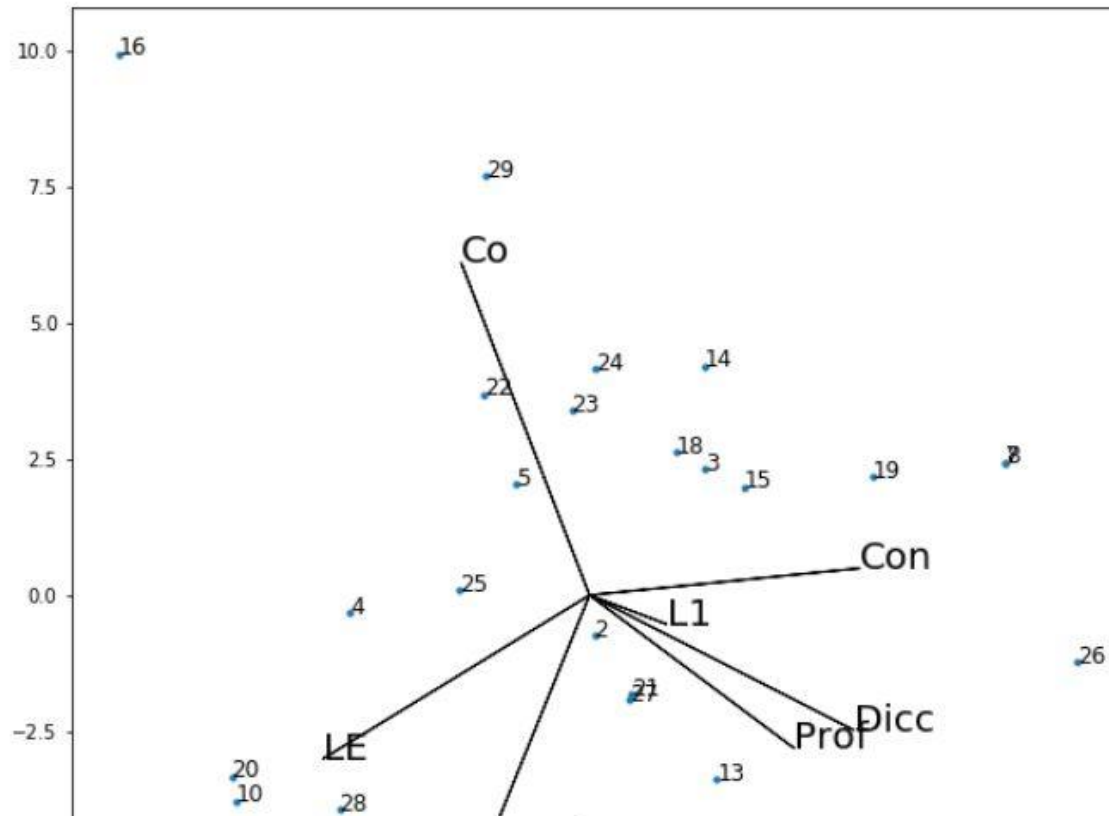
```
Out[32]:
```

	Axis_1	Axis_2	Axis_3	Axis_4	Axis_5	Axis_6	Axis_7
ID_alumno							
1	7.812657	36.536490	6.926975	0.850467	27.001013	5.564122	15.308276
2	0.010366	0.699793	3.234423	45.925108	7.964813	14.725019	27.440478
3	18.657148	20.652116	16.126947	36.177166	7.638956	0.341211	0.406455
4	34.195345	0.177427	0.699517	0.075264	21.827818	41.760687	1.263942

```
In [34]: fig = plt.figure(figsize=(10,10))
ax1 = fig.add_subplot(111)

ax1.scatter(R[:,0],R[:,1], s = 9)
for i in range(R.shape[0]):
    ax1.text(R[i,0],R[i,1], str(CTA2.index[: -1][i]), fontsize=12 )
for i in range(C.shape[0]):
    ax1.arrow(0,0,C[i,0],C[i,1])
    #ax1.scatter(C[i,0],C[i,1])
    ax1.text(C[i,0],C[i,1],CTA2.columns[: -1][i],fontsize=20)

plt.show()
```



## Cluster-Analyse Studenten - verwendete Strategien (A2)

Mit den Werten der Kreuztabelle können auch Cluster berechnet werden, um Gruppen differenzieren zu können.

Diese können in der Folge in einer Grafik mit den Ergebnissen des Biplots dargestellt werden.

```
In [35]: #X = CTA2.values[:,-1,:-1]
```

```
In [36]: #kmeans = KMeans(n_clusters=5, random_state=0).fit(X)
ward = AgglomerativeClustering(n_clusters=5).fit(CTA2.values[:,-1,:-1])
```

Beide Berechnungsarten führen zu ähnlichen Ergebnissen, in der Folge wird "ward" verwendet.

Tests mit unterschiedlicher Anzahl an Clustern zeigen, dass 5 sinnvolle Ergebnisse liefert.

## Darstellung Biplot + Cluster

```
In [37]: fig = plt.figure(figsize=(10,10))
ax = fig.add_subplot(111)

for i in range(C.shape[0]):
    ax.arrow(0,0,C[i,0],C[i,1], alpha = 0.7)
    #ax.scatter(C[i,0],C[i,1]) # Hier würden Linien nicht angezeigt.
    ax.text(C[i,0],C[i,1],CTA2.columns[:,-1][i],fontsize=20, alpha = 0.7)

ax.scatter(R[:,0],R[:,1], s = 12, c = ward.labels_)
texts = [plt.text(R[i,0],R[i,1], CTA2.index[:,-1][i], ha='center', va='center') for i in range(len(R[:,0]))]
adjust_text(texts);
```

```
In [38]: def vector_to_shape(v):
          markers = [",", "o", "v", "^", "x", "D", "*"]
          return [markers[el] for el in v]
```

```
In [39]: def vector_to_color(v):
          col = ['b', 'g', 'r', 'c', 'm', 'k', 'y']
          return [col[el] for el in v]
```

```
In [40]: def graf_cplot(data, dim, nclust, dim1 = 0, dim2 = 1, sx = 10, sy = 10):

          B = biplot.Classic(data, dim, method = 1)
          R = B.RowCoord
          C = B.ColCoord

          ward = AgglomerativeClustering(n_clusters=nclust).fit(data)

          fig = plt.figure(figsize=(sx,sy))
          ax = fig.add_subplot(111)

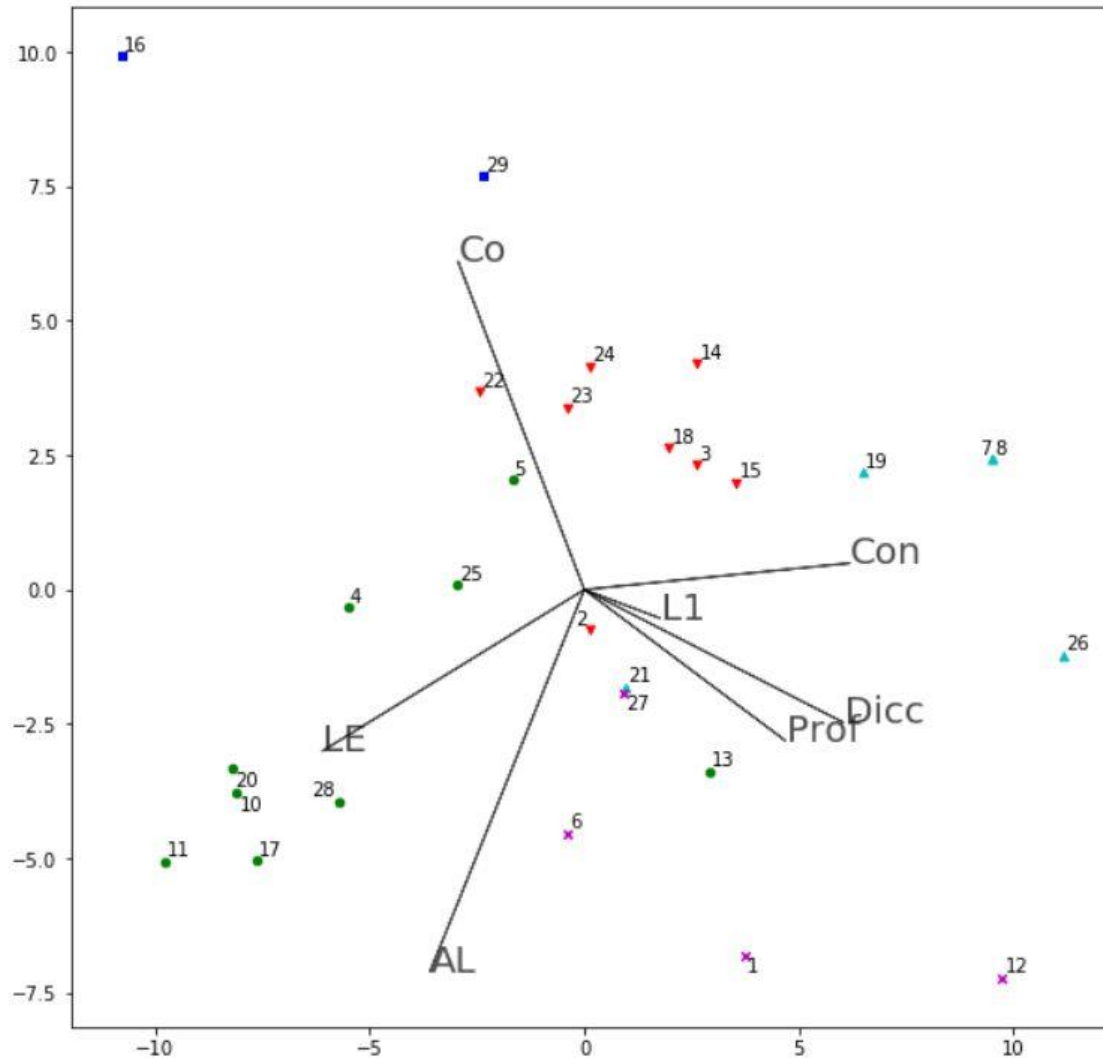
          sh_v = vector_to_shape(ward.labels_)
          color_v = vector_to_color(ward.labels_)

          for i in range(C.shape[0]):
              ax.arrow(0,0,C[i,dim1],C[i,dim2], alpha = 0.7)
              #ax.scatter(C[i,dim1],C[i,dim2]) # Hier würden Linien nicht angezeigt.
              ax.text(C[i,dim1],C[i,dim2],data.columns[i],fontsize=20, alpha = 0.7)

          for i in range(R.shape[0]):
              ax.scatter(R[i,dim1],R[i,dim2], s = 20, c = color_v[i] ,marker = sh_v[i])

          texts = [plt.text(R[i,dim1],R[i,dim2], data.index[i], ha='center', va='center') for i in range(len(R
         [:,0]))]
          adjust_text(texts);
```

In [42]: graf\_cplot(CTA2.iloc[:,-1], 7, 5)





## Kreuztabelle Wörter - verwendete Strategien (A2)

```
In [43]: Pal_Es_A2 = pd.crosstab(A2.Palabra, A2.Estrategia_1, margins = True)
Pal_Es_A2_2 = pd.crosstab(A2.Palabra, A2.Estrategia_2, margins = True)
```

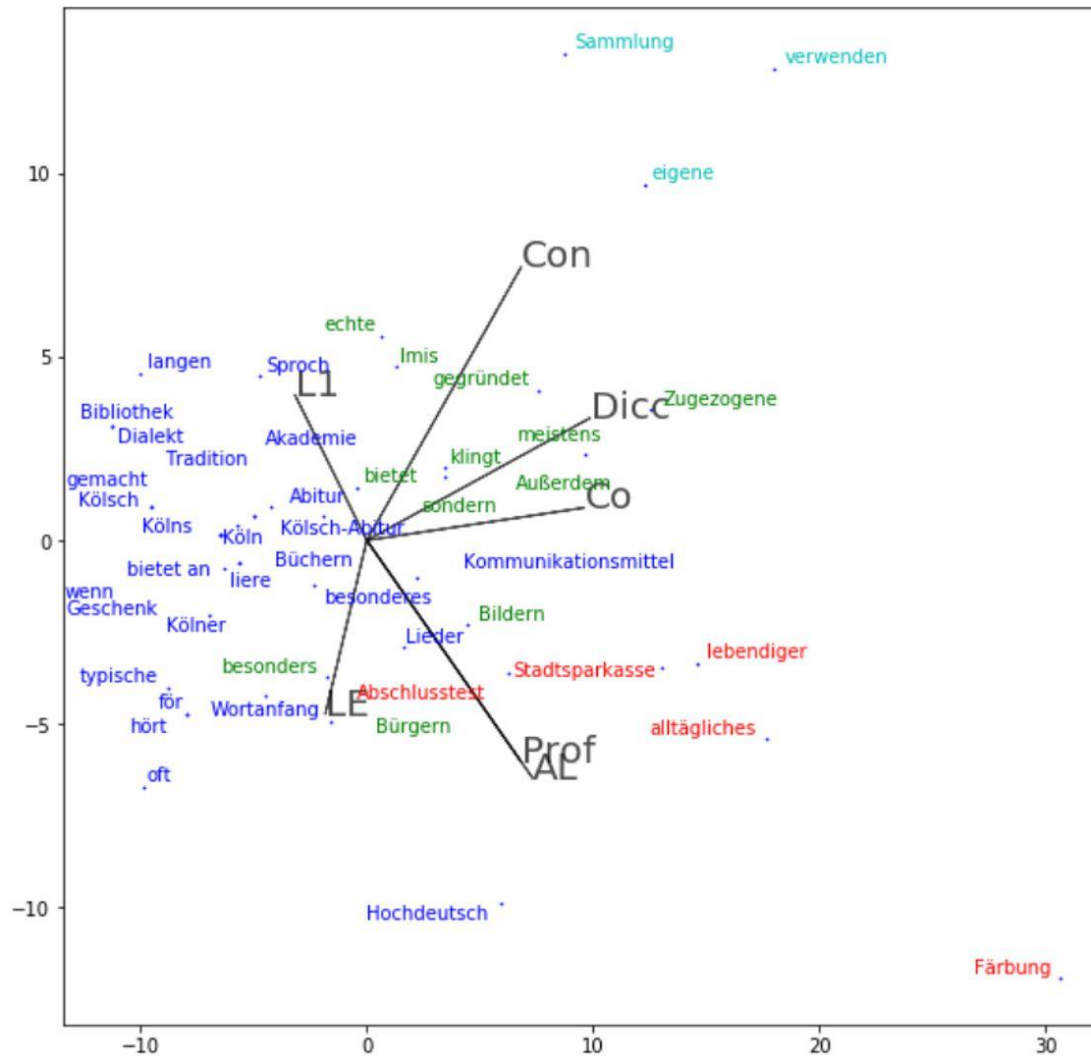
```
In [44]: for val in Pal_Es_A2_2.index:
          for col in Pal_Es_A2_2.columns:
              if Pal_Es_A2_2.loc[val,col]!=0:
                  Pal_Es_A2.loc[val,col] = Pal_Es_A2.loc[val,col] + Pal_Es_A2_2.loc[val,col]
```

```
In [45]: Pal_Es_A2.sort_values(['All'], ascending=[0])
```

Out[45]:

Estrategia_1	AL	Co	Con	Dicc	L1	LE	Prof	All
Palabra								
All	75	31	62	177	13	11	5	374
Färbung	9	3	2	10	0	0	2	26
Stadtsparkasse	11	2	3	6	1	1	0	24
verwenden	0	2	7	14	0	0	0	23
lebendiger	9	2	1	10	0	0	0	22
eigene	0	1	5	14	0	0	0	20
alltägliches	7	2	3	5	0	0	1	18
Abschlusstest	7	0	0	11	0	0	0	18
Sammlung	0	1	7	8	1	0	0	17
gegründet	2	1	3	9	0	0	0	15
Zugezogene	0	4	1	9	0	0	0	14
meistens	0	3	0	11	0	0	0	14
Bürgern	1	0	0	10	0	2	0	13

```
In [50]: graf_cplot2(Pal_Es_A2.iloc[:,-1],7, 4)
```





## 5. Retos futuros

*Análisis de texto libre*

**Métodos de visualización  
avanzados (3D, interactivos...)**

**NLP**

Seguir aprendiendo...

## Referencias

Morais, M. & Pillai, S. R. (2017). Data Analysis for Social Science and Marketing Research Using Python: A Non-Programmer's Guide. Aspire Analytic Solutions.

Caren, N. (2018). Learning Python for Social Scientists. En:

<https://nealcaren.github.io/python-tutorials/> (inglés)

Klein, B. (2017). Python-Kurs. En: [www.python-kurs.eu/index.php](http://www.python-kurs.eu/index.php) (alemán)

Muller, R. (2018). A Crash Course in Python for Scientists. En:

<http://nbviewer.jupyter.org/gist/rpmuller/5920182> (inglés)

### Contacto:

`martina.kienberger@usal.es`

<http://diarium.usal.es/martinakienberger/>

<https://github.com/martinakienberger>