Use either `nlp` or `bertopic` virtualenv

```
$ cd ~/projects/NLP
$ source bertopic/bin/activate
```

In [1]:
```
1 from textnets import Corpus, Textnet
```

In [2]:
```
1 from textnets import examples
```

In [3]:
```
1 corpus = Corpus(examples.moon_landing, lang="en_core_web_md")
```

In [4]:
```
1 corpus
```

Out[4]:

label	
The Guardian	3:56 am: Man Steps On to the Moon
New York Times	Men Walk on Moon -- Astronauts Land on Plain, Collect Rocks, Plant Flag
Boston Globe	Man Walks on Moon
Houston Chronicle	Armstrong and Aldrich "Take One Small Step for Man" on the Moon
Washington Post	The Eagle Has Landed -- Two Men Walk on the Moon
Chicago Tribune	Giant Leap for Mankind -- Armstrong Takes 1st Step on Moon
Los Angeles Times	Walk on Moon -- That's One Small Step for Man, One Giant Leap for Mankind

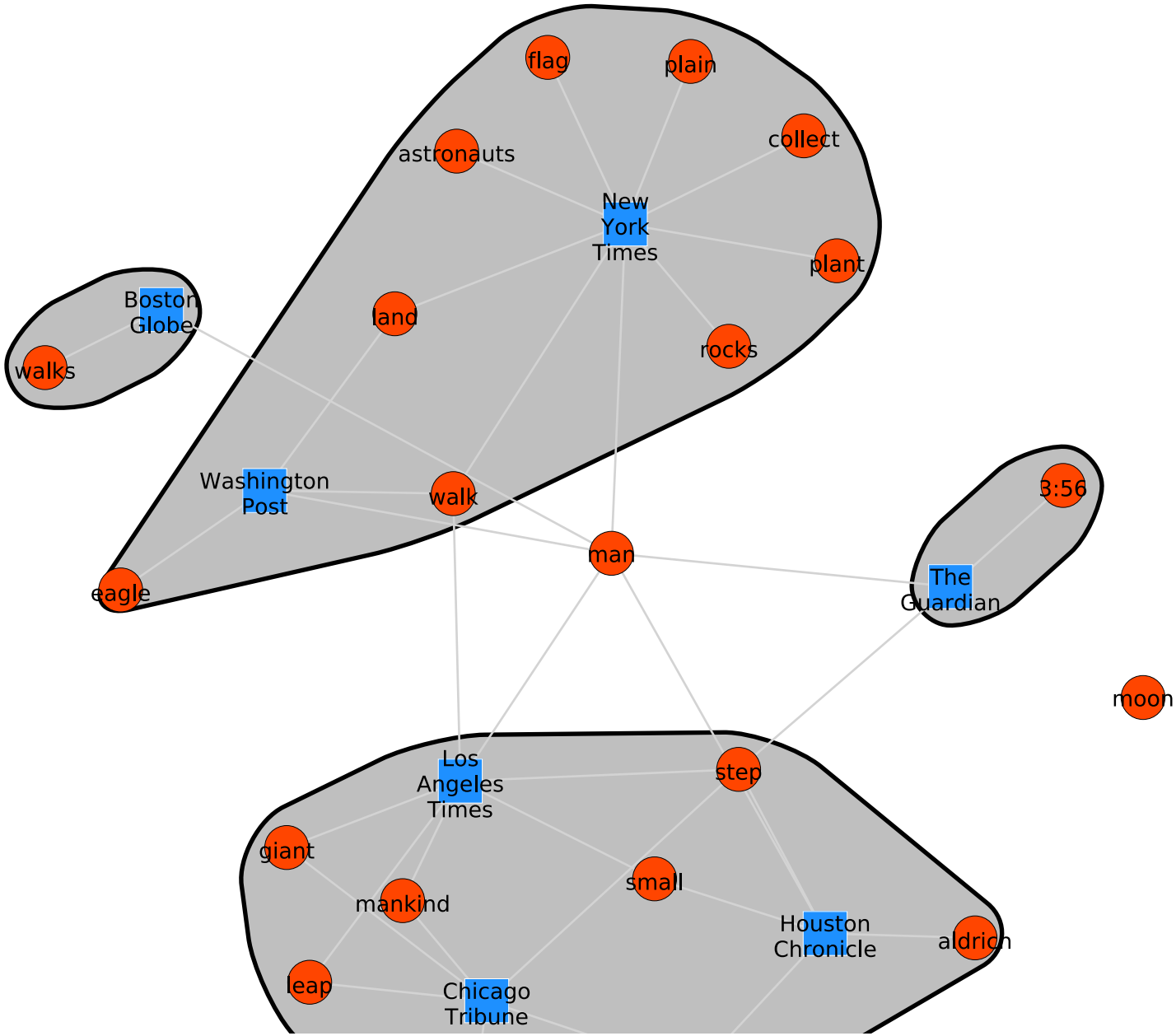
`Corpus` ■ **Docs: 7** **Lang: en_core_web_md**

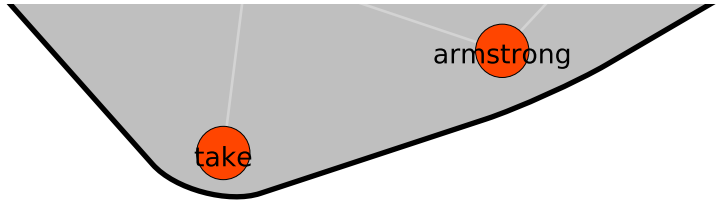
In [5]:
```
1 tn = Textnet(corpus.tokenized(), min_docs=1)
```

In [6]:

```
1 tn.plot(label_term_nodes=True,
2         label_doc_nodes=True,
3         show_clusters=True)
```

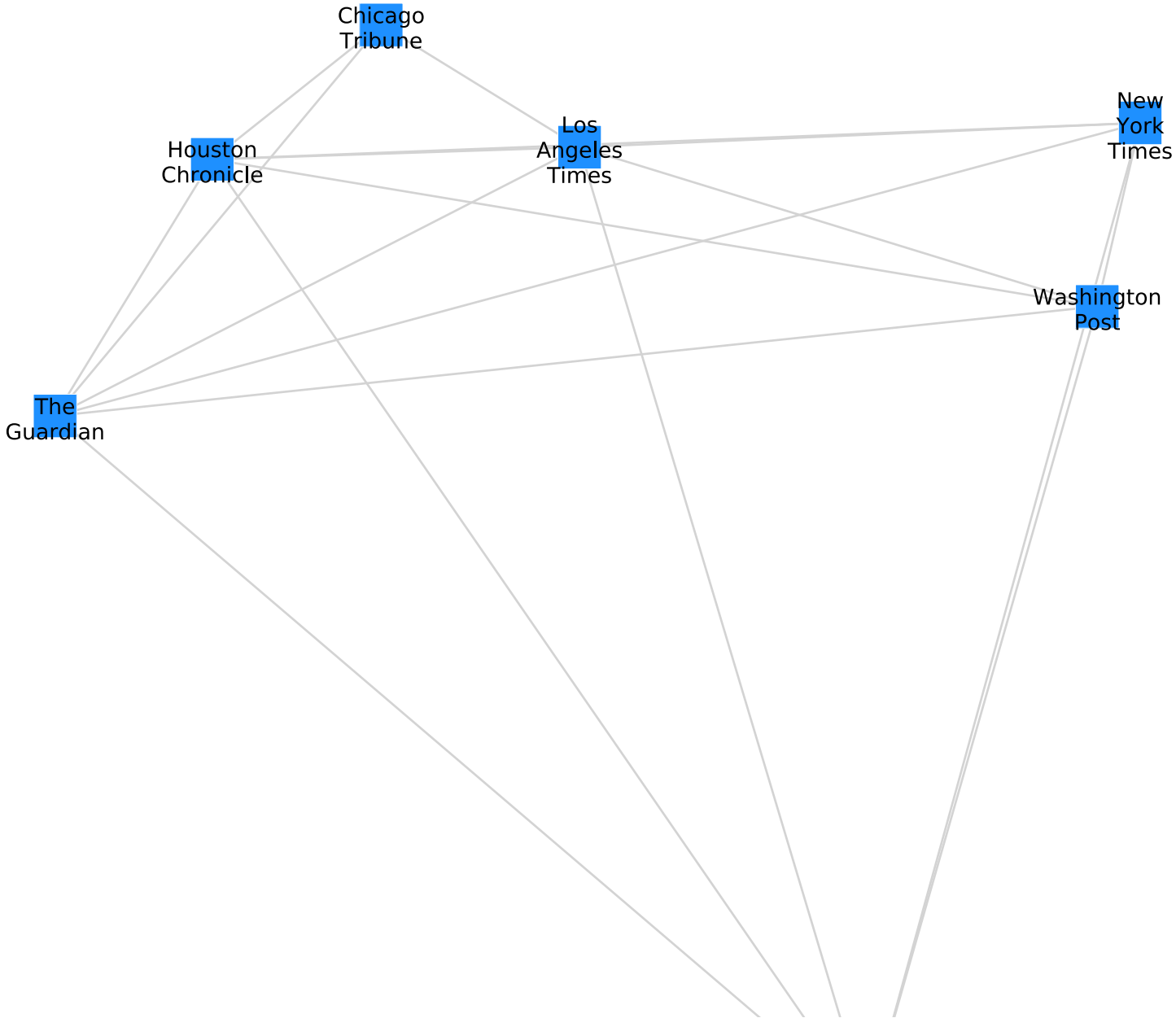
Out[6]:

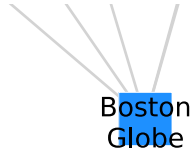



armstrong
take

In [7]:
```
papers = tn.project(node_type='doc')
papers.plot(label_nodes=True)
```

Out[7]:

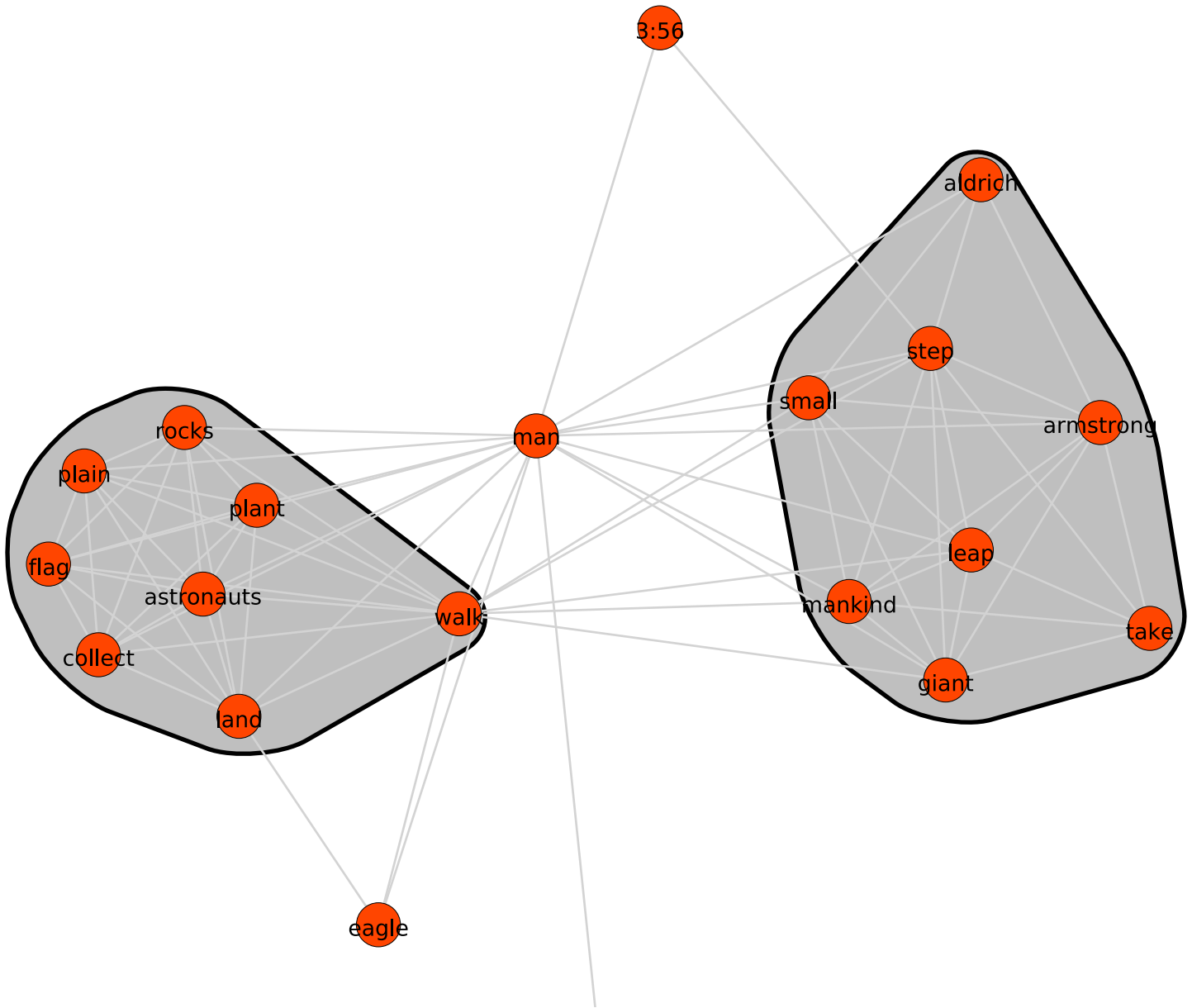



Boston
Globe

In [8]:

```
words = tn.project(node_type='term')
words.plot(label_nodes=True,
           show_clusters=True)
```

Out[8]:



In [9]:
```
1 papers.top_betweenness()
```

Out[9]:
```
Los Angeles Times    7.0
Boston Globe         0.0
Chicago Tribune      0.0
Houston Chronicle    0.0
New York Times       0.0
The Guardian         0.0
Washington Post      0.0
dtype: float64
```

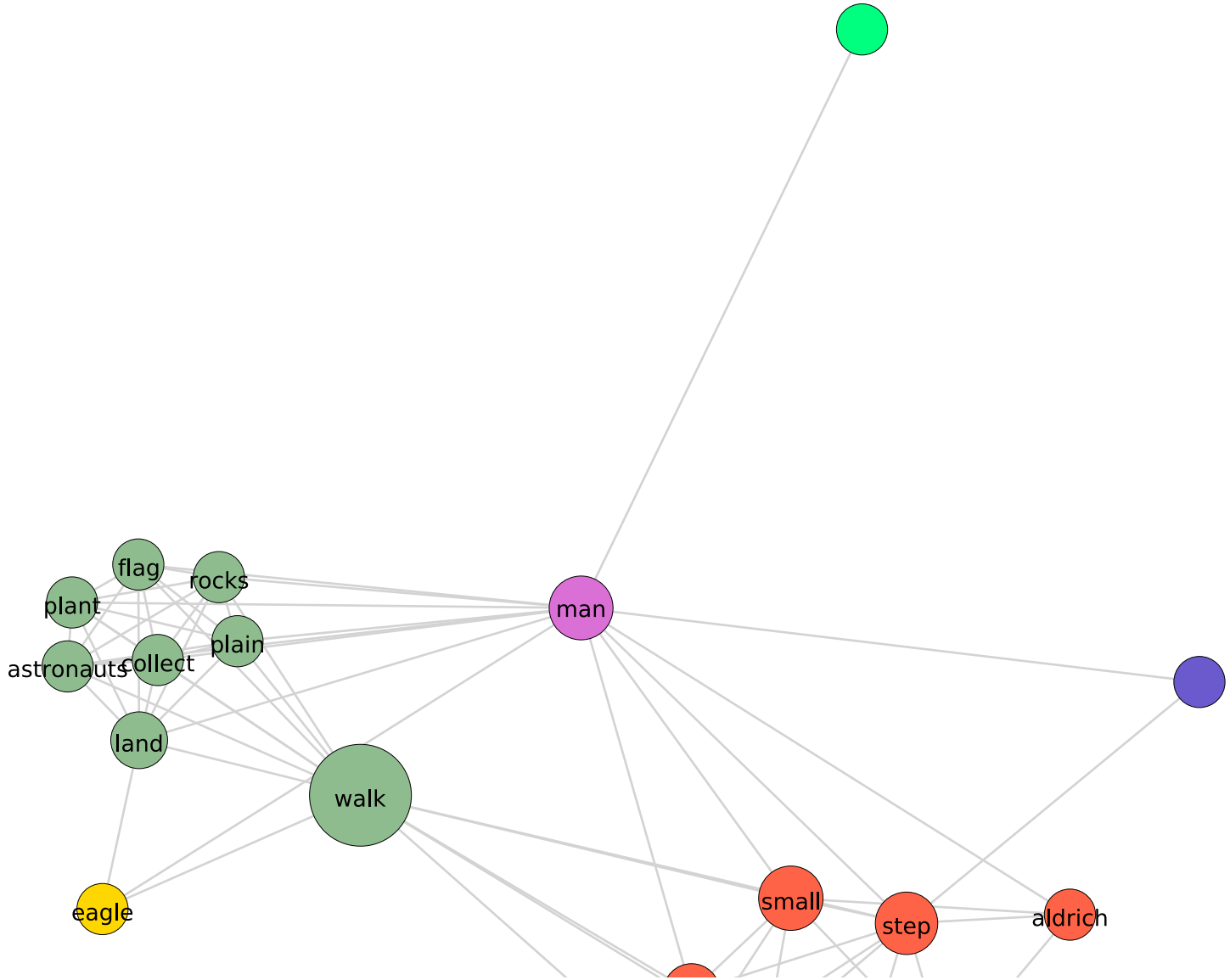
In [10]:
```
1 words.top_betweenness()
```

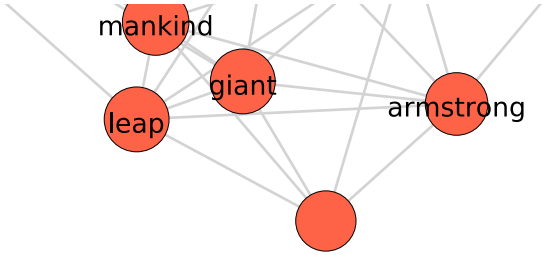
Out[10]:
```
walk         72.00
man          18.00
step         16.00
small        12.75
land          6.00
giant         6.00
leap          6.00
mankind       6.00
armstrong     3.25
plain         0.00
dtype: float64
```

In [11]:

```
words.plot(label_nodes=True,
           scale_nodes_by='betweenness',
           color_clusters=True,
           alpha=0.5,
           node_label_filter=lambda n: n.betweenness() > words.betweenness.median())
```

Out[11]:





In [ ]:
```
1
```

In [ ]:
```
1
```