Colexification and Word Associations

Arjun Singh Puri (arjsingh.puri@mail.utoronto.ca)

Hassan Nishat (hassan.nishat@mail.utoronot.ca

Abstract

Colexification is a phenomenon when two concepts in a language share the same word. There exists lots of variation in how often two concepts are colexified across languages and in this paper we investigated whether this variation is non-arbitrary. Specifically, we tested the hypothesis that more frequently colexified concepts are more conceptually related and associated with each other. We used the CLICS database to get the frequency of colexifications, and data from a game by Small World of Words to get frequency of word associations or conceptual relatedness of two words. Then we created a base lexicon that comprised of pairs of concepts that were common to both datasets. We computed the spearman correlation between the frequency of colexifications across languages and associations for the pairs in the base lexicon and found a weak correlation between the two. These results suggest that conceptual relatedness or associativity might be a factor in explaining colexification variation across languages but does not tell the complete story. Other factors that might be playing a role are discussed and future directions are suggested.

Keywords: colexification; word association; conceptual relatedness; cognitive economy;

Introduction

There are various instances in every language where a single word form has multiple meanings; this phenomenon is called colexification (Comrie, 1989; Greenberg, 2010; Majid, Jordan, & Dunn, 2015). Across all languages, there are patterns of variation in how often two meanings are paired together using the same word (Srinivasan & Rabagliati, 2015; Youn et al., 2016). For instance, lots of languages colexify "moon" and "month" by using the same word for the two concepts but not many might, for example, colexify "moon" and "water". This paper intends to explore and examine why this variation occurs and its correlation with other factors.

One possible reason as to why colexification may be occurring that is suggested in the literature is the principle of least effort or the principle of cognitive economy which proposes that languages must make a trade-off and maximize expressiveness for the listener while minimizing effort on the part of the speaker (Zipf, 2016; Rosch, 1978). This could explain colexification as it is reducing the burden on the speaker by saving them the effort of learning more vocabulary to express a different meaning.

One implication of the above idea is that the conceptual relatedness of the two concepts might be a relevant factor when trying to understand the variation in colexification. The idea is that a necessary condition for two concepts being colexified is that there should be a minimal degree of relatedness between the two concepts, and this would also help with cognitive economy as the speaker only needs to memorize one word to convey two meanings in distinct contexts (Xu, Duong, Malt, Jiang, & Srinivasan, 2020). For instance, one will not expect things like "brick" and "headphones" to be colexified. Additionally, context might play a key factor because if two concepts that frequently occur in the same context used the same word it would lead to ambiguity; so, two concepts that are colexified might have some conceptual relatedness and probably occur in different contexts (Piantadosi, Tily, & Gibson, 2012).

Like Xu et al.(2020) we want to investigate and see whether the variation in colexification frequency across languages is non-arbitrary and how this connects to the idea of cognitive economy. More specifically, our hypothesis is that more frequently colexified concepts are more conceptually related and associated with each other. Accordingly, the alternate hypothesis is that colexification is arbitrary and that there is no correlation between how frequently two concepts are colexified and how closely related or associated they are conceptually.

Methods

All the python code for the analysis described below can be found in Appendix A at the end of the paper.

Colexification Across Languages

We used the CLICS database to collect colexification data on concept pairs across 3,050 unique languages and 201 unique language families. From this dataset we were able to collect the colexification counts of 74,330 concept pairs across all of these unique languages and language families. The database contains a great deal of information for each concept ranging from the underlying meaning of the concept to the geographic location where the language of the concept is found to be spoken. From this database we were able to collect the colexification counts for each language.

This is done by creating all possible unique pairs of concepts found within a language and creating a count for how many times that pair appears in the language (Figure 1). This count is the number of times those concepts are colexified in that language. After doing this for all languages in the database, we have the colexification count for all concept

pairs. One limitation that could skew results from our analysis is that concept pairs within the database are English representations of the concepts from each language. This may cause issues as the concepts within each language may not have a direct English translation.

	concept_1	concept_2	colexification_count
66603	RIVER	WATER	205
49861	HOW MANY PIECES	HOW MUCH	206
58930	MOTHER-IN-LAW (OF MAN)	MOTHER-IN-LAW (OF WOMAN)	208
37302	FATHER-IN-LAW (OF MAN)	FATHER-IN-LAW (OF WOMAN)	209
11217	BLUE	GREEN	210
5420	BARK	SKIN	215
24550	COUNTRY	LAND	226
67768	SAY	SPEAK	227
21042	CLAW	FINGERNAIL	244
27143	DAUGHTER-IN-LAW (OF MAN)	DAUGHTER-IN-LAW (OF WOMAN)	263
40118	FLESH	MEAT	270
54245	LEATHER	SKIN	279
70963	SON-IN-LAW (OF MAN)	SON-IN-LAW (OF WOMAN)	285
52776	KNIFE	KNIFE (FOR EATING)	288
2693	ARM	HAND	301
74236	WIFE	WOMAN	302
58509	MONTH	MOON	328
44184	GO	WALK	336
41201	FOOT	LEG	354
73687	TREE	WOOD	356

Figure 1: Database with frequencies of colexifications between two concepts across all languages

Association Between Concepts

To find data for association or how conceptually related words were, we used data taken from a word association game conducted by Small World of Words, a large-scale scientific study that aims to learn about words through associations. The game prompted players with a cue word and asked the player to respond with the first three words that came to their minds (Figure 2). The data consists of 100 primary, secondary and tertiary responses to 12,292 cues collected between 2011 and 2018. The data itself accounts for capitalisation and spelling and normalizes all entries to a set standard.

From this, we were able to create a list of all associated concepts by creating a pair between each cue word and its three responses. Meaning that for each instance of the game, three pairs would be added to a list: the cue and first response, the cue and second response, and the cue and third response. This was done for all entries and resulted in a list of length 3,687,600. We also made a list that only contained each cue matched with the first response given, as it allowed us to have a secondary list containing words that were more highly associated as they were the first words that came to players' minds when prompted with the cue. The data was provided in English, Spanish, and Dutch. We chose to use only the English data to match the concept pairs from the CLICS database.

Base Lexicon

Given the array of colexified concepts taken from the CLICS database and the array of associated pairs taken from the word association game, we were able to create a new list by taking only the pairs common in both datasets, giving us our base

	cue	R1	R2	R3
0	although	nevertheless	yet	but
1	deal	no	cards	shake
2	music	notes	band	rhythm
3	inform	tell	rat on	NaN
4	way	path	via	method
...	...	...	...	...
1228195	strange	mask	weird	stranger
1228196	sunset	sea	sky	clause
1228197	useless	pitty	worthless	worth
1228198	volume	loud	music	key
1228199	whenever	who	where	always

1228200 rows × 4 columns

Figure 2: Database with words and associations in the form of three responses

lexicon. We also created a base lexicon between concept pairs within the CLICS database and the list of associations with only the first response.

With these two base lexicons, we then created lists that contained colexification and association data for only the pairs within the base lexicons. We got the colexification counts for each concept pair in both base lexicons by searching for each pair within the colexification count database that we created using the original CLICS database, and extracting the colexification count for that pair. We were also able to extract the number of associations between each pair in both base lexicons by calculating the number of times the pair appears in the original association data. Now that we had access to the colexification and association counts of each pair within our two base lexicons, we were able to analyze the data.

	Base_Lexicon	Colex_Counts	Assoc	Assoc_R1
0	[paper, book]	67	6	3
1	[town, village]	68	29	11
2	[mind, brain]	24	55	22
3	[read, learn]	63	10	2
4	[board, wood]	19	35	9
...	...	...	...	...
71898	[preserve, keep]	166	30	20
71899	[stream, river]	32	52	30
71900	[toe, foot]	5	65	49
71901	[breast, chicken]	1	26	7
71902	[bark, tree]	3	79	29

71903 rows × 4 columns

Figure 3: Base lexicon pairs with frequency of colexification and association for each pair

Spearman Correlation

To determine a link between colexification and association, we decided to calculate the Spearman Correlation (Figure 4). The spearman correlation is a way to describe how strongly a function is monotonic. A monotonic function is a function in which as one variable increases, the other either also increases or decreases. In the case that both variables are increasing, the spearman correlation would result in a positive value and if one of the variables is decreasing, the result would be a negative value. If the correlation is 1 or -1 it implies an exactly monotonic function. The spearman correlation is used to assess our data instead of the pearson correlation as the pearson correlation assumes the data follows a normative distribution while our data does not. The Spearman correlation avoids this problem as it creates a list of ranked values based on the data rather than using the raw data itself. When this new rank-based list is created, the pearson correlation can be calculated.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

ρ = Spearman's rank correlation coefficient

d_i = difference between the two ranks of each observation

n = number of observations

Figure 4: Formula for Spearman Correlation

Results

Using the spearman correlation we were able to quantify the relation between colexification and association. After calculating the correlation for colexification with all association data and collexification with just data on the first responses, we found the correlation coefficients to be 0.244, and 0.249 respectively with a p-vlaue of 0.0 in both cases (Figure 5).

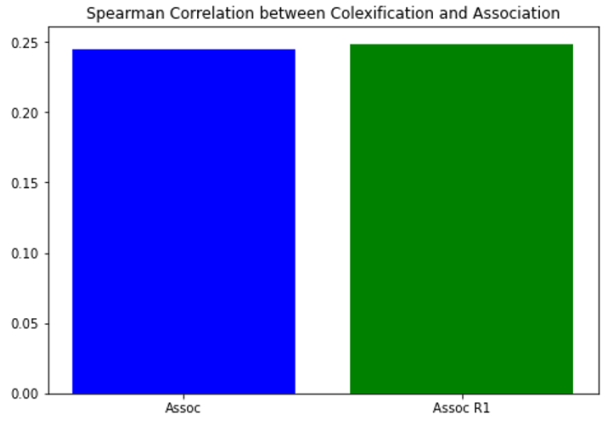


Figure 5: Spearman correlation values for the two base lexicons

From these two values we can see that in both cases there is a positive but somewhat weak correlation. This could show that as conceptual relatedness between concepts increases, the colexification between those concepts also increases but not at the level at which we had initially thought. We can also see that the correlation is slightly higher in the case where only the first response in the association data was used. This could indicate that more strongly associated concepts have a higher rate of colexification. These results are also comparable to the correlation that Xu et al.(2020), a paper we were partly trying to replicate, found.

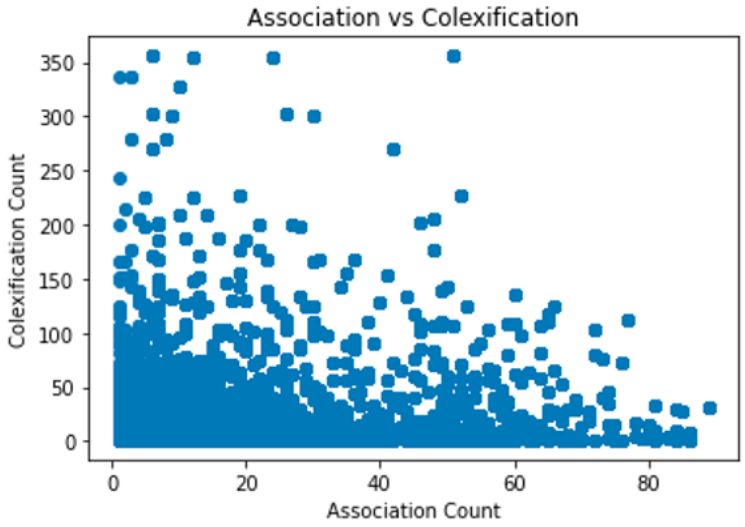


Figure 6: Scatter plot with colexification Count (y-axis) vs Association Count (x-axis)

Conclusion

Our results suggest that there is a weak, but positive, correlation between word associativity and concept colexification. This means conceptual relatedness or associativity may be a factor in explaining the variation in the frequency of colexification across languages but there could possibly be other factors at play. For instance, the effect of context may be much greater than anticipated. For example, concepts like "hot" and "cold" or "sister" and "brother" are highly associated but may never be colxified because they mostly occur in the same context.

Some further extensions we could make to our study could be to account for what is known as the Goldilocks Principle. This principle is the idea that the relation between colexification and association between concepts is not a linear relation but rather starts to decrease as concepts become too related (Brochhagen & Boleda, 2022). This implies that there is a high correlation between association and colexification up until a certain point after which, concepts become too associated and probably occur within the same context breaking the correlation. Future works could assess if this is true by replicating our analysis separately with a set of pairs that are

loosely associated, moderately associated, and highly associated.

References

Brochhagen, T., & Boleda, G. (2022). The interaction between cognitive ease and informativeness shapes the lexicons of natural languages. *Proceedings of the Society for Computation in Linguistics*, *5*(1), 217–219.

Comrie, B. (1989). *Language universals and linguistic typology: Syntax and morphology*. University of Chicago press.

Greenberg, J. H. (2010). Language universals. In *Language universals.* De Gruyter Mouton.

Majid, A., Jordan, F., & Dunn, M. (2015). *Semantic systems in closely related languages* (Vol. 49). Elsevier.

Piantadosi, S. T., Tily, H., & Gibson, E. (2012). The communicative function of ambiguity in language. *Cognition*, *122*(3), 280–291.

Rosch, E. (1978). Principles of categorization.

Srinivasan, M., & Rabagliati, H. (2015). How concepts and conventions structure the lexicon: Cross-linguistic evidence from polysemy. *Lingua*, *157*, 124–152.

Xu, Y., Duong, K., Malt, B. C., Jiang, S., & Srinivasan, M. (2020). Conceptual relations predict colexification across languages. *Cognition*, *201*, 104280.

Youn, H., Sutton, L., Smith, E., Moore, C., Wilkins, J. F., Maddieson, I., ... Bhattacharya, T. (2016). On the universal structure of human lexical semantics. *Proceedings of the National Academy of Sciences*, *113*(7), 1766–1771.

Zipf, G. K. (2016). *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books.

Appendix A

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tqdm import tqdm
from itertools import product
import plotly.express as px
from scipy.stats import pearsonr, spearmanr, linregress


pd.options.mode.chained_assignment = None
```

Muhammad Hassan Nishat 1005835250 & Arjun Puri 1004707490

Required Data:

CLICS Database: https://clics.clld.org/

Word Association Data (SWOW-EN.R100.csv): https://osf.io/hjvm5/

Base Lexicon Files: https://github.com/hassan3301/COG260-Data.git

```
df = pd.read_csv("df_all_raw.csv")
df.columns = list(map(str.lower, df.columns))
df = df.drop(columns=['dataset_id', 'form_id', 'form',
'gloss_in_source', 'iso639p3code', 'mrc_word',
'kucera_francis_frequency'])
```

```
/var/folders/6j/k_ngy9gs69sckj60lmgylm1r0000gn/T/
ipykernel_47912/2801652837.py:1: DtypeWarning: Columns (4) have mixed
types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv("df_all_raw.csv")
```

```
def per_lang_colexification(curr_df):
    """
    Calculate the colexification frequency of pairs of concepts
present in the current language.
    """
    all_combos_dict = {}
    # We iterate through each row, which has the concepts associated
with a specific word
    for i, row in curr_df.iterrows():
        # Get the current set of concepts
        a = row['concepticon_gloss']
        # Create all possible unique combinations of concepts, where
each pair is alphabetically sorted
        combos = list(set(map(lambda x: tuple(sorted(x)), product(a,
a))))
        # Ensure the concepts in the pair are not identical
        combos = [combo for combo in combos if combo[0] != combo[1]]
        # Add counts for a pair of combinations being colexified
        for combo in combos:
            if combo in all_combos_dict:
```

```
                all_combos_dict[combo] += 1
            else:
                all_combos_dict[combo] = 1

    # Create a DataFrame out of our dictionary and return the
colexification counts for two concepts
    tmp = pd.DataFrame.from_dict(all_combos_dict,
"index").reset_index()
    per_lang = pd.DataFrame(tmp['index'].tolist(),
columns=['concept_1', "concept_2"])
    per_lang['colexification_count'] = tmp[0]
    return per_lang

def main():
    all_dfs = []
    for variety in tqdm(df['variety'].unique()):
        sub = df[df['variety'] == variety]
        agg = sub.groupby("clics_form")[['concepticon_gloss',
'concepticon_id']].agg(list)
        agg['num_concepts'] = agg['concepticon_gloss'].apply(lambda x:
len(set(x)))
        colex = agg[agg['num_concepts']>1]
        colex['concepticon_gloss'] =
colex['concepticon_gloss'].apply(lambda x: sorted(list(set(x))))
        # We skip any language where no concepts are colexified
        if colex.shape[0] == 0:
            continue
        curr_df = per_lang_colexification(colex)
        all_dfs.append(curr_df)
    mega = pd.concat(all_dfs)
    colex_counts = mega.groupby(["concept_1",
"concept_2"]).sum().reset_index()
    return colex_counts

colex_counts = main()
```

```
100%|██████████| 3050/3050 [03:17<00:00, 15.43it/s]
```

```
assoc_en = pd.read_csv('SWOW-EN.R100.csv')
assoc = assoc_en[assoc_en.columns[9:]]
assoc
```

```
              cue            R1         R2        R3
0        although  nevertheless        yet       but
1            deal            no      cards     shake
2           music         notes       band    rhythm
3          inform          tell     rat on       NaN
4             way          path        via    method
...           ...           ...        ...       ...
1228195   strange          mask      weird  stranger
1228196    sunset           sea        sky    clause
```

```
1228197   useless         pitty  worthless     worth
1228198    volume          loud      music       key
1228199  whenever           who      where    always

[1228200 rows x 4 columns]
```

```
colexarr =[]
for index, row in colex_counts.iterrows():
    colexarr.append([str(row['concept_1']).lower(),
str(row['concept_2']).lower()])
```

```
assocarr = []
for index, row in assoc.iterrows():
    for i in range(3):
        assocarr.append([str(row['cue']).lower(),
str(row['R'+str(i+1)]).lower()])
```

```
assocarr_R1 = []
for index, row in assoc.iterrows():
    assocarr_R1.append([str(row['cue']).lower(),
str(row['R1']).lower()])
```

```
base_lexicon = []
for i in tqdm(assocarr):
    if i in colexarr or [i[1], i[0]] in colexarr:
        base_lexicon.append(i)
```

These 4 code cells below create lists of colexification and association counts for our base lexicon. There is also an option to import this data from csv files below.

```
base_lexicon_colex = []

for i in tqdm(range(len(base_lexicon))):
    query1 = "concept_1=='{}' and
concept_2=='{}'".format(base_lexicon[i][0].upper(),base_lexicon[i]
[1].upper())
    query2 = "concept_1=='{}' and
concept_2=='{}'".format(base_lexicon[i][1].upper(),base_lexicon[i]
[0].upper())
    df1 = colex_counts.query(query1)

    if not df1.empty:
        base_lexicon_colex.append(df1['colexification_count'])
    else:
        df2 = colex_counts.query(query2)
        base_lexicon_colex.append(df2['colexification_count'])

base_lexicon_assoc = []
for i in tqdm(base_lexicon):
    count = 0
    for j in assocarr:
```

```
        if (i[0] == j[0] and i[1] == j[1]):
            count+=1
    base_lexicon_assoc.append(count)

base_lexicon_assoc_R1 = []
for i in tqdm(base_lexicon):
    count = 0
    for j in assocarr_R1:
        if (i[0] == j[0] and i[1] == j[1]):
            count+=1
    base_lexicon_assoc_R1.append(count)
```

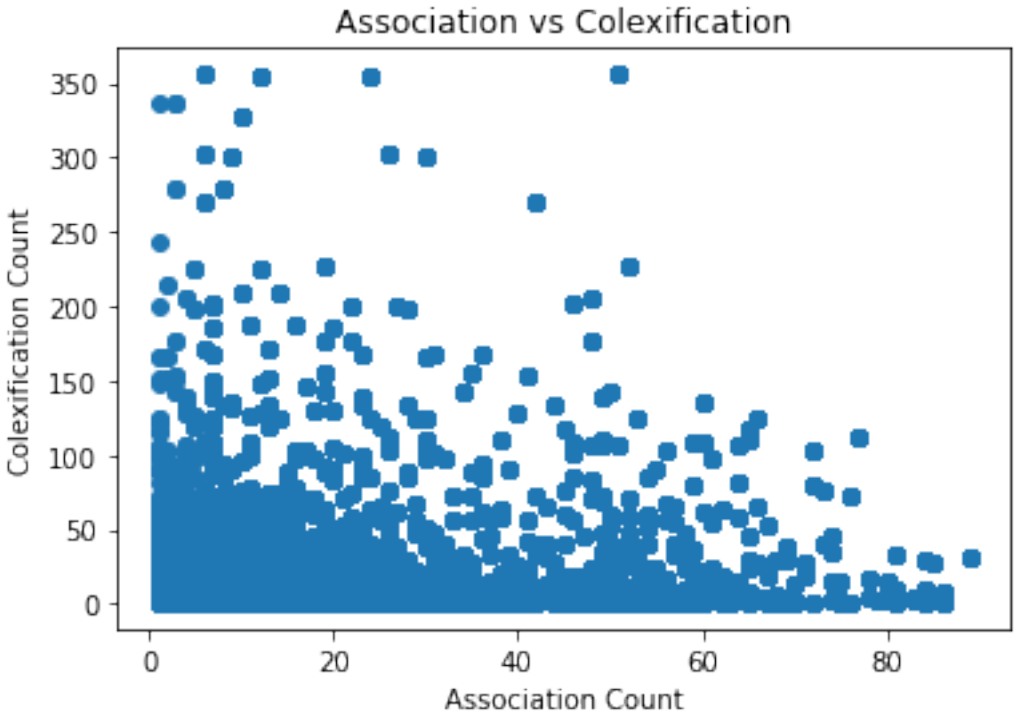
Run the below code to import the csv files filled with the lists created by the code above.

```
import csv
with open('base_lexicon.csv', newline='') as f:
    reader = csv.reader(f)
    base_lexicon = list(reader)

for i in range(len(base_lexicon)):
    base_lexicon[i][1] = base_lexicon[i][1].strip()

del base_lexicon[33082]

with open('base_lexicon_assoc.csv', newline='') as f:
    reader = csv.reader(f)
    base_lexicon_assoc = list(reader)
del base_lexicon_assoc[33082]

with open('base_lexicon_assoc_R1.csv', newline='') as f:
    reader = csv.reader(f)
    base_lexicon_assoc_R1 = list(reader)
del base_lexicon_assoc_R1[33082]

with open('base_lexicon_colex.csv', newline='') as f:
    reader = csv.reader(f)
    base_lexicon_colex = list(reader)

base_lexicon_colex = [int(item) for sublist in base_lexicon_colex for
item in sublist]
base_lexicon_assoc = [int(item) for sublist in base_lexicon_assoc for
item in sublist]
base_lexicon_assoc_R1 = [int(item) for sublist in
base_lexicon_assoc_R1 for item in sublist]

d = {'Base_Lexicon': base_lexicon, 'Colex_Counts': base_lexicon_colex,
'Assoc': base_lexicon_assoc, 'Assoc_R1': base_lexicon_assoc_R1}
df2 = pd.DataFrame(data=d)
df2
```

```
            Base_Lexicon  Colex_Counts  Assoc  Assoc_R1
0          [paper, book]            67      6         3
1        [town, village]            68     29        11
2          [mind, brain]            24     55        22
3          [read, learn]            63     10         2
4          [board, wood]            19     35         9
...                  ...           ...    ...       ...
71898   [preserve, keep]           166     30        20
71899    [stream, river]            32     52        30
71900        [toe, foot]             5     65        49
71901  [breast, chicken]             1     26         7
71902       [bark, tree]             3     79        29

[71903 rows x 4 columns]
```

```
plt.scatter(base_lexicon_assoc, base_lexicon_colex)
plt.title('Association vs Colexification')
plt.xlabel("Association Count")
plt.ylabel("Colexification Count")
```

```
Text(0, 0.5, 'Colexification Count')
```




```
spearmanR123 = spearmanr(base_lexicon_assoc, base_lexicon_colex)
spearmanR1 = spearmanr(base_lexicon_assoc_R1, base_lexicon_colex)

print(spearmanR123, spearmanR1)
```

```
SpearmanrResult(correlation=0.24433085061261967, pvalue=0.0)
SpearmanrResult(correlation=0.24878294815024218, pvalue=0.0)
```

```
R123corr = spearmanR123[0]
R1corr = spearmanR1[0]

fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
x = ['Assoc', 'Assoc R1']
y = [R123corr, R1corr]
ax.bar(x, y, color=['blue', 'green'])
plt.title('Spearman Correlation between Colexification and
Association')
plt.show()
```