

# The Difference between Schools of Philosophy

This project mainly talked about the differences between schools of philosophy. Those differences are talked by analized the texts of spanning 13 major schools of philosophy. The data sets used in this project can be found at <https://www.kaggle.com/kouroshalizadeh/history-of-philosophy>.

```
In [56]: import pandas as pd
import matplotlib.pyplot as plt
from wordcloud import WordCloud, STOPWORDS
```

```
In [57]: df = pd.read_csv(' /Users/zhang/Desktop/philosophy_data.csv' )
```

There are 13 schools included in this data set. They are analytic, aristotle, german\_idealism, plato, continental, phenomenology, rationalism, empiricism, feminism, capitalism, communism, nietzsche and stoicism.

```
In [67]: df['school'].value_counts().index.to_list()
```

```
Out[67]: ['analytic',
'aristotle',
'german_idealism',
'plato',
'continental',
'phenomenology',
'rationalism',
'empiricism',
'feminism',
'capitalism',
'communism',
'nietzsche',
'stoicism']
```

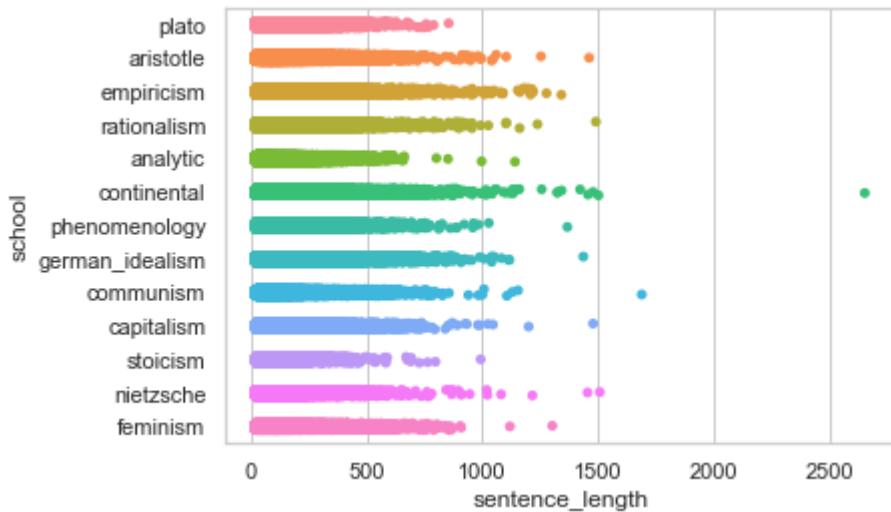
## Lenth of Texts

Firstly, I focused on the lenth of the texts of different schools.

```
In [88]: import seaborn
seaborn.set(style='whitegrid')

seaborn.stripplot(x="sentence_length",
                  y="school",
                  data=df,
                  size=5)
```

```
Out[88]: <AxesSubplot:xlabel='sentence_length', ylabel='school'>
```



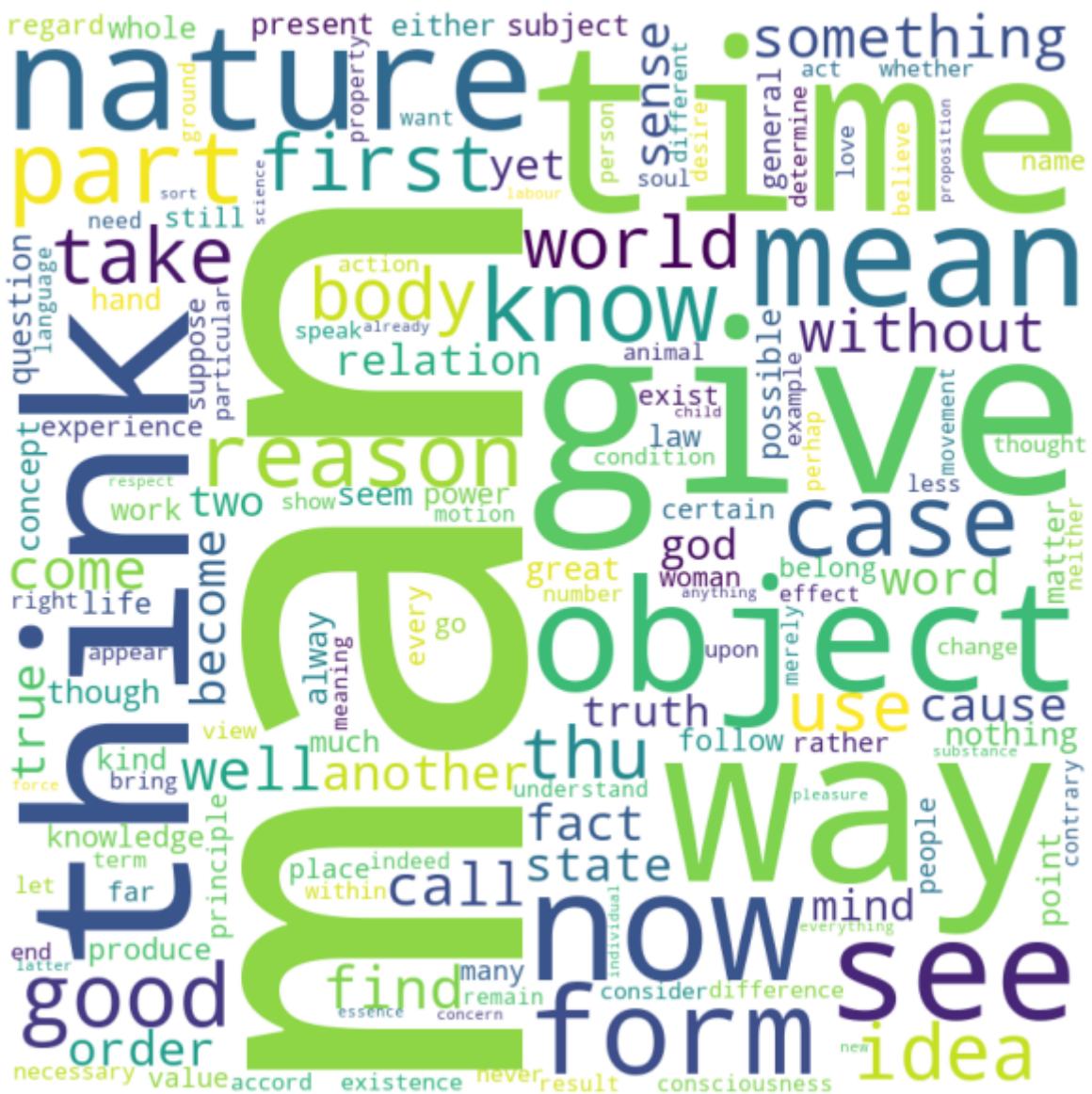
It appears that plato, analytic and stoicism less long texts but the difference is not significant in the plot above.

## Content Different Philosophy Schools Focus on

Secondly, I focus on the content those philosophy schools mainly mentioned in the texts.

```
In [64]: comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev
for val in df.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



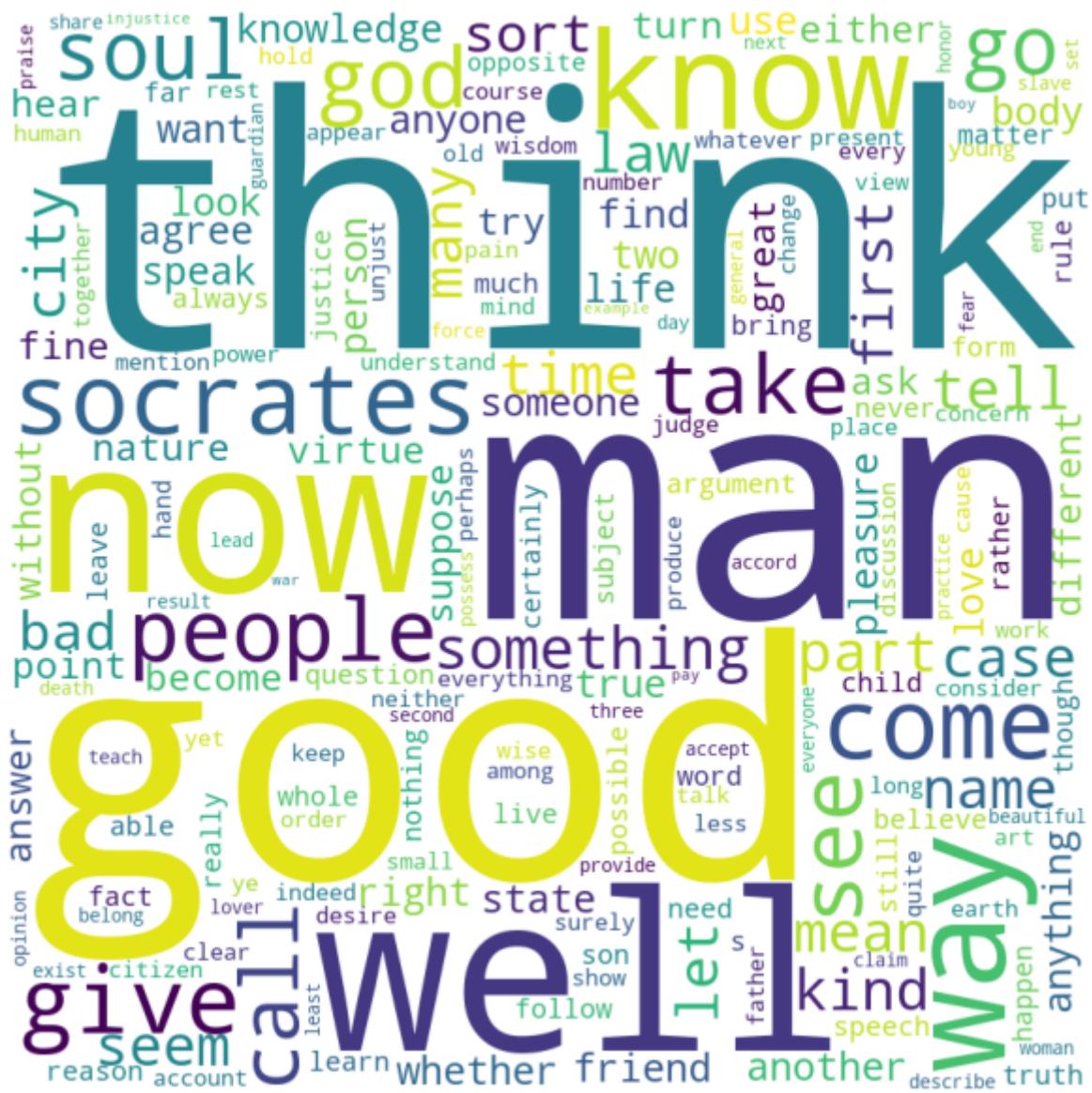
It appears that 'man', 'time' and 'nature' are the main things philosophers discuss.

When I separate those texts according to the schools, I found that different schools mainly talked about different things.

```
In [65]: df1=df[df['school']=='plato']
comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev

for val in df1.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

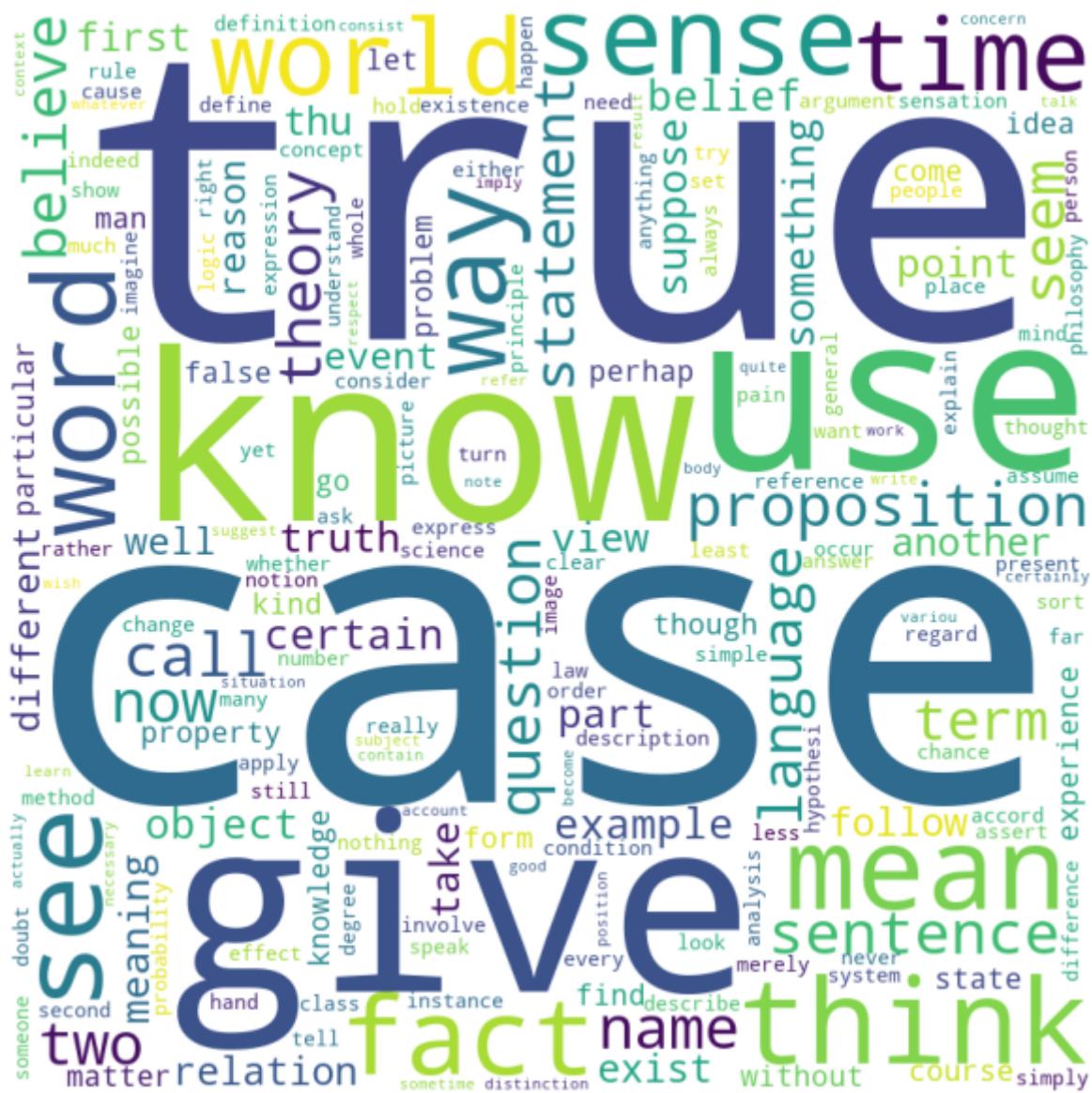


Platonic school mainly talks about 'man', 'now' and 'people's thoughts'.

```
In [66]: df2=df[df['school']=='analytic']
comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev

for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Analytic school mainly talks about 'case', 'truth' and 'give'.

```
In [68]: df2=df[df['school']=='aristotle']
comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev

for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

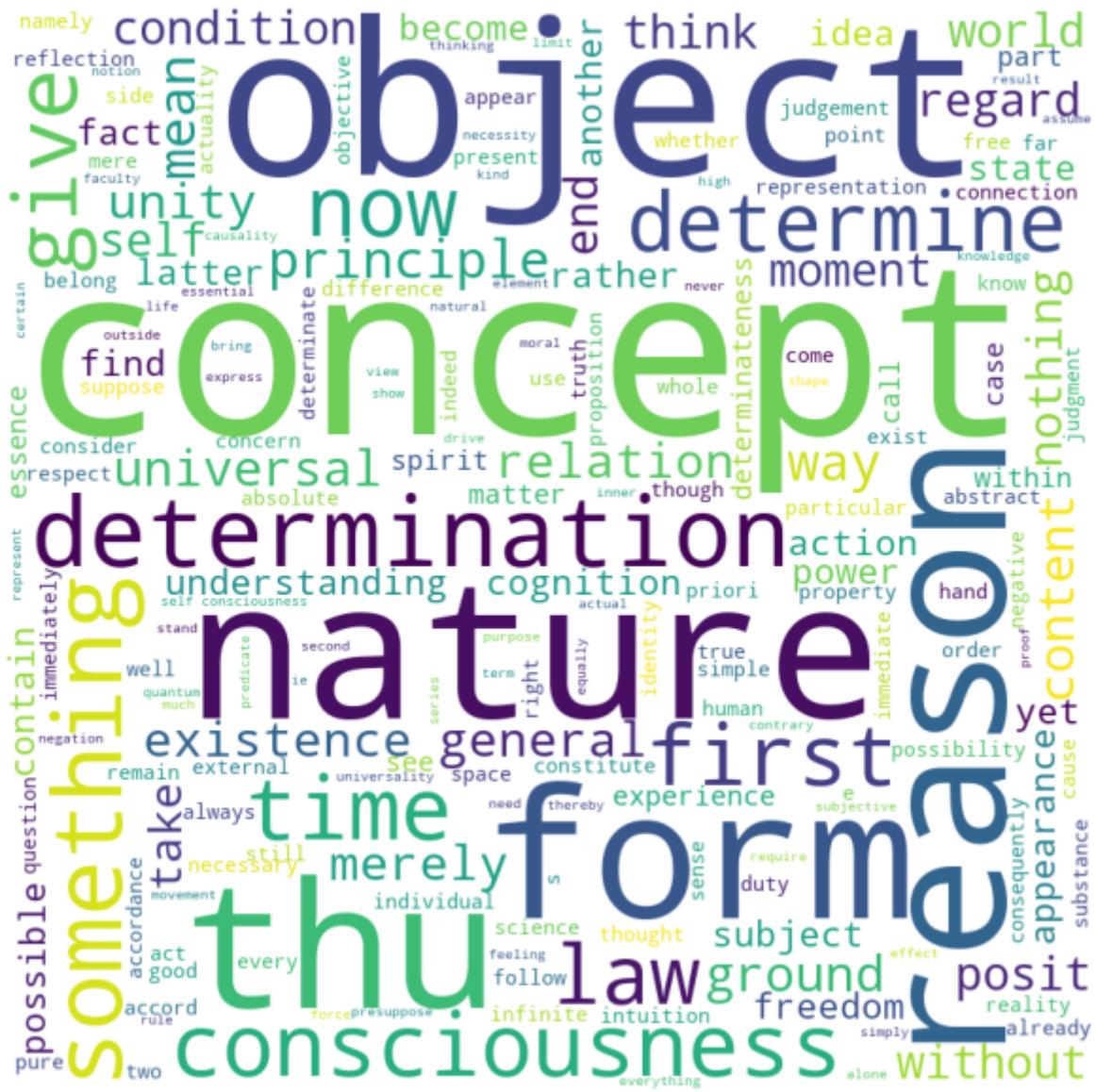


Aristotelian school mainly talks about 'man', 'animal' and 'body'.

```
In [69]: df2=df[df['school']=='german_idealism']
comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev

for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

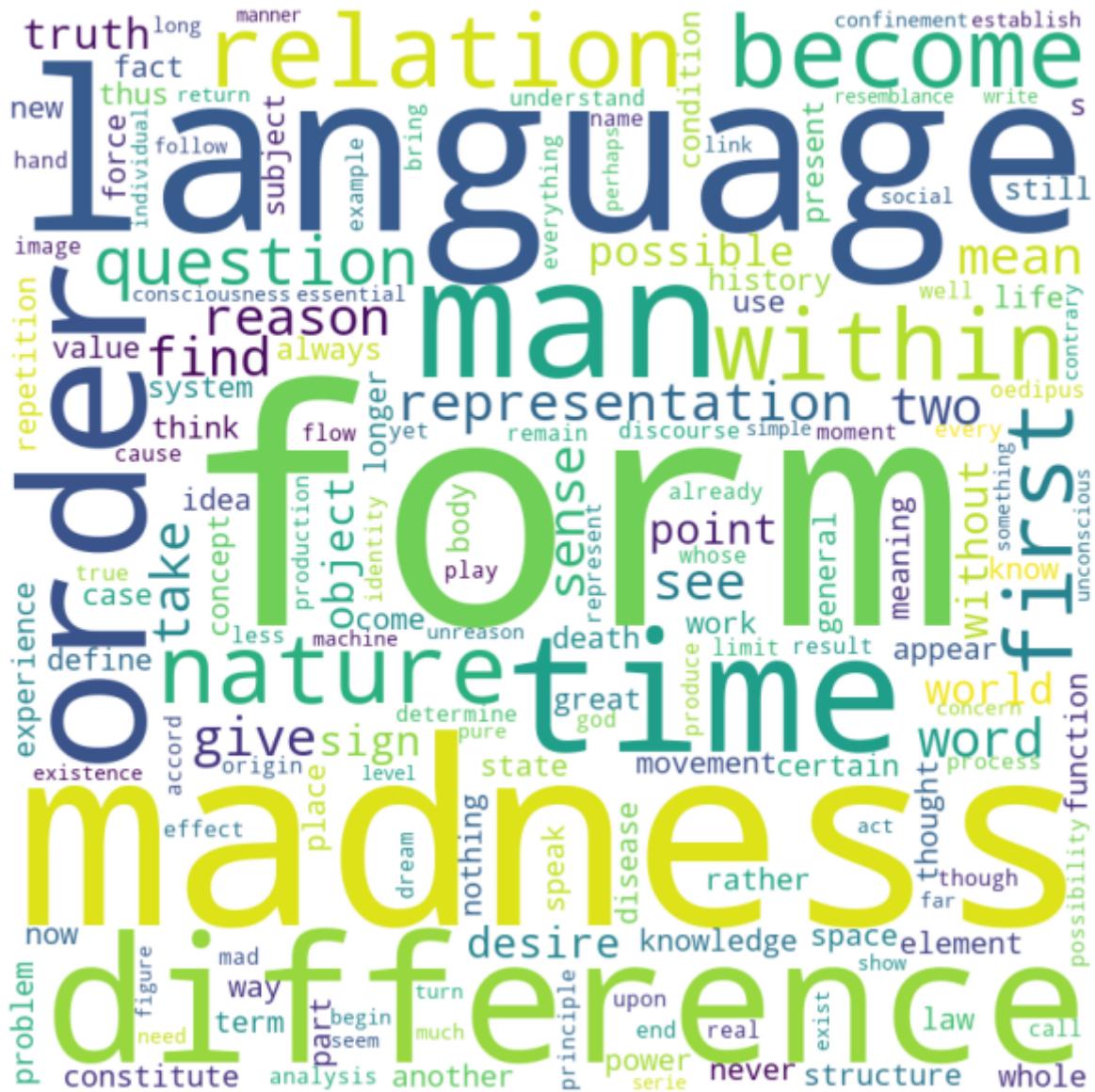
plt.show()
```



School of German Idealism mainly talks about 'concept', 'nature' and 'thu'.

```
In [70]: df2=df[df['school']=='continental']
comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev
for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

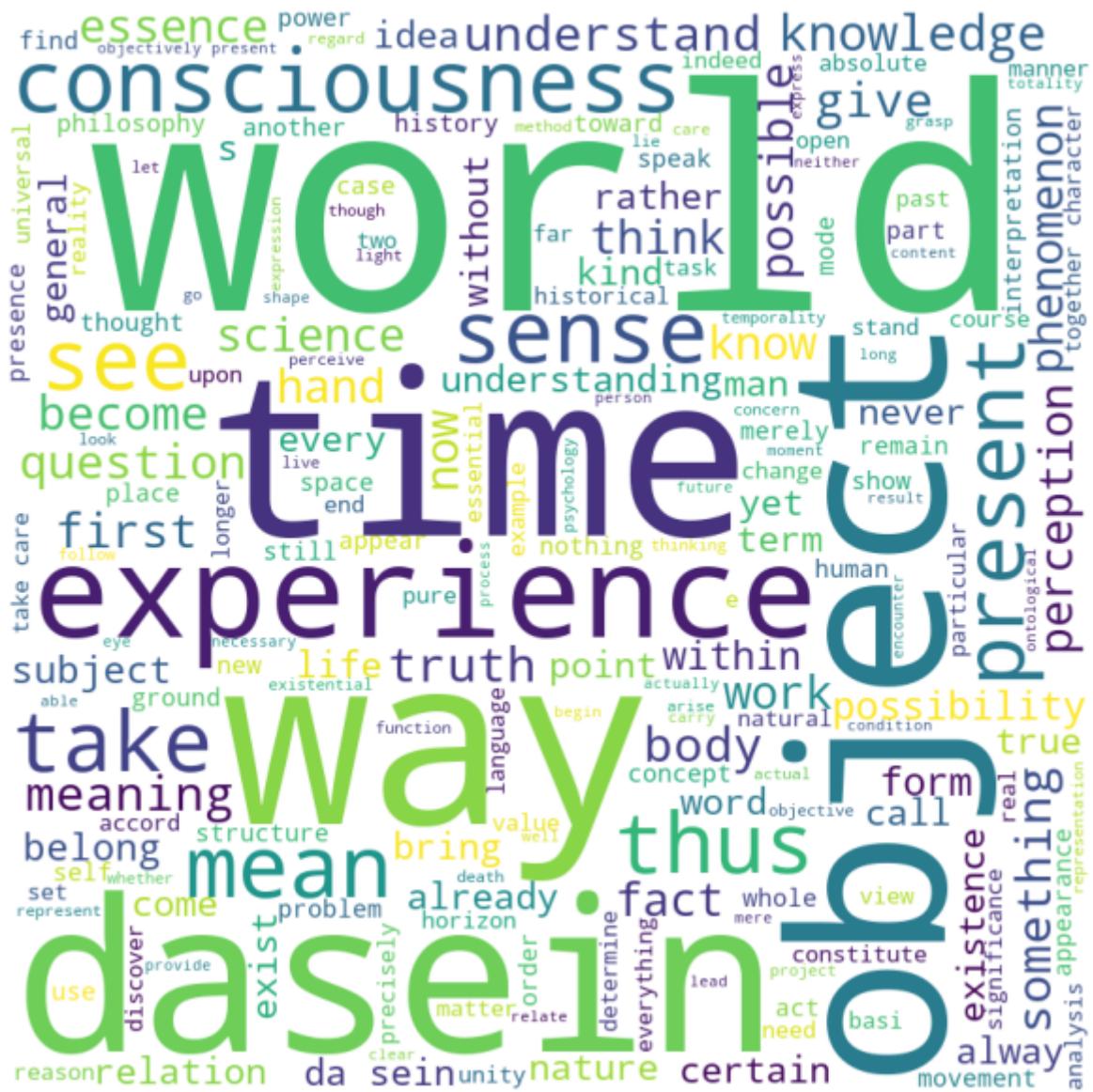
plt.show()
```



Continental school mainly talks about 'language', 'madness' and 'difference'.

```
In [71]: df2=df[df['school']=='phenomenology']
comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev
for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Phenomenology mainly talked about 'world', 'time', 'dasein'.

```
In [145...]: df2=df[df['school']=='rationalism']
comment_words = ''
stopwords = [ "pron", "one", "will", "thing", "say", "make", "must", "may", "ev

for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

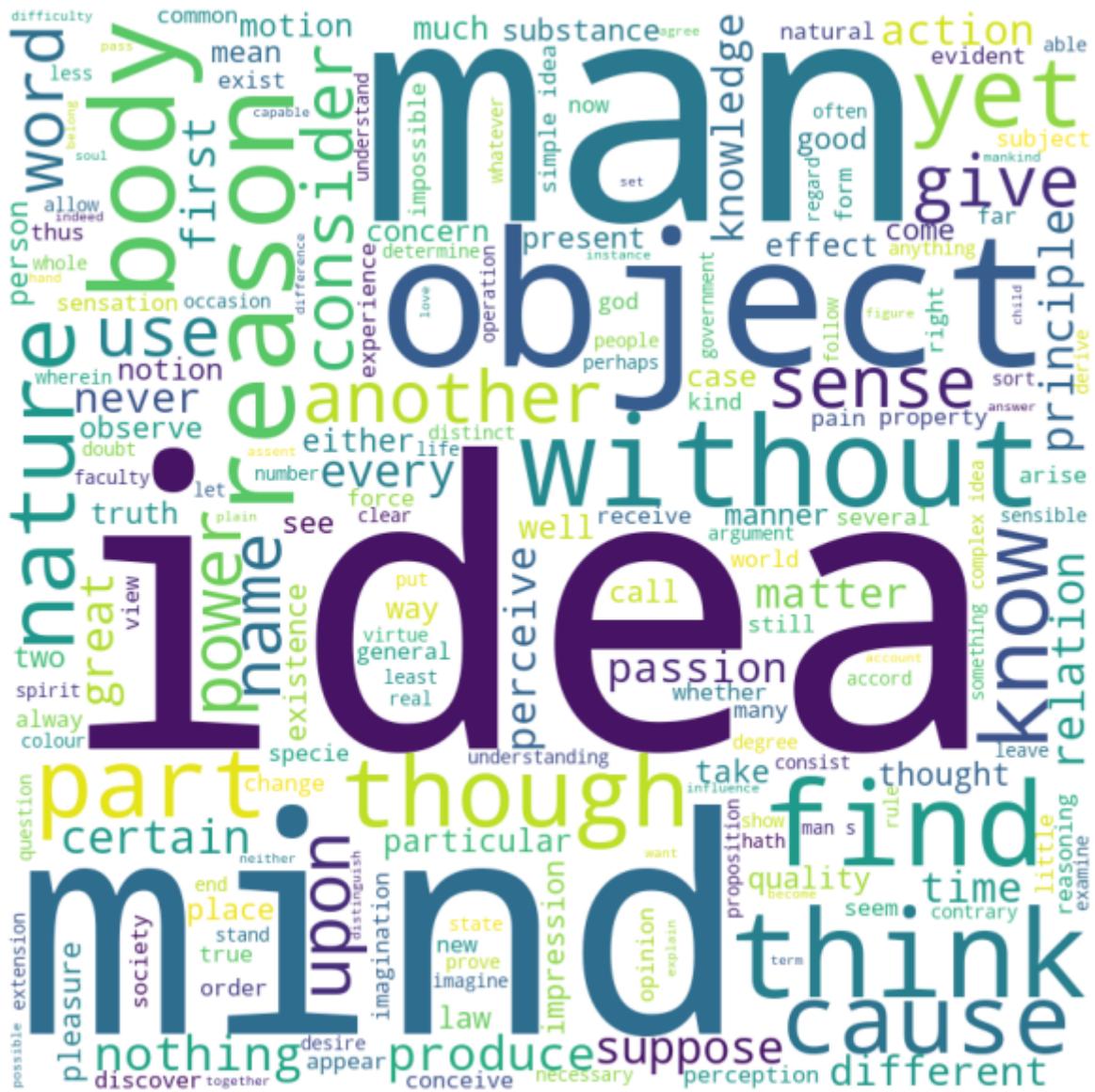


Rationalism mainly talked about 'mind', 'god', 'body'.

```
In [73]: df2=df[df['school']== 'empiricism']
comment_words = ''
stopwords = ["pron", "one","will","thing", "say", "make", "must", "may", "ev

for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```

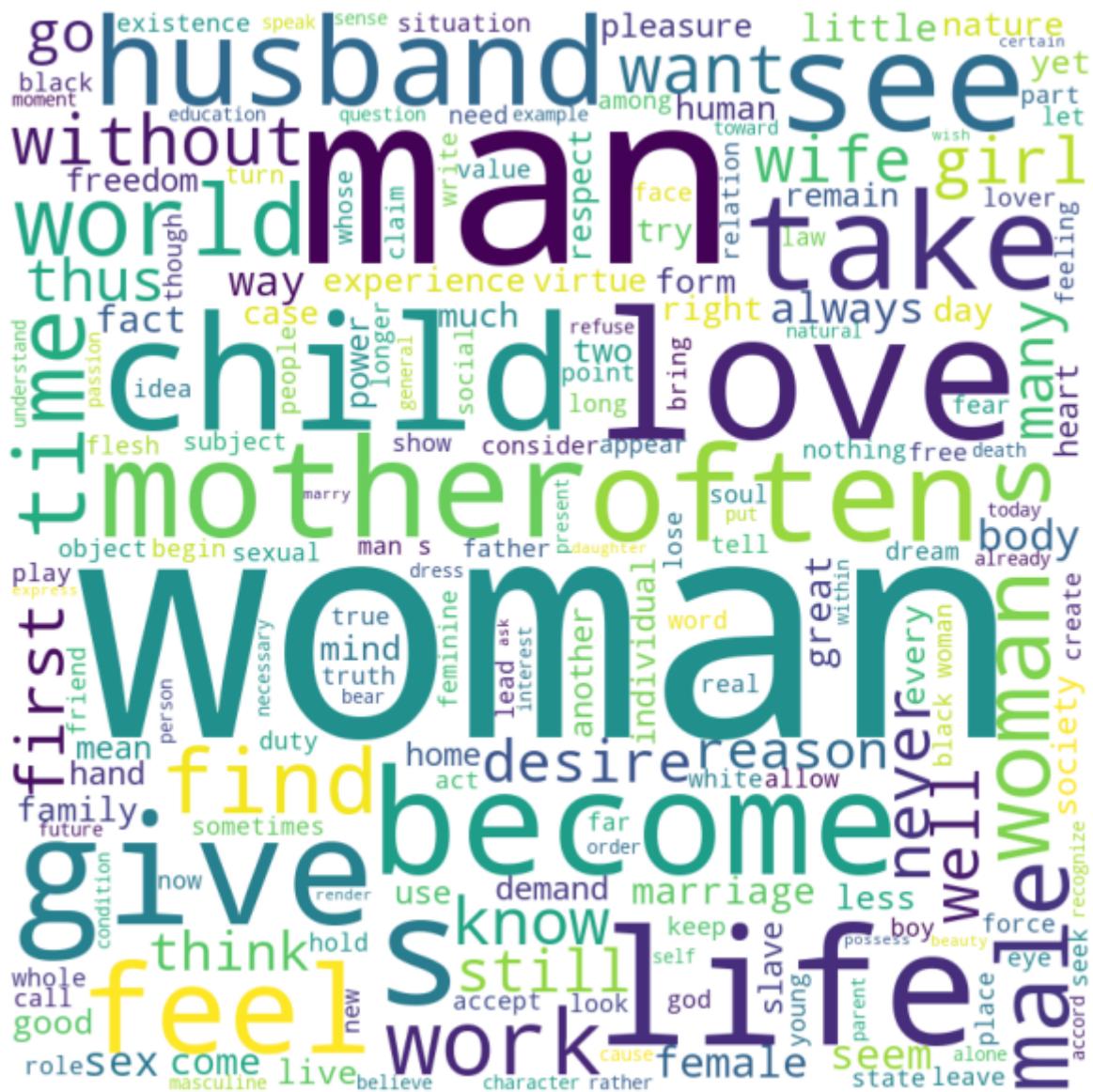


Empiricism mainly talked about 'mind', 'man', 'idea'.

```
In [74]: df2=df[df['school']=='feminism']
comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev

for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Feminism mainly talked about 'woman', 'man', 'love'.

```
In [75]: df2=df[df['school']=='capitalism']
comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev

for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Capitalism mainly talked about 'value', 'country', 'labour'.

```
In [76]: df2=df[df['school']== 'communism']
comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev
for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Communism mainly talked about 'commodity', 'labour', 'value'.

```
In [77]: df2=df[df['school']=='nietzsche']
comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev

for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

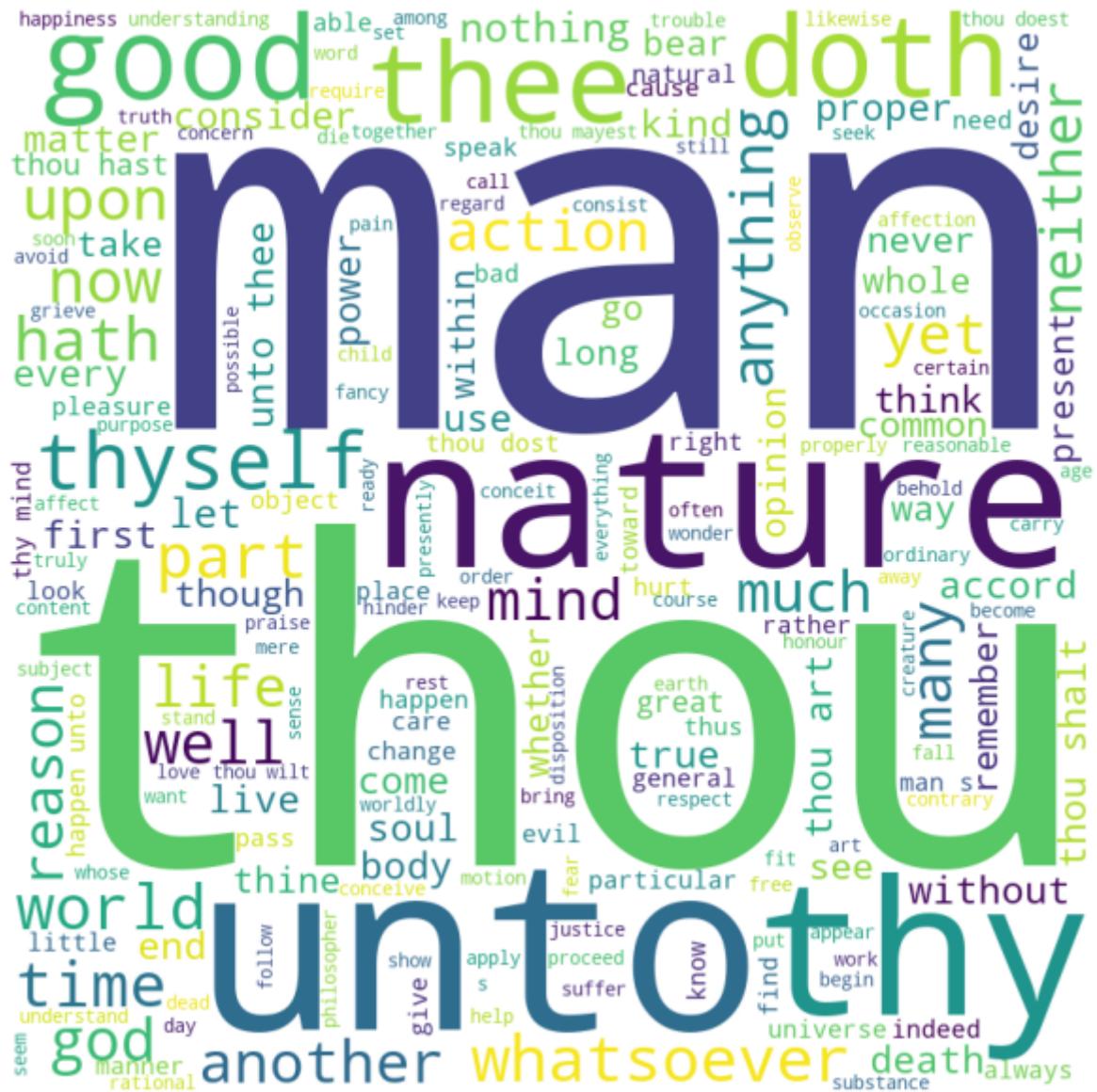
plt.show()
```



Nietzsche school mainly talked about 'man', 'life', 'god'.

```
In [81]: df2=df[df['school']=='stoicism']
comment_words = ''
stopwords = ["pron", "one", "will", "thing", "say", "make", "must", "may", "ev
for val in df2.lemmatized_str:
    val = str(val)
    tokens = val.split()
    for i in range(len(tokens)):
        tokens[i] = tokens[i].lower()
    comment_words += " ".join(tokens)+" "
wordcloud = WordCloud(width = 800, height = 800,
                      background_color ='white',
                      stopwords = stopwords,
                      min_font_size = 10).generate(comment_words)
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



Stoicism school mainly talked about 'man', 'nature', 'thou'.

## Frequency of Using Interrogative Sentence

Finally, I focus on the frequency of using interrogative sentence in each school's text.

```
In [150]: total_number = list(df.school.value_counts())
total_number
```

```
Out[150]: [55425,
 48779,
 42136,
 38366,
 33779,
 28573,
 22949,
 19931,
 18635,
 18194,
 17958,
 13548,
 2535]
```

```
In [126]:  
def numberOfQuestionMarks(str):  
    number = str.count("?")  
    return number  
dic={}  
for a in ['analytic', 'aristotle', 'german_idealism', 'plato', 'continental'  
df2=df[df['school']==a]  
df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)  
dic.update({a: sum(df2['number'])})  
dic
```

```
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy  
df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)  
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy  
df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)  
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy  
df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)  
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy  
df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)  
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy  
df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)  
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy  
df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)  
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:  
A value is trying to be set on a copy of a slice from a DataFrame.  
Try using .loc[row_indexer,col_indexer] = value instead  
  
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy  
df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)
```

```
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
    df2['number']=df2['sentence_spacy'].apply(numberOfQuestionMarks)
/var/folders/r3/h0_qjgy17_5d6qwjzr9f7yfm0000gn/T/ipykernel_878/2100915909.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

Out[126]: {'analytic': 3230,
           'aristotle': 3425,
           'german_idealism': 655,
           'plato': 8424,
           'continental': 1795,
           'phenomenology': 1562,
           'rationalism': 573,
           'empiricism': 1402,
           'feminism': 708,
           'capitalism': 164,
           'communism': 473,
           'nietzsche': 1310,
           'stoicism': 407}
```

In [154...]

```
newdf = pd.DataFrame(dic.items(),columns=['school', 'number of question mark'])
newdf['total_number'] = total_number
newdf['frequency'] = newdf['number of question marks'] / newdf['total_number']
newdf
```

Out [154] :

	school	number of question marks	total_number	frequency
0	analytic	3230	55425	0.058277
1	aristotle	3425	48779	0.070215
2	german_idealism	655	42136	0.015545
3	plato	8424	38366	0.219569
4	continental	1795	33779	0.053140
5	phenomenology	1562	28573	0.054667
6	rationalism	573	22949	0.024968
7	empiricism	1402	19931	0.070343
8	feminism	708	18635	0.037993
9	capitalism	164	18194	0.009014
10	communism	473	17958	0.026339
11	nietzsche	1310	13548	0.096693
12	stoicism	407	2535	0.160552

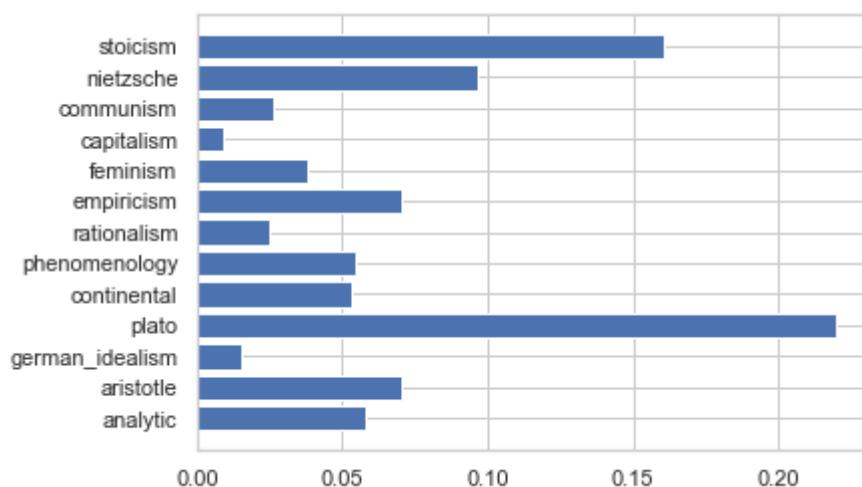
It can be seen that the frequency for Analytic school to use interrogative sentences is 5.8%, Aristotelian school 7.0%, School of German Idealism 1.6%, Platonic school 22.0%, Continental school 5.3%, Phenomenology 5.5%, Rationalism 2.5%, Empiricism 7.0%, Feminism 3.8%, Capitalism 0.9%, Communism 2.6%, Nietzsche school 9.7%, Stoicism 16.1%.

In [159] :

```
import matplotlib.pyplot as plt
school = list(newdf['school'])
frequency = list(newdf['frequency'])

plt.barh(school, frequency)
```

Out[159] : &lt;BarContainer object of 13 artists&gt;



From the plot above, it can be shown more obviously that Platonic school uses interrogative sentences most frequently, followed by Stoicism, Whereas capitalism uses interrogative sentences least frequently.