In [1]:
```
import pandas as pd
import csv
import jieba
import gensim
from gensim import corpora, models, similarities
from gensim.models import CoherenceModel, LdaModel
from gensim.models.doc2vec import Doc2Vec, LabeledSentence
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
#加载文本
df0 = pd.read_csv(r'C:\Users\23031\Desktop\长沙市_内容_0.csv', encoding='gb18030', sep=',', header=None)
df1 = pd.read_csv(r'C:\Users\23031\Desktop\长沙市_内容_1.csv', encoding='gb18030', sep=',', header=None)
df2 = pd.read_csv(r'C:\Users\23031\Desktop\长沙市_内容_2.csv', encoding='gb18030', sep=',', header=None)
df = df0.append(df1)
df = df.append(df2)

#加载停用词
sw = pd.read_csv(r'C:\Users\23031\Desktop\停用词.txt',
                 encoding='utf-8', sep='\n', quoting=csv.QUOTE_NONE, header=None)
```

In [3]:
```
#将文档分词并去除停用词
stop_list = sw[0].tolist()
df_cut = df0[0].apply(lambda x : [i for i in jieba.lcut(x) if i not in stop_list])
```

```
Building prefix dict from the default dictionary ...
Dumping model to file cache C:\Users\23031\AppData\Local\Temp\jieba.cache
Loading model cost 0.946 seconds.
Prefix dict has been built successfully.
```

In [4]:
```
for line in df_cut:#去除无关字符串
    while True:
        if '\n' in line:
            line.remove('\n')
        elif '\t' in line:
            line.remove('\t')
        elif ' ' in line:
            line.remove(' ')
        elif '\r' in line:
            line.remove('\r')
        elif '\r\n' in line:
            line.remove('\r\n')
        else:
            break
```

In [5]:
```
words=[]
for content in df_cut:
    words.extend(content)

#创建分词数据框
corpus = pd.DataFrame(words, columns=['word'])
corpus['cnt'] = 1

#分组统计
g = corpus.groupby(['word']).agg({'cnt': 'count'}).sort_values('cnt', ascending=False)

g.head(10)
```

Out[5]:

word	cnt
	95884
业主	59170
装修	37384
开发商	35132
相关	30251
小区	27935
建设	21649
部门	20575
项目	19571
造价	19228

In [16]:
```
g.to_csv('C:\\Users\\23031\\Desktop\\词频统计.csv',
         mode='a', encoding='gb18030', sep=',')
```

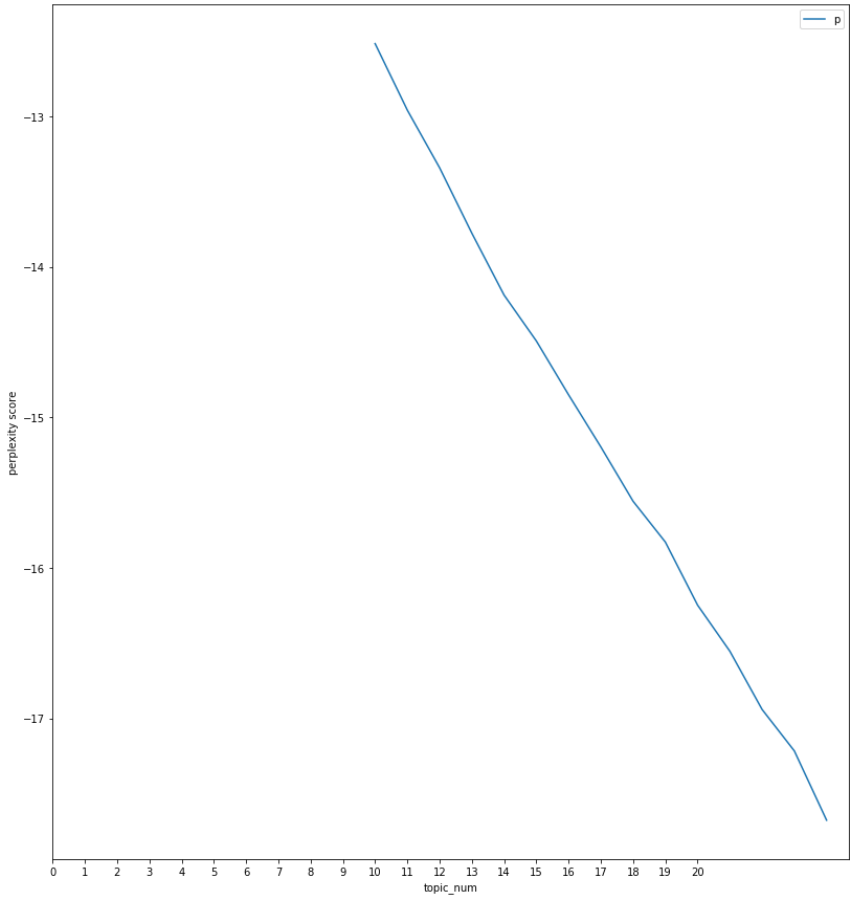
In [12]:
```
df_cut.to_csv('C:\\Users\\23031\\Desktop\\分词.csv',
              mode='a', encoding='gb18030', sep=',', header=False)
```

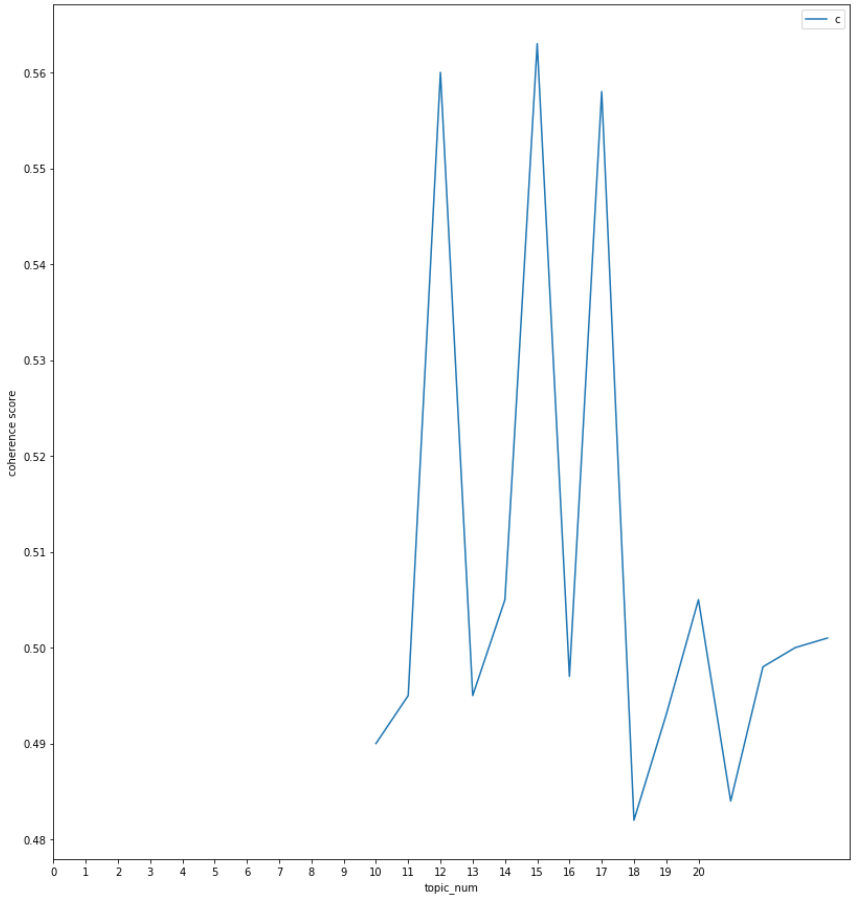
In [8]:
```
dictionary = corpora.Dictionary(df_cut)#制作词袋

#将分词列表转换为索引，并计数
corpus = [dictionary.doc2bow(text) for text in df_cut]

#计算tf-idf值
corpus_tfidf = models.TfidfModel(corpus)[corpus]
```

In [10]:
```
model_list = []
p_list = []
c_list = []

for topic_num in range(10,25):
    print('完成一个')
    lda = LdaModel(corpus_tfidf, num_topics=topic_num, id2word=dictionary)
                # alpha=0.01, eta=0.01, minimum_probability=0.001,update_every=1, chunksize=100, passes=1
    model_list.append(topic_num)

    #计算困惑度
    p_values = lda.log_perplexity(corpus_tfidf)
    p_list.append(p_values)
    #print('%d 个主题的Perplexity为: ' % topic_num, p_values)

    #计算一致性
    cmodel = CoherenceModel(model=lda,texts=df_cut,dictionary=dictionary,coherence='c_v')
    c_list.append(round(cmodel.get_coherence(),3))
    #print('%d 个主题的Coherence为: ' % (topic_num), round(cmodel.get_coherence(),3))
```

```
完成一个
完成一个
完成一个
完成一个
完成一个
完成一个
完成一个
完成一个
完成一个
完成一个
完成一个
完成一个
完成一个
完成一个
完成一个
```

In [11]:
```
#困惑度和一致性折线图
plt.figure(figsize=(30,15))
plt.subplot(1,2,1)
plt.plot(model_list,p_list)
plt.xticks(np.linspace(0, 20, 21))
plt.xlabel('topic_num')
plt.ylabel('perplexity score')
plt.legend(('perplexity_values'), loc='best')

plt.subplot(1,2,2)
plt.plot(model_list,c_list)
plt.xticks(np.linspace(0, 20, 21))
plt.xlabel('topic_num')
plt.ylabel('coherence score')
plt.legend(('coherence_values'), loc='best')

plt.show()
```


p
perplexity score
topic_num


c
coherence score
topic_num

In [12]:
```
topic_num = 15 #主题数（根据折线图确定）
lda = models.LdaModel(corpus_tfidf,num_topics=topic_num,id2word=dictionary) #LDA模型训练
for i in range(topic_num):
    print('主题%d:' % (i+1))
    print(lda.show_topic(i))#输出每个主题
```

```
主题1:
[('消防栓', 0.033467446), ('供水', 0.008969501), ('水业', 0.0056823874), ('用水', 0.0046558403), ('96533', 0.0034209024), ('阀门', 0.0030601532), ('自来水', 0.002579365), ('停水', 0.0022529112), ('水压', 0.0020421096), ('水厂', 0.0015174877)]
主题2:
[('梅岭', 0.0007537427), ('网约车', 0.00056061946), ('滴滴', 0.0004354866), ('电杆', 0.00030221223), ('平川', 0.00027524002), ('东湖', 0.00024361126), ('既有', 0.00015868875), ('汽配城', 0.00013187868), ('接续', 0.00012008343), ('转出', 8.671256e-05)]
主题3:
[('奥克斯', 0.00879153), ('万家', 0.007519541), ('湖湘', 0.005968581), ('丽', 0.0048333164), ('公交', 0.004404626), ('重罚', 0.0033926757), ('西侧', 0.0033763433), ('北路', 0.003257283), ('须知', 0.0031472652), ('车辆', 0.0031072902)]
主题4:
[('兑付', 0.002926335), ('工程量', 0.002567016), ('计价', 0.002382044), ('芒果', 0.0020144456), ('零售价格', 0.0019525693), ('主材', 0.0019452738), ('执行长', 0.0019104346), ('套内', 0.0009514859), ('竹塘', 0.00092797424), ('金辉', 0.00087184104)]
主题5:
[('奥林匹克', 0.015093059), ('基准', 0.008236988), ('缴满', 0.0018507342), ('推算', 0.0016664265), ('向前', 0.0015972181), ('水墨', 0.0014500829), ('林溪', 0.0013091201), ('长望', 0.0012121374), ('右转', 0.00094592624), ('余易贷', 0.0008594935)]
主题6:
[('筹备组', 0.016271852), ('批', 0.006006995), ('干', 0.004682583), ('迹象', 0.0042856033), ('何在', 0.004014285), ('站台', 0.0038282939), ('小区', 0.003391977), ('开放式', 0.003108004), ('小学', 0.0030820097), ('岳麓', 0.003004544)]
主题7:
[('造价', 0.022847211), ('装修', 0.0167151), ('咨询机构', 0.015607031), ('第三方', 0.015573169), ('核算', 0.013317245), ('价格', 0.012348606), ('全', 0.010567292), ('工程造价', 0.006325622), ('复核', 0.006092777), ('公正', 0.005705292)]
主题8:
[('\xa0', 0.0067951926), ('监制', 0.0061024185), ('业委会', 0.0056538717), ('均价', 0.005283363), ('包', 0.005097702), ('阳光', 0.0044649686), ('热线', 0.0041249883), ('新奥', 0.0038380807), ('投票', 0.0037155321), ('街道', 0.0037089798)]
主题9:
[('招录', 0.0016599398), ('北延线', 0.00090966484), ('北延', 0.0008282598), ('贴纸', 0.0006422577), ('合能', 0.0005881597), ('绿道', 0.0005588651), ('独生子女', 0.0005043403), ('三馆', 0.00049407664), ('中转站', 0.00048880454), ('爱人', 0.00044828627)]
主题10:
[('东方明珠', 0.0023968332), ('明发', 0.0012725312), ('建发央著', 0.0006542168), ('体育中心', 0.00039332535), ('长丰', 0.00031756153), ('庄园', 0.00030991473), ('白鹤', 0.00025944877), ('步步高', 0.00024922544), ('电缆', 0.00022942656), ('10kV', 0.000210606066)]
主题11:
[('松雅湖', 0.016981287), ('一平', 0.008901623), ('号线', 0.0050288094), ('地铁', 0.0044932924), ('轨道交通', 0.004247319), ('美的', 0.0036762608), ('规划', 0.0035090738), ('线网', 0.0028443567), ('翰城', 0.0026721987), ('轨道', 0.002491082)]
主题12:
```

```
[('级', 0.008456824), ('装修', 0.0073116883), ('刚需', 0.0068942155), ('摇号', 0.0066436646), ('首套', 0.0062052393), ('相近', 0.005829303), ('第三项', 0.0058273356), ('开发商', 0.0051383744), ('\xa0', 0.0051304037), ('公积金', 0.0049063712)]
主题13:
[('航天', 0.0010757288), ('琨', 0.00086662825), ('瑜', 0.00085559057), ('狗', 0.0008142984), ('犬', 0.0005545889), ('景区', 0.0004963255), ('金源', 0.000434467), ('养犬', 0.00037709947), ('续期', 0.0002691308), ('拆墙', 0.0002516709)]
主题14:
[('楚天', 0.023237772), ('入伍', 0.0014189101), ('城镇职工', 0.0011233875), ('生育', 0.0010684476), ('鑫苑', 0.0010666715), ('富力', 0.00095859007), ('现役军人', 0.00093314063), ('山南路', 0.0008324897), ('服役', 0.0008320269), ('产假', 0.00081836473)]
主题15:
[('缤纷', 0.011000474), ('幼儿园', 0.008712724), ('世界', 0.006344532), ('天著', 0.0053932043), ('万润', 0.0053215753), ('入学', 0.00480416), ('84013149', 0.0046173595), ('教育局', 0.0041252756), ('小学', 0.003675304), ('实际操作', 0.0030103866)]
```

In [*]:
```
import pyLDAvis
import pyLDAvis.gensim
vis_data = pyLDAvis.gensim.prepare(lda,corpus,dictionary)
pyLDAvis.show(vis_data)
```

In [ ]:
```
```