案例 4：t-SNE 可视化高维数据

t-SNE 是目前最为流行的一种高维数据降维的算法
全称为 t-distributed Stochastic Neighbor Embedding，翻译为 t 分布-随机邻近嵌入。

t-SNE 本质是一种嵌入模型，能够将高维空间中的数据映射到低维空间中，并保留数据集的局部特性。

缺点主要是占用内存较多、运行时间长。

t-SNE 变换后，如果在低维空间中具有可分性，则数据是可分的；如果在低维空间中不可分，则可能是因为数据集本身不可分，或者数据集中的数据不适合投影到低维空间。

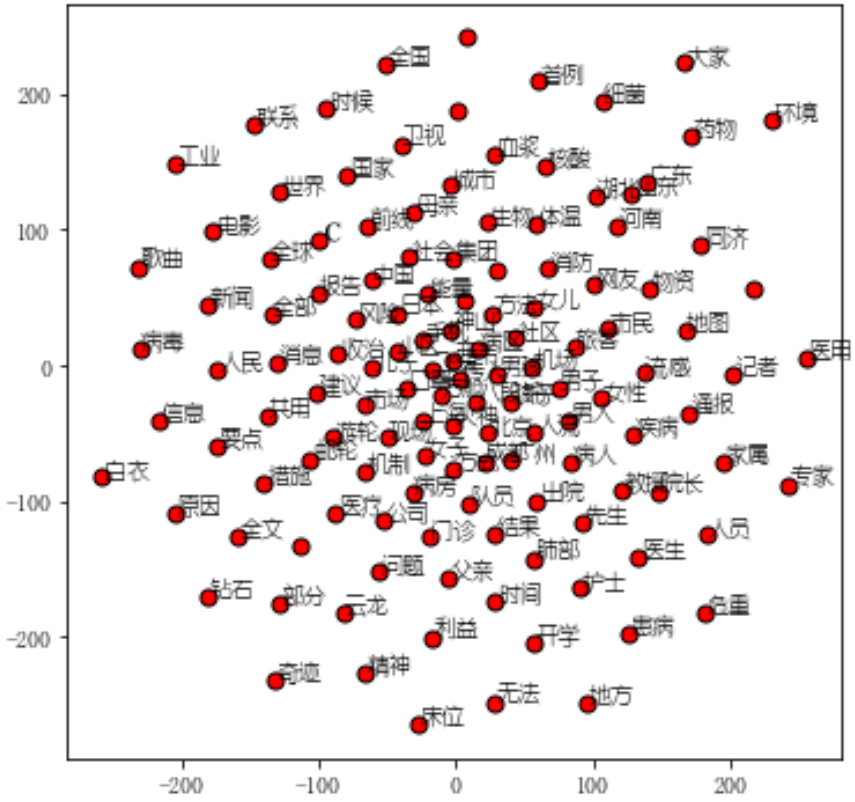
```
# 调用百度云 NLP 进行词向量表示
APP_ID = '11617353'
API_KEY = 'eV2R48IOWKcLgBrZwtf0ZF7N'
SECRET_KEY = 'HHtuGb3BPGaXAguPld5r9gfrY4xCCdzh'
client = AipNlp(APP_ID, API_KEY, SECRET_KEY)
# 获取 textrank2 中 Top200 词语的词向量
textrankList = textrank2.split(' ')
# 整理格式 原始格式：('简书', 0.9286492277441794) 后者是特征权重
words_list = []
word_vectors = []

for word in textrankList:
    try:
        data = client.wordEmbedding(word)
#         print(data)
        word_vector = data['vec']
        words_list.append(data['word'])
        word_vectors.append(word_vector)
    except:
        #print('No words:{}'.format(word))
        continue
word_vectors = np.array(word_vectors)
print(words_list)
print("words_list:",len(words_list))
print("word_vector:",word_vectors.shape)
```

```
['湖北', '全文', '病毒', '国家', '中国', '医生', '全国', '大家',
 '口罩', '病人', '新闻', '出院', '医疗', '前线', '专家', '人民',
 '护士', '原因', '孩子', '通报', '上海', '收治', '疾病', 'C', 
'首例', '', '飞沫', '广东', '药物', '北京', '男子', '母亲', '建
```

议', '方式', '火神', '广州', '人员', '公司', '病房', '时间', '院长', '游轮', '方舱', '机场', '儿子', '社区', '问题', '记者', '日本', '钻石', '开学', '门诊', '全球', '家属', '邮轮', '世界', '时候', '共用', '教授', '市民', '女儿', '危重', '血浆', '联系', '白衣', '[illegible]', '地方', '工业', '集团', '男孩', '现场', '风险', '奇迹', '女性', '网友', '电影', '传人', '小区', '物资', '歌曲', '精神', '要点', '利益', '八段锦', '[illegible]', '核酸', '全部', '女子', '体温', '流感', '报告', '结果', '城市', '信息', '病区', '床位', '措施', '队员', '男人', '人流', '山东', '手机', '市场', '神山', '消息', '方法', '医用', '[illegible]', '旅客', '肺部', '机制', '地图', '成都', '先生', '卫视', '患病', '云龙', '同济', '能量', '西韦', '生物', '消防', '无法', '细菌', '社会', '[illegible]', '部分', '父亲', '[illegible]', '河南', '环境']

```
words_list: 131
word_vector: (131, 1024)
#利用TSND将数据降维，可视化关键词

def plotTsne2D(word_vectors,words_list):
    tsne = TSNE(n_components=2, #int，可选（默认值：2）嵌入式空间的维度。
                random_state=0, #int或RandomState实例或None（默认）。伪随机数发生器种子控制
                n_iter=600, #int，可选（默认值：1000）优化的最大迭代次数。至少应该200。
                perplexity=15   #浮点型，可选（默认：30）较大的数据集通常需要更大的perplexity。考虑选择一个介于5和50之间的值。由于t-SNE对这个参数非常不敏感，所以选择并不是非常重要。
               )
    # 在控制台输出过程中，默认小数会以科学计数法的形式输出，若不需要加上下面这句
    np.set_printoptions(suppress=True)
```

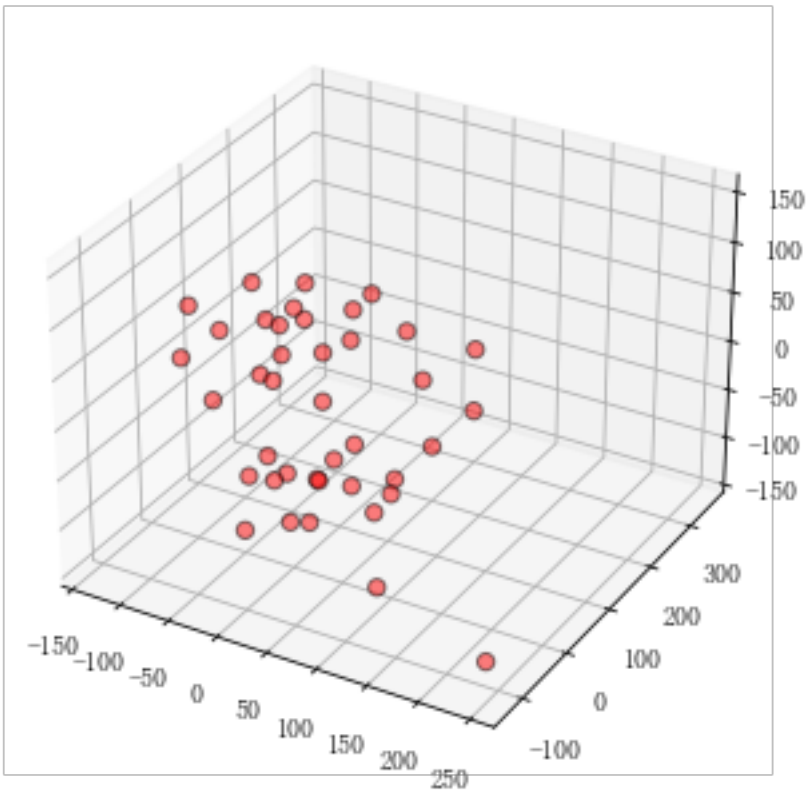
```
    T = tsne.fit_transform(word_vectors)
    labels = words_list

    plt.figure(figsize=(6,6))
    plt.scatter(T[:,0],T[:,1], s=50, c='r',edgecolors='k')
    print(T[0:5])
    for label,x,y in zip(labels,T[:,0],T[:,1]):
        plt.annotate(label,xy=(x+1,y+1),xytext=(0,0),textcoords='offset points')


matplotlib.rc("font",family='YouYuan')
# 解决中文显示问题
# plt.rcParams['font.sans-serif'] = ['Kai']
plt.rcParams['axes.unicode_minus'] = False

plotTsne2D(word_vectors,words_list)
[[ 101.84345   124.70265 ]
 [-158.6042   -125.55698 ]
 [-231.07275    11.329214]
 [ -79.33789   140.73094 ]
 [ -61.97661    63.342842]]
```

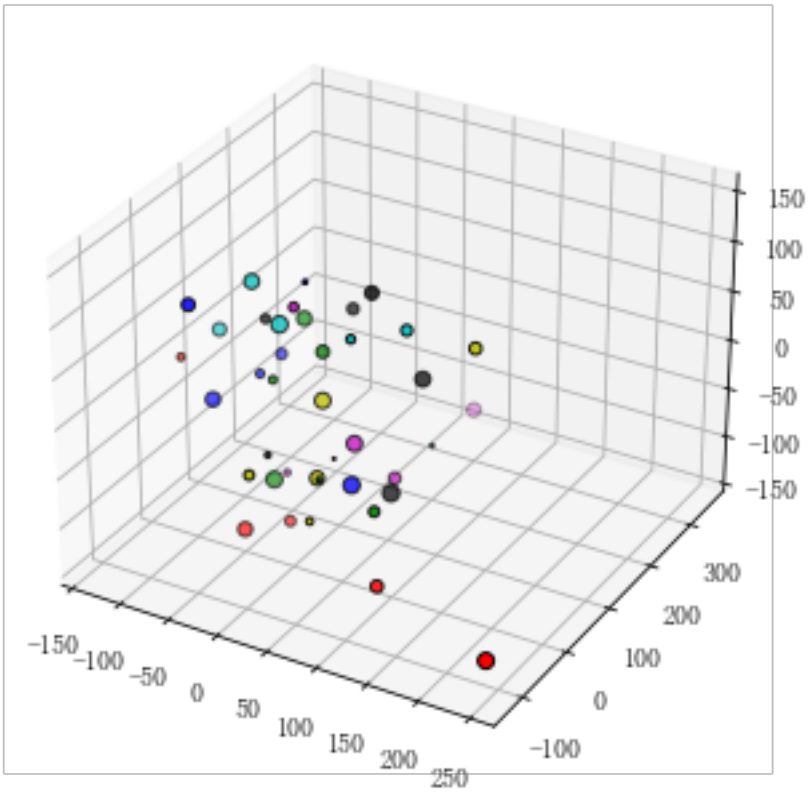



```
from mpl_toolkits.mplot3d import Axes3D

tsne = TSNE(n_components=3,
```

```
                    random_state=0, #int 或 RandomState 实例或 None（默认）。伪随机数发生器种子控制
                    n_iter=600, #int，可选（默认值：1000）优化的最大迭代次数。至少应该 200。
                    perplexity=15  #浮点型，可选（默认：30）较大的数据集通常需要更大的 perplexity。考虑选择一个介于 5 和 50 之间的值。由于 t-SNE 对这个参数非常不敏感，所以选择并不是非常重要。
                    )
# 在控制台输出过程中，默认小数会以科学计数法的形式输出，若不需要加上下面这句
np.set_printoptions(suppress=True)
T = tsne.fit_transform(word_vectors)
T = T[:40,:]
# Axes3D.scatter(xs, ys, zs=0, zdir='z', s=20, c=None, depthshade=True, *args, **kwargs)
ax = Axes3D(plt.figure())
NumP = T.shape[0]
x = T[:,0]
y = T[:,1]
z = T[:,2]
#-----画一个平行于 xy 的平面-----#
ax.scatter(x,y,z,s=40,c='r',edgecolor='k',alpha=0.5)

#-------改变大小和颜色--------#
ColorBase = ('r','g','b','c','k','m','y')
ax = Axes3D(plt.figure())  # 重新创建一个
S = [i for i in range(1,NumP+1)]  # It is a scalar or an array of the same length as x and y.
def GetColor(N):
    NumColor = len(ColorBase)
    Color = [ColorBase[0]]
    Iter = 1
    for i in range(N-1):
        if Iter>=NumColor:
            Iter = 0
        Color.append(ColorBase[Iter])
        Iter = Iter+1
    return(Color)
Color = GetColor(NumP)
print(len(Color),Color)  # c can be a single color format string;
# or a sequence of color specifications of length N;
# or a sequence of N numbers to be mapped to colors using the cmap and norm specified via kwargs (see below).
```

```
# Note that c should not be a single numeric RGB or RGBA sequence becaus
e that is indistinguishable from an array of values to be colormapped.
# c can be a 2-D array in which the rows are RGB or RGBA, however, inclu
ding the case of a single row to specify the same color for all points.
ax.scatter(x,y,z,s=S,c=Color,marker='o',edgecolors='k')
plt.show()
40 ['r', 'g', 'b', 'c', 'k', 'm', 'y', 'r', 'g', 'b', 'c',
'k', 'm', 'y', 'r', 'g', 'b', 'c', 'k', 'm', 'y', 'r', 'g',
 'b', 'c', 'k', 'm', 'y', 'r', 'g', 'b', 'c', 'k', 'm', 'y
', 'r', 'g', 'b', 'c', 'k']
```




150
100
50
0
−50
−100
−150
300
200
100
0
−100
−150
−100
−50
0
50
100
150
200
250