**词云：wordcloud**

**ImageColorGenerator()**

#默认RGB取色，可以设置从背景图片生成颜色值

**WordCloud(builtins.object)**

#width : int (default=400)

#height : int (default=200)

#max\_words : number (default=200)

#background\_color : color value (default="black")

#min\_font\_size : int (default=4)

#max\_font\_size : int or None (default=None)

#mask : nd-array or None (default=None) 设置背景图片

#stopwords : set of strings or None 用于忽略某些常见词汇

生成词云, 可以用generate输入全部文本(中文不好分词),也可以我们计算好词频后使用generate\_from\_frequencies函数，例如[('词a', 100),('词b', 90),('词c', 80)]

**中文分词：jieba**

**1分词**

1）精确模式：试图将句子最精确地切开，适合文本分析；

2）全模式：速度非常快，但不能解决歧义；

3）搜索引擎模式：在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

**cut(sentence, cut\_all=False, HMM=True)**

#cut\_all 参数用来控制是否采用全模式(默认精确模式)； HMM 模型

**cut\_for\_search(sentence, HMM=True)**

#该方法适合用于搜索引擎构建倒排索引的分词，粒度比较细

#cut和cut\_for\_search返回的结构都是一个可迭代的 generator，可以用lcut 以及 lcut\_for\_search 直接返回 list

2关键词提取

基于 TF-IDF（term frequency–inverse document frequency） 算法

**analyse.extract\_tags(sentence, topK=20, withWeight=False, allowPOS=())**

#topK： 为返回几个 TF/IDF 权重最大的关键词，默认值为 20

#withWeight ： 为是否一并返回关键词权重值，默认值为 False

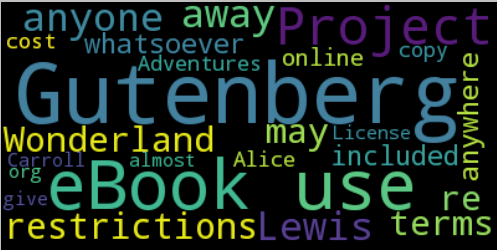
#allowPOS ： 仅包括指定词性的词，默认值为空，即不筛选

基于 TextRank 算法的关键词抽取

**analyse.textrank(sentence, topK=20, withWeight=False, allowPOS=(‘ns’, ‘n’, ‘vn’, ‘v’))**

**十行版**

**import** matplotlib.pyplot **as** plt  
**from** wordcloud **import** WordCloud  
**import** jieba  
  
text\_from\_file\_with\_apath = open(**''**).read()  
  
wordlist\_after\_jieba = jieba.cut(text\_from\_file\_with\_apath, cut\_all = **True**)  
wl\_space\_split = **" "**.join(wordlist\_after\_jieba)  
  
my\_wordcloud = WordCloud().generate(wl\_space\_split)  
  
plt.imshow(my\_wordcloud)  
plt.axis(**"off"**)  
plt.show()

****

**图片背景版**

**from** scipy.misc **import** imread  
**import** matplotlib.pyplot **as** plt  
**from** wordcloud **import** WordCloud, STOPWORDS, ImageColorGenerator  
**import** random  
**import** matplotlib **as** mpl  
  
mpl.rcParams[**"font.sans-serif"**] = [**"SimHei"**]  
text = open(**"wordCloudV2.txt"**).read()  
mask = imread(**"helmet.png"**)  
  
**def** grey\_color\_func(word, font\_size, position, orientation, random\_state=**None**, \*\*kwargs):  
 **return "hsl(0, 0%%, %d%%)"** % random.randint(60, 100)  
  
wc=WordCloud(background\_color=**"white"**,max\_words=2000,mask=mask,stopwords=STOPWORDS.add(**"said"**),max\_font\_size=40)  
wc.generate(text)  
default\_color = wc.to\_array()  
  
plt.title(**"灰色"**)  
plt.imshow(wc.recolor(color\_func=grey\_color\_func, random\_state=3))  
plt.axis(**"off"**)  
wc.to\_file(**"grey.jpg"**)  
  
plt.figure()  
plt.title(**"彩色"**)  
plt.imshow(default\_color)  
plt.axis(**"off"**)  
wc.to\_file(**"color.jpg"**)  
  
plt.show()