法律声明

- 本课件包括演示文稿、示例、代码、题库、视频和声音等内容，小象学院和主讲老师拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意及内容，我们保留一切通过法律手段追究违反者的权利。

- 课程详情请咨询
  - 微信公众号：小象
  - 新浪微博：ChinaHadoop

主题模型

邹博

主要内容

- LDA开源实现库介绍
- LDA1.0.4/Gensim的使用
- 复习：
  - TF-IDF
  - 相似度计算

LDA的实现

- LDA-C：David Blei，C实现，VBEM参数估计
  - http://www.cs.princeton.edu/~blei/lda-c/index.html
- GibbsLDA++/JGibbLDA ：C/C++实现/Java实现
  - http://gibbslda.sourceforge.net/ http://jgibblda.sourceforge.net/
  - Xuan-Hieu Phan/Cam-Tu Nguyen，输入输出一致
- Matlab Topic Modeling Toolbox 1.4，Mark Steyvers，Gibbs采样
  - http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm
- Gensim：Online VB
  - 官网：http://radimrehurek.com/gensim/index.html
  - github：http://www.cs.columbia.edu/~blei/topicmodeling_software.html
- Scikit-learn：sklearn.decomposition.LatentDirichletAllocation/Online VB
- LDA/Online VB：https://pypi.python.org/pypi/lda
- LDA不完全列表：
  - http://www.cs.columbia.edu/~blei/topicmodeling_software.html

例：Gensim的安装

```
C:\Users\zou>pip install gensim
Collecting gensim
  Downloading gensim-0.13.2-cp27-cp27m-win32.whl (4.2MB)
    100% |████████████████████████████████| 4.2MB 141kB/s
Requirement already satisfied (use --upgrade to upgrade): scipy>=0.7.0 in c:\python27\lib\site-packages (from gensim)
Requirement already satisfied (use --upgrade to upgrade): six>=1.5.0 in c:\python27\lib\site-packages (from gensim)
Requirement already satisfied (use --upgrade to upgrade): numpy>=1.3 in c:\python27\lib\site-packages (from gensim)
Collecting smart-open>=1.2.1 (from gensim)
  Downloading smart_open-1.3.4.tar.gz
Collecting boto>=2.32 (from smart-open>=1.2.1->gensim)
  Downloading boto-2.42.0-py2.py3-none-any.whl (1.3MB)
    100% |████████████████████████████████| 1.4MB 249kB/s
Collecting bz2file (from smart-open>=1.2.1->gensim)
  Downloading bz2file-0.98.tar.gz
Collecting requests (from smart-open>=1.2.1->gensim)
  Downloading requests-2.11.1-py2.py3-none-any.whl (514kB)
    100% |████████████████████████████████| 522kB 333kB/s
Installing collected packages: boto, bz2file, requests, smart-open, gensim
  Running setup.py install for bz2file ... done
  Running setup.py install for smart-open ... done
Successfully installed boto-2.42.0 bz2file-0.98 gensim-0.13.2 requests-2.11.1 smart-open-1.3.4
```

TF-IDF

```
Text =
[['human', 'machine', 'interface', 'lab', 'abc', 'computer', 'applications'],
 ['survey', 'user', 'opinion', 'computer', 'system', 'response', 'time'],
 ['eps', 'user', 'interface', 'management', 'system'],
 ['system', 'human', 'system', 'engineering', 'testing', 'eps'],
 ['relation', 'user', 'perceived', 'response', 'time', 'error', 'measurement'],
 ['generation', 'random', 'binary', 'unordered', 'trees'],
 ['intersection', 'graph', 'paths', 'trees'],
 ['graph', 'minors', 'iv', 'widths', 'trees', 'well', 'quasi', 'ordering'],
 ['graph', 'minors', 'survey']]
TF-IDF:
[(0, 0.4301019571350565), (1, 0.4301019571350565), (2, 0.4301019571350565), (3, 0.4301019571350565), (4, 0.2944198962221451), (5
[(4, 0.3726494271826947), (7, 0.27219160459794917), (8, 0.3726494271826947), (9, 0.27219160459794917), (10, 0.3726494271826947),
[(6, 0.438482464916089), (7, 0.32027755044706185), (9, 0.32027755044706185), (13, 0.6405551008941237), (14, 0.438482464916089)]
[(5, 0.3449874408519962), (7, 0.5039733231394895), (14, 0.3449874408519962), (15, 0.5039733231394895), (16, 0.5039733231394895)]
[(9, 0.21953536176370683), (10, 0.30055933182961736), (12, 0.30055933182961736), (17, 0.43907072352741366), (18, 0.4390707235274
[(21, 0.48507125007266594), (22, 0.48507125007266594), (23, 0.48507125007266594), (24, 0.48507125007266594), (25, 0.242535625036
[(25, 0.31622776601683794), (26, 0.31622776601683794), (27, 0.6324555320336759), (28, 0.6324555320336759)]
[(25, 0.20466057569885868), (26, 0.20466057569885868), (29, 0.2801947048062438), (30, 0.40932115139771735), (31, 0.4093211513977
[(8, 0.6282580468670046), (26, 0.45889394536615247), (29, 0.6282580468670046)]
```

LSI

```
LSI Model:
[[(0, 0.34057117986841989), (1, -0.20602251622679696)],
 [(0, 0.69330400021715577), (1, 0.0072327583903918488)],
 [(0, 0.59026076703897357), (1, -0.35260469490855789)],
 [(0, 0.52149018218251453), (1, -0.33887976154055377)],
 [(0, 0.39533193176354431), (1, -0.059192853366596486)],
 [(0, 0.036353173528493307), (1, 0.18146550208818862)],
 [(0, 0.14709012328778862), (1, 0.49432948127822229)],
 [(0, 0.21407117317565286), (1, 0.640645666445394)],
 [(0, 0.40066568318170664), (1, 0.64131082990940158)]]
LSI Topics:
[(0,
  u'0.400*"system" + 0.318*"survey" + 0.290*"user" + 0.274*"eps" + 0.236*"management"'),
 (1,
  u'0.421*"minors" + 0.420*"graph" + 0.293*"survey" + 0.239*"trees" + 0.226*"intersection"')]
```

思考

- LSI/LFM/ICA的关系
- LSI和pLSA的关系

相似度

```
Similarity:
[array([ 1.        ,  0.85017949,  0.99998462,  0.99948108,  0.92283762,
        -0.33944285, -0.2520774 , -0.21974573,  0.01438823], dtype=float32),
 array([ 0.85017949,  1.        ,  0.85309052,  0.83277911,  0.98737705,
         0.20664607,  0.29518002,  0.32680073,  0.53867108], dtype=float32),
 array([ 0.99998462,  0.85309052,  1.        ,  0.99928677,  0.92496276,
        -0.33421332, -0.24669874, -0.214324  ,  0.01994151], dtype=float32),
 array([ 0.99948108,  0.83277911,  0.99928677,  1.        ,  0.90995121,
        -0.36956567, -0.28311783, -0.25105584, -0.01782739], dtype=float32),
 array([ 0.92283762,  0.98737705,  0.92496276,  0.90995121,  1.        ,
         0.04906873,  0.14012395,  0.1729846 ,  0.39842743], dtype=float32),
 array([-0.33944285,  0.20664607, -0.33421332, -0.36956567,  0.04906873,
         1.        ,  0.99581695,  0.99222624,  0.93564534], dtype=float32),
 array([-0.2520774 ,  0.29518002, -0.24669874, -0.28311783,  0.14012395,
         0.99581695,  1.        ,  0.99944651,  0.96397996], dtype=float32),
 array([-0.21974573,  0.32680073, -0.214324  , -0.25105584,  0.1729846 ,
         0.99222624,  0.99944651,  0.99999994,  0.97229445], dtype=float32),
 array([ 0.01438823,  0.53867108,  0.01994151, -0.01782739,  0.39842743,
         0.93564534,  0.96397996,  0.97229445,  1.        ], dtype=float32)]
```

主题和主题分布

LDA Model:

Document-Topic:

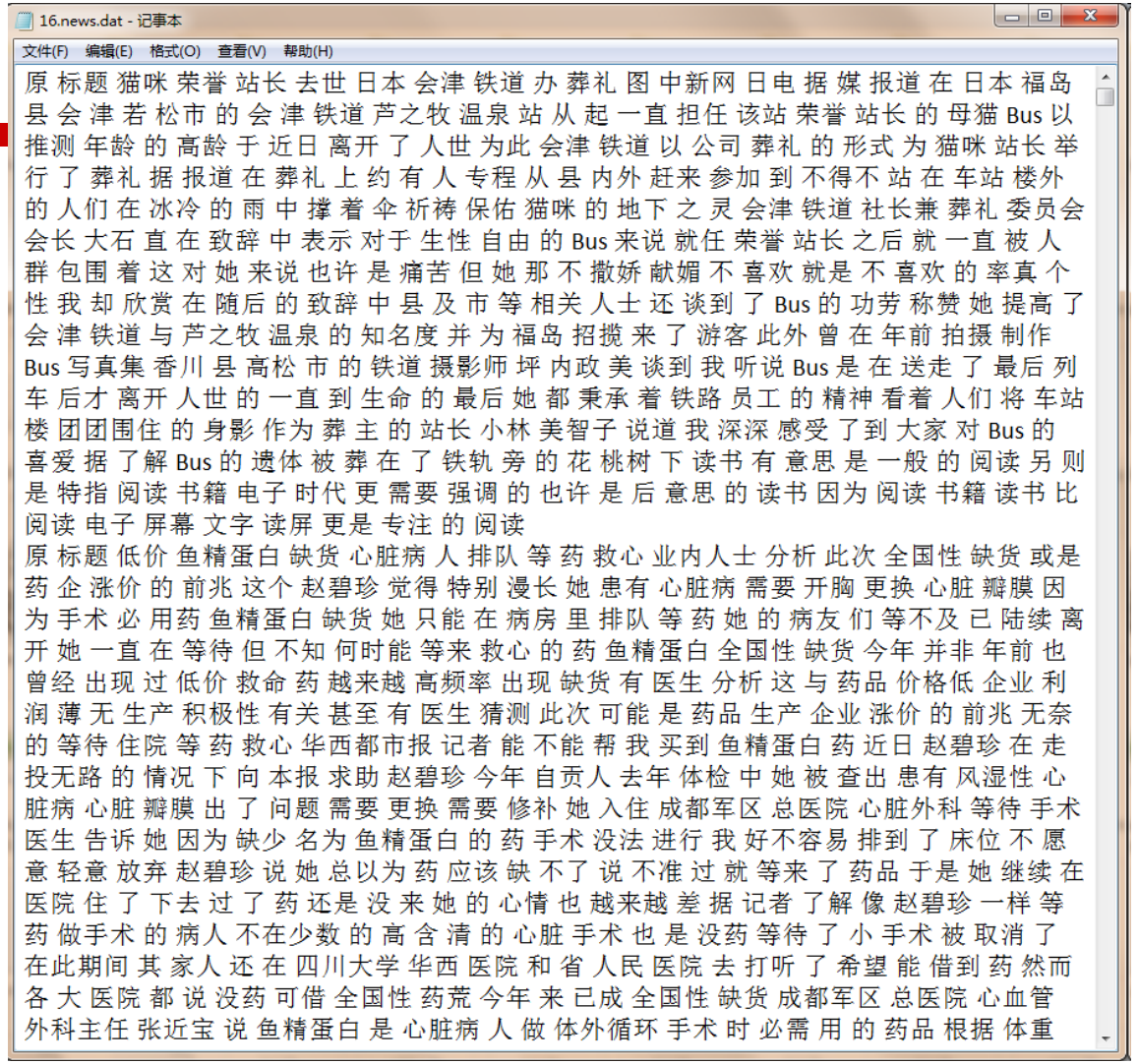
```
[[(0, 0.68548441915170544), (1, 0.31451558084829462)],
 [(0, 0.65732202058761513), (1, 0.34267797941238493)],
 [(0, 0.67101883898793013), (1, 0.32898116101206987)],
 [(0, 0.29774557750241137), (1, 0.70225442249758874)],
 [(0, 0.55150516193766697), (1, 0.44849483806233303)],
 [(0, 0.25456933670287446), (1, 0.7454306632971256)],
 [(0, 0.67476418767307922), (1, 0.32523581232692073)],
 [(0, 0.29509659300584296), (1, 0.7049034069941571)],
 [(0, 0.69445879658152987), (1, 0.30554120341847024)]]
```

```
Topic 0
[(u'survey', 0.042573497130974247),
 (u'minors', 0.03943557671036535),
 (u'graph', 0.038776707760135178),
 (u'system', 0.034575198665359616),
 (u'trees', 0.032742027152788719),
 (u'opinion', 0.031680224783503845),
 (u'generation', 0.031141365123546434),
 (u'unordered', 0.030981049002428096),
 (u'time', 0.030911535753312992),
 (u'random', 0.03090147631201922)]
Topic 1
[(u'system', 0.037724259436260198),
 (u'eps', 0.03524885080697393),
 (u'interface', 0.034303635122775261),
 (u'intersection', 0.03398428810730824),
 (u'user', 0.033982740385072041),
 (u'management', 0.033477115230294417),
 (u'human', 0.032957111835837112),
 (u'paths', 0.032333361709319365),
 (u'engineering', 0.030715385582341159),
 (u'computer', 0.030706245324429286)]
```

LDA计算的相似度

```
Similarity:
[array([ 0.99999994,  0.79683411,  0.99871153,  0.9988395 ,  0.99509394,
         0.68600154,  0.98457313,  0.66293609,  0.71091771], dtype=float32),
 array([ 0.79683411,  1.        ,  0.82646847,  0.82500935,  0.85270071,
         0.98624408,  0.89025998,  0.98059875,  0.99140108], dtype=float32),
 array([ 0.99871153,  0.82646847,  1.        ,  0.99999666,  0.9988324 ,
         0.72204095,  0.99218392,  0.70007479,  0.74569058], dtype=float32),
 array([ 0.9988395 ,  0.82500935,  0.99999666,  1.        ,  0.99870408,
         0.72024882,  0.99185783,  0.69822526,  0.74396449], dtype=float32),
 array([ 0.99509394,  0.85270071,  0.9988324 ,  0.99870408,  0.99999994,
         0.75462055,  0.99705368,  0.7337535 ,  0.77700794], dtype=float32),
 array([ 0.68600154,  0.98624408,  0.72204095,  0.72024882,  0.75462055,
         1.        ,  0.80272919,  0.9995119 ,  0.9993937 ], dtype=float32),
 array([ 0.98457313,  0.89025998,  0.99218392,  0.99185783,  0.99705368,
         0.80272919,  1.        ,  0.78370738,  0.82300484], dtype=float32),
 array([ 0.66293609,  0.98059875,  0.70007479,  0.69822526,  0.7337535 ,
         0.9995119 ,  0.78370738,  1.        ,  0.99781829], dtype=float32),
 array([ 0.71091771,  0.99140108,  0.74569058,  0.74396449,  0.77700794,
         0.9993937 ,  0.82300484,  0.99781829,  1.        ], dtype=float32)]
```

网易新闻语料


16.news.dat - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

原 标题 猫咪 荣誉 站长 去世 日本 会津 铁道 办 葬礼 图 中新网 日电 据 媒 报道 在 日本 福岛 县 会 津 若 松市 的 会 津 铁道 芦之牧 温泉 站 从 起 一直 担任 该站 荣誉 站长 的 母猫 Bus 以 推测 年龄 的 高龄 于 近日 离开 了 人世 为此 会津 铁道 以 公司 葬礼 的 形式 为 猫咪 站长 举 行 了 葬礼 据 报道 在 葬礼 上 约有 人 专程 从 县 内外 赶来 参加 到 不得不 站 在 车站 楼外 的 人们 在 冰冷 的 雨 中 撑着 伞 祈祷 保佑 猫咪 的 地下 之 灵 会津 铁道 社长兼 葬礼 委员会 会长 大石 直 在 致辞 中 表示 对于 生性 自由 的 Bus 来说 就任 荣誉 站长 之后 就 一直 被 人 群 包围 着 这 对 她 来说 也许 是 痛苦 但 她 那 不 撒娇 献媚 不 喜欢 就是 不 喜欢 的 率真 个 性 我 却 欣赏 在 随后 的 致辞 中 县 及 市 等 相关 人士 还 谈到 了 Bus 的 功劳 称赞 她 提高 了 会 津 铁道 与 芦之牧 温泉 的 知名度 并 为 福岛 招揽 来 了 游客 此外 曾 在 年前 拍摄 制作 Bus 写真集 香川 县 高松 市 的 铁道 摄影师 坪 内政 美 谈到 我 听说 Bus 是 在 送走 了 最后 列 车 后才 离开 人世 的 一直 到 生命 的 最后 她 都 秉承 着 铁路 员工 的 精神 看着 人们 将 车站 楼 团团围住 的 身影 作为 葬 主 的 站长 小林 美智子 说道 我 深深 感受 了到 大家 对 Bus 的 喜爱 据 了解 Bus 的 遗体 被 葬 在 了 铁轨 旁 的 花 桃树 下 读书 有 意思 是 一般 的 阅读 另 则 是 特指 阅读 书籍 电子 时代 更 需要 强调 的 也许 是 后 意思 的 读书 因为 阅读 书籍 读书 比 阅读 电子 屏幕 文字 读屏 更是 专注 的 阅读

原 标题 低价 鱼精蛋白 缺货 心脏病 人 排队 等 药 救心 业内人士 分析 此次 全国性 缺货 或是 药 企 涨价 的 前兆 这个 赵碧珍 觉得 特别 漫长 她 患有 心脏病 需要 开胸 更换 心脏 瓣膜 因 为 手术 必 用药 鱼精蛋白 缺货 她 只能 在 病房 里 排队 等 药 她 的 病友 们 等不及 已 陆续 离 开 她 一直 在 等待 但 不知 何时能 等来 救心 的 药 鱼精蛋白 全国性 缺货 今年 并非 年前 也 曾经 出现 过 低价 救命 药 越来越 高频率 出现 缺货 有 医生 分析 这 与 药品 价格低 企业 利 润 薄 无 生产 积极性 有关 甚至 有 医生 猜测 此次 可能 是 药品 生产 企业 涨价 的 前兆 无奈 的 等待 住院 等 药 救心 华西都市报 记者 能 不能 帮 我 买到 鱼精蛋白 药 近日 赵碧珍 在 走 投无路 的 情况 下 向 本报 求助 赵碧珍 今年 自贡人 去年 体检 中 她 被 查出 患有 风湿性 心 脏病 心脏 瓣膜 出 了 问题 需要 更换 需要 修补 她 入住 成都军区 总医院 心脏外科 等待 手术 医生 告诉 她 因为 缺少 名为 鱼精蛋白 的 药 手术 没法 进行 我 好不容易 排到 了 床位 不 愿 意 轻意 放弃 赵碧珍 说 她 总以为 药 应该 缺 不了 说 不准 过 就 等来 了 药品 于是 她 继续 在 医院 住 了 下去 过 了 药 还是 没 来 她 的 心情 也 越来越 差 据 记者 了解 像 赵碧珍 一样 等 药 做手术 的 病人 不在少数 的 高 含 清 的 心脏 手术 也 是 没药 等待 了 小 手术 被 取消 了 在此期间 其 家人 还 在 四川大学 华西 医院 和 省 人民 医院 去 打听 了 希望 能 借到 药 然而 各 大 医院 都 说 没药 可借 全国性 药荒 今年 来 已成 全国性 缺货 成都军区 总医院 心血管 外科主任 张近宝 说 鱼精蛋白 是 心脏病 人 做 体外循环 手术 时 必需 用 的 药品 根据 体重

LDA

```
初始化停止词列表 --
开始读入语料数据 --
读入语料数据完成，用时9.256秒
文本数目：2043个
正在建立词典 --
正在计算文本向量 --
正在计算文档TF-IDF --
建立文档TF-IDF完成，用时0.185秒
LDA模型拟合推断 --
LDA模型完成，训练时间为 37.687秒
10个文档的主题分布：
第532个文档的前10个主题： [20 18 25  4 13  6 19 28 22 24]
[ 0.50757285  0.10239849  0.07296044  0.05082813  0.02763301  0.02729199
  0.02345142  0.02105525  0.01749429  0.01665937]
第1043个文档的前10个主题： [23 14  4 18 10 24  6 15  0 20]
[ 0.4981378   0.06441008  0.05225744  0.05120348  0.0392068   0.03119684
  0.02928527  0.02618645  0.02403664  0.02328459]
第1035个文档的前10个主题： [19 25  4 11 15  0 16 28  6  7]
[ 0.26742334  0.16533452  0.08484096  0.07141483  0.0688248   0.05389866
  0.05103031  0.03916642  0.03554837  0.027476  ]
第588个文档的前10个主题： [ 7 20  5 12 19 21 15 17 23 14]
[ 0.26408634  0.20762942  0.1160332   0.10415797  0.06068137  0.05660975
  0.02997539  0.01992539  0.01928632  0.01816658]
第1412个文档的前10个主题： [ 6 25  3 22 26 16 19  4 18  7]
[ 0.16465983  0.15589012  0.15210117  0.1234063   0.08512253  0.0831406
  0.04052934  0.03234385  0.0246687   0.02238315]
第805个文档的前10个主题： [ 1 25 19  4 10 15 23 28 26 18]
[ 0.33525038  0.23190863  0.09825045  0.06684136  0.05441141  0.0435245
  0.03313123  0.01985919  0.01859455  0.01445226]
```

主题

6.2.netease_news

```
每个主题的词分布：
主题#0：
词：      村民 乘客 云南 旅客 地上 裤子 妈妈
概率：    [ 0.00682393  0.0042878   0.00323379  0.00318589  0.00316816  0.00306
  0.0029545 ]
主题#1：
词：      广东省 王某 刘某 睾丸 参议院 榆阳区 陈满
概率：    [ 0.00751067  0.00640128  0.00602311  0.00560491  0.00496156  0.00429384
  0.00428849]
主题#2：
词：      工匠 台当局 失误 退役 假如 暴力事件 其一
概率：    [ 0.00298584  0.00257124  0.00240675  0.002152    0.0019788   0.00188708
  0.00166307]
主题#3：
词：      李某 充值 工资 毫米 小杰 平均工资 徐某
概率：    [ 0.01106934  0.00335774  0.00318256  0.00301278  0.00299619  0.00285581
  0.00280268]
主题#4：
词：      阅读 读书 李 女子 视频 书籍 电子
概率：    [ 0.00996042  0.00583026  0.00567708  0.00562837  0.00477045  0.00399124
  0.0039597 ]
主题#5：
词：      普京 伦敦 俄 会谈 安倍 身份证 被捕
概率：    [ 0.00617236  0.00446138  0.00441558  0.0041976   0.00326962  0.00310108
  0.00297686]
主题#6：
词：      企业 政府 患者 公司 医院 建设 医疗
概率：    [ 0.00433506  0.00424583  0.0039865   0.00391137  0.00326307  0.0031044
  0.00304006]
```

路透社数据

159 0:1 2:1 6:1 9:1 12:5 13:2 20:1 21:4 24:2 29:1 35:1 38:2 39:7 48:1 49:1 54:1 59:2 60:1 61:7 66:1
107 0:7 2:2 7:1 16:1 17:1 20:1 24:1 38:3 42:1 59:1 62:1 65:2 70:1 76:2 84:1 87:1 90:2 101:1 107:1 1
153 3:1 4:10 6:4 7:1 8:1 11:9 13:1 20:1 31:3 32:1 33:1 35:2 44:5 45:3 48:5 49:1 62:1 64:1 68:1 71:2
156 0:6 2:1 6:1 7:1 8:1 12:7 18:3 19:1 21:3 22:1 24:3 26:3 27:1 37:1 39:2 40:1 45:1 57:2 60:2 61:3
192 3:2 4:14 5:1 6:1 8:2 9:1 11:11 13:2 14:1 15:3 20:1 26:1 30:1 31:5 33:1 34:1 35:2 37:1 41:1 43:1
180 2:2 3:2 4:24 6:2 8:2 9:1 11:16 13:2 15:2 26:1 31:3 33:3 34:1 35:2 37:3 44:1 48:4 49:1 57:3 64:1
147 3:2 4:7 5:1 6:1 8:1 11:5 13:1 14:1 15:1 31:1 32:1 33:2 34:1 35:2 37:1 41:1 44:4 45:1 48:2 49:2
184 2:2 3:2 4:20 6:2 8:3 9:1 11:15 13:1 15:1 2
163 1:1 3:2 4:17 5:2 6:2 11:14 13:2 14:2 26:1
187 0:2 2:2 5:2 7:1 9:3 12:11 14:1 16:1 18:1 1
170 0:2 3:1 4:1 7:1 12:15 15:2 18:3 19:1 20:1
224 0:1 2:4 4:3 5:1 6:2 7:2 8:2 9:1 10:3 13:1
193 0:1 1:1 2:1 3:1 4:10 6:2 7:3 8:2 11:10 13:
180 0:1 1:1 2:1 3:1 4:12 6:1 7:2 8:2 11:12 13:
237 0:1 2:4 6:2 7:1 8:1 9:2 10:1 12:11 15:1 18
195 0:1 2:1 4:1 12:5 15:1 18:5 19:1 21:3 22:2
194 0:2 2:3 3:1 5:1 6:1 7:3 9:1 12:17 15:4 18:
165 0:1 3:1 5:1 7:3 12:5 15:3 19:1 20:1 21:2 2
134 0:4 5:1 6:2 9:1 15:1 18:1 19:1 23:3 26:1 3
193 0:3 1:1 2:1 3:1 6:3 8:1 9:2 10:1 13:2 14:2
177 0:4 2:2 5:1 6:3 8:1 9:3 13:1 14:1 15:2 16:
180 2:2 3:1 8:1 14:1 17:6 19:1 34:1 36:3 41:1
113 0:1 3:1 5:1 6:1 9:1 15:1 30:1 36:1 37:2 42
93 0:3 4:1 5:1 7:4 9:1 14:1 15:1 19:1 20:1 24:
166 0:2 1:11 3:2 5:2 6:2 7:2 10:1 14:5 15:1 18

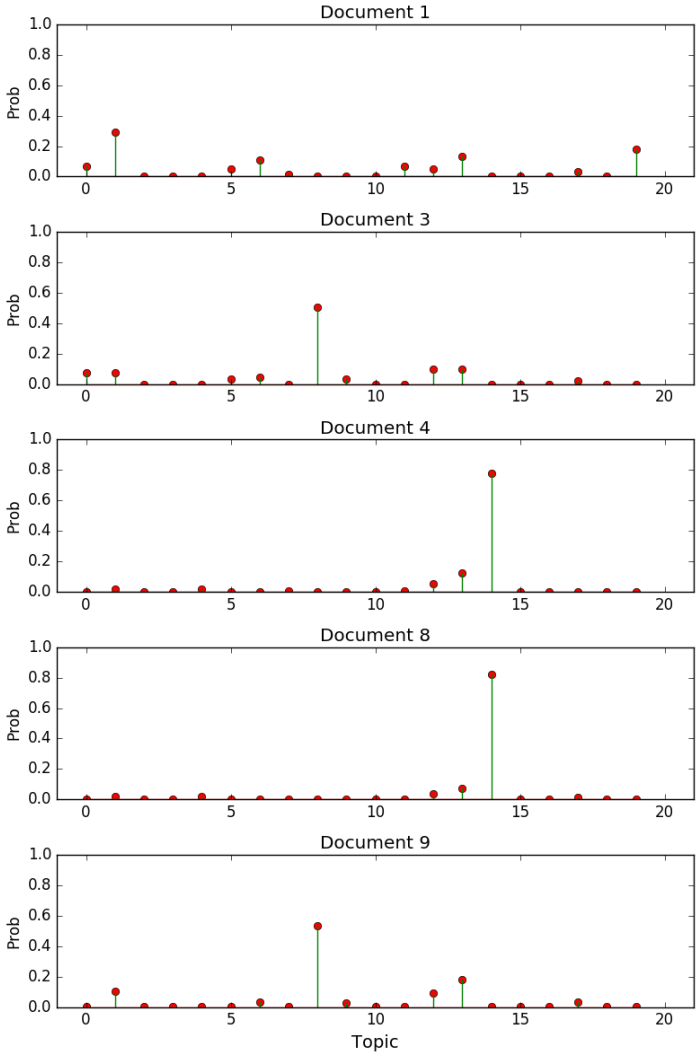
0 UK: Prince Charles spearheads British royal revolution. LONDON 1996-08-20
1 GERMANY: Historic Dresden church rising from WW2 ashes. DRESDEN, Germany 1996-08-21
2 INDIA: Mother Teresa's condition said still unstable. CALCUTTA 1996-08-23
3 UK: Palace warns British weekly over Charles pictures. LONDON 1996-08-25
4 INDIA: Mother Teresa, slightly stronger, blesses nuns. CALCUTTA 1996-08-25
5 INDIA: Mother Teresa's condition unchanged, thousands pray. CALCUTTA 1996-08-25
6 INDIA: Mother Teresa shows signs of strength, blesses nuns. CALCUTTA 1996-08-26
7 INDIA: Mother Teresa's condition improves, many pray. CALCUTTA, India 1996-08-25
8 INDIA: Mother Teresa improves, nuns pray for "miracle". CALCUTTA 1996-08-26
9 UK: Charles under fire over prospect of Queen Camilla. LONDON 1996-08-26
10 UK: Britain tells Charles to forget Camilla. LONDON 1996-08-27
11 COTE D'IVOIRE: FEATURE - Quiet homecoming for reprieved Ivory Coast maid. ABIDJAN 1996-08-28
12 INDIA: Mother Teresa ("I want to go home") sits and prays. CALCUTTA 1996-08-28
13 INDIA: Mother Teresa nears end of crisis, nuns rejoice. CALCUTTA 1996-08-28
14 UK: Prosaic end for marriage of Charles and Diana. LONDON 1996-08-28
15 UK: No respite for British royals despite divorce. LONDON 1996-08-28
16 UK: Camilla, love of Charles' life, an unlikely queen. LONDON 1996-08-28
17 UK: Diana sets out on new life as single woman. LONDON 1996-08-28
18 USA: O.J. Simpson attacks media, hints at lawsuits. WASHINGTON 1996-08-28
19 USA: U.S. Cardinal Bernardin has one year or less to live. CHICAGO 1996-08-30
20 USA: U.S. Cardinal Bernardin says has terminal cancer. CHICAGO 1996-08-30
21 ROMANIA: German architect wins Bucharest rebuilding prize. BUCHAREST 1996-09-02
22 ARGENTINA: Argentina's "Blond Angel" finally quits Navy. BUENOS AIRES, Argentina 1996-09-02
23 UK: Disney lights up Pocahontas resting place. GRAVESEND, England 1996-09-06
24 HUNGARY: POPE LEAVES HUNGARY AFTER DEMANDING TWO-DAY VISIT. BUDAPEST 1996-09-07
25 HUNGARY: Pope says mass in Hungary, health in spotlight. GYOR, Hungary 1996-09-07
26 UK: Prince Charles' love will not wed him, paper says. LONDON 1996-09-09

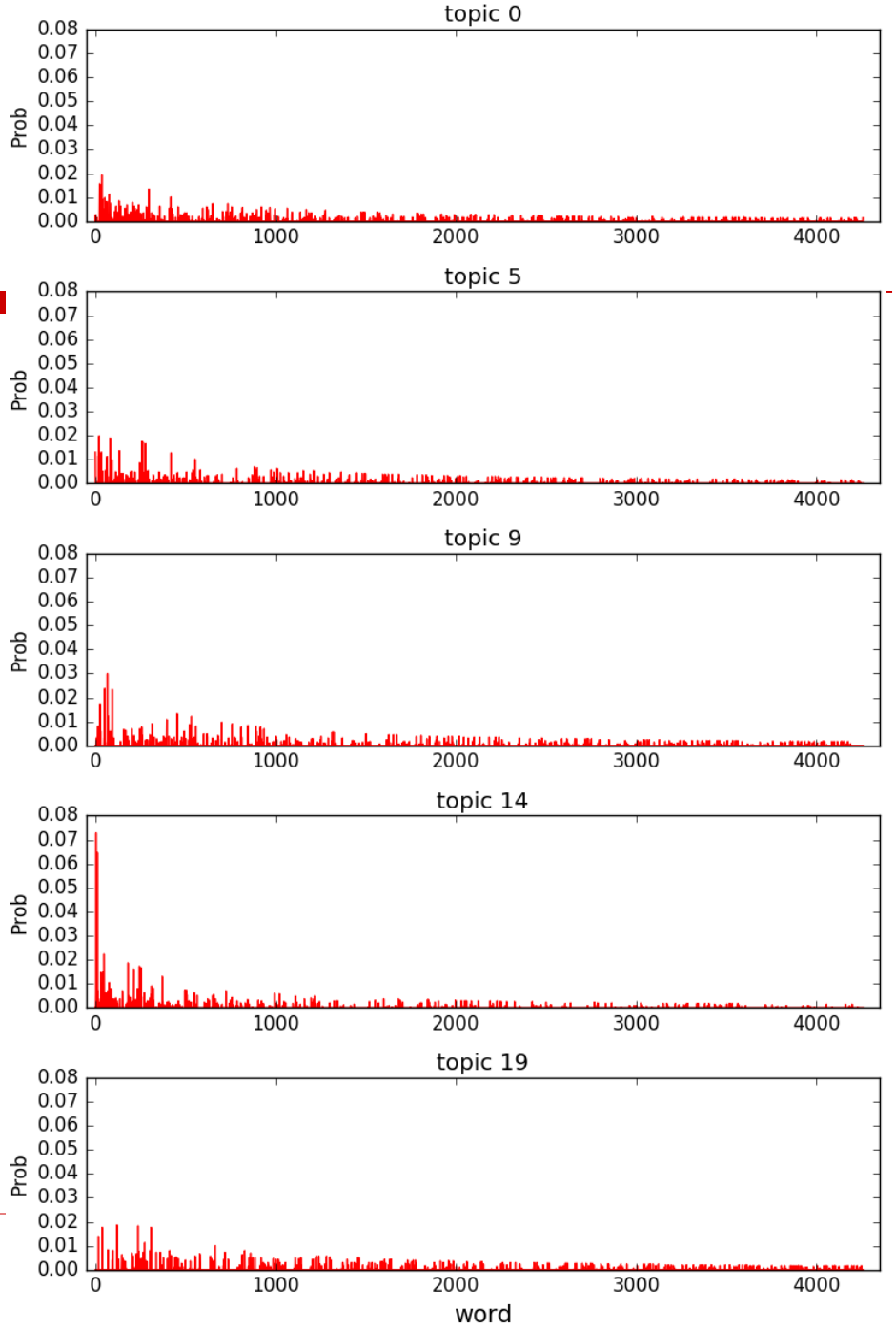
church
pope
years
people
mother
last
told
first
world
year
president
teresa
charles
catholic
during
life
u.s
city
public
time
since
family
king
former
british
harriman
against
country
vatican
made
three
hospital

LDA

```
C:\Python27\python.exe D:/Python/16.3.reuters.py
type(X): <type 'numpy.ndarray'>
shape: (395, 4258)

[[ 1  0  1  0  0  0  1  0  0  1]
 [ 7  0  2  0  0  0  0  1  0  0]
 [ 0  0  0  1 10  0  4  1  1  0]
 [ 6  0  1  0  0  0  1  1  1  0]
 [ 0  0  0  2 14  1  1  0  2  1]
 [ 0  0  2  2 24  0  2  0  2  1]
 [ 0  0  0  2  7  1  1  0  1  0]
 [ 0  0  2  2 20  0  2  0  3  1]
 [ 0  1  0  2 17  2  2  0  0  0]
 [ 2  0  2  0  0  2  0  1  0  3]]
type(vocab): <type 'tuple'>
len(vocab): 4258

('church', 'pope', 'years', 'people', 'mother', 'last', 'told', 'first', 'world', 'year')
type(titles): <type 'tuple'>
len(titles): 395

('0 UK: Prince Charles spearheads British royal revolution. LONDON 1996-08-20', '1 GERMANY:
LDA start ----
INFO:lda:n_documents: 395
INFO:lda:vocab_size: 4258
INFO:lda:n_words: 84010
INFO:lda:n_topics: 20
INFO:lda:n_iter: 500
INFO:lda:<0> log likelihood: -1051748
INFO:lda:<10> log likelihood: -719800
INFO:lda:<20> log likelihood: -699115
INFO:lda:<30> log likelihood: -689370
```

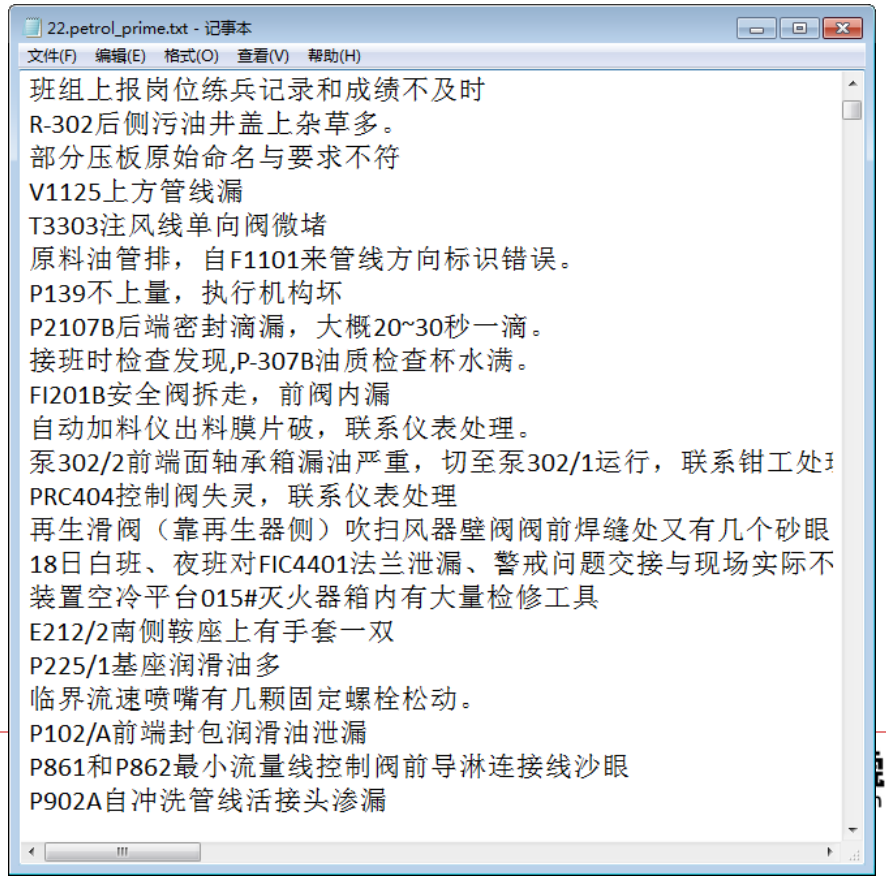
主题和主题分布


Document 1
Document 3
Document 4
Document 8
Document 9
Prob
Topic


topic 0
topic 5
topic 9
topic 14
topic 19
Prob
word

石油例检结果处理

- 针对国内某石油企业的例行检查处理结果，试通过主题模型方案，分析例检结果中最突出的问题是什么？
  - 文本共4700个，
  - 单个文档十数字



聚类“主主题”

22.4.petrol 22.4.petrol

每个主题的词分布：

主题#0： 溶脱 地面 下 发现 溜 吹 漏洞

概率： [0.03567895 0.0305515 0.02995125 0.02905559 0.02847266 0.02672661 0.026

主题#1： 卫生 清扫 新增 本月 缺陷 无 空调

概率： [0.09127588 0.04710893 0.03864091 0.0385492 0.03507678 0.03243568 0.021

主题#2： 过滤器 设备 高 差压 少 入口 17

概率： [0.07382152 0.04735673 0.03770307 0.03342033 0.03160451 0.02609018 0.023

主题#3： 号牌 位 脱落 铁皮 北侧 塔 规范

概率： [0.06217475 0.06146815 0.04165677 0.03554779 0.03128229 0.02976478 0.024

主题#4： 松 建议 液位 损坏 皮带 区域 断裂

概率： [0.03845036 0.03260667 0.03243315 0.031336 0.03074858 0.03062344 0.028

主题#5： 盘根 漏 阀 出口 采样 引出 内

概率： [0.09527604 0.08630304 0.07457675 0.0508139 0.04410229 0.03406847 0.033

主题#6： 日 月 8 红线 压力表 技术员 23

概率： [0.04320296 0.04268257 0.04081915 0.03381051 0.02013684 0.01981086 0.016

主题#7： 地沟 错误 单向阀 螺丝 装车 旁 年

概率： [0.03169998 0.02547323 0.02350422 0.02196816 0.02146054 0.02122409 0.019

主题#8： 皮带 断 不准 松动 一次 盖 被

概率： [0.15231247 0.10833394 0.05339595 0.04884857 0.04585281 0.03503216 0.034

主题#9： 电机 杂音 接头 大盖 冲洗 有 活

概率： [0.07620952 0.0712979 0.06082735 0.03423004 0.03310958 0.02874415 0.027

主题#10： 蒸汽 砂眼 管线 法兰 漏 前 有

概率： [0.04160303 0.03960238 0.03123862 0.0293527 0.02685108 0.02634925 0.024

主题#11： 泄漏 伴热 量 牌 入口 底 法兰

概率： [0.07545736 0.05467109 0.05165556 0.03689506 0.03672955 0.03457064 0.031

主题#12： 密封 平台 保温 脱开 缺失 堵头 汽提

概率： [0.04631792 0.04157294 0.0396999 0.03857663 0.03751675 0.03526295 0.034

主题#13： 后端 长明灯 无法 号 灭火器 器 油站

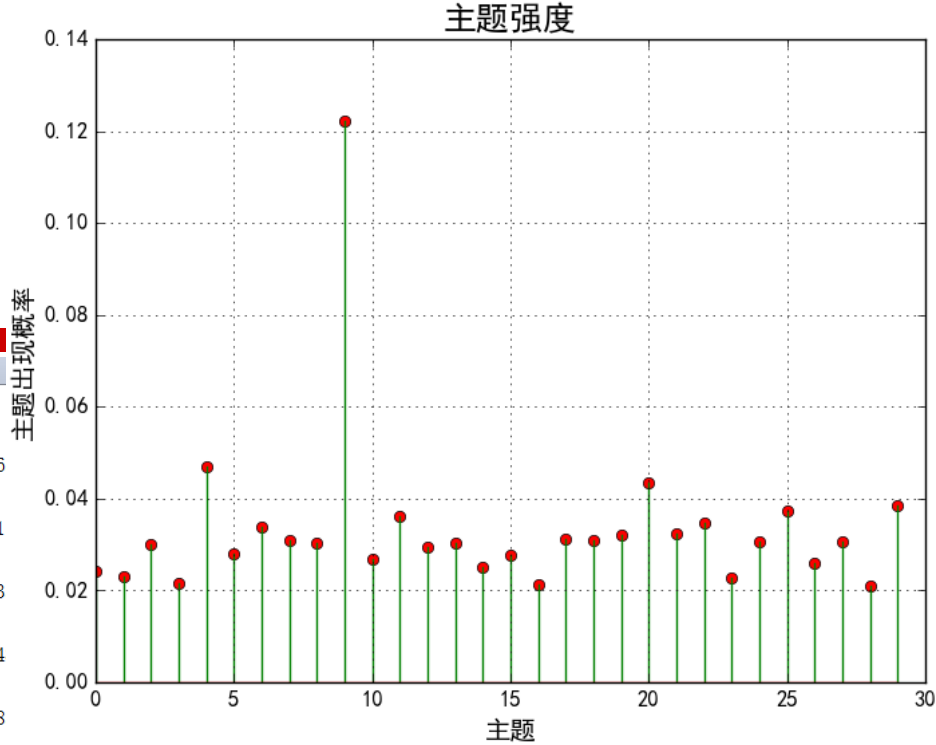
概率： [0.04065264 0.03001566 0.02572 0.02520191 0.02510101 0.02370398 0.022

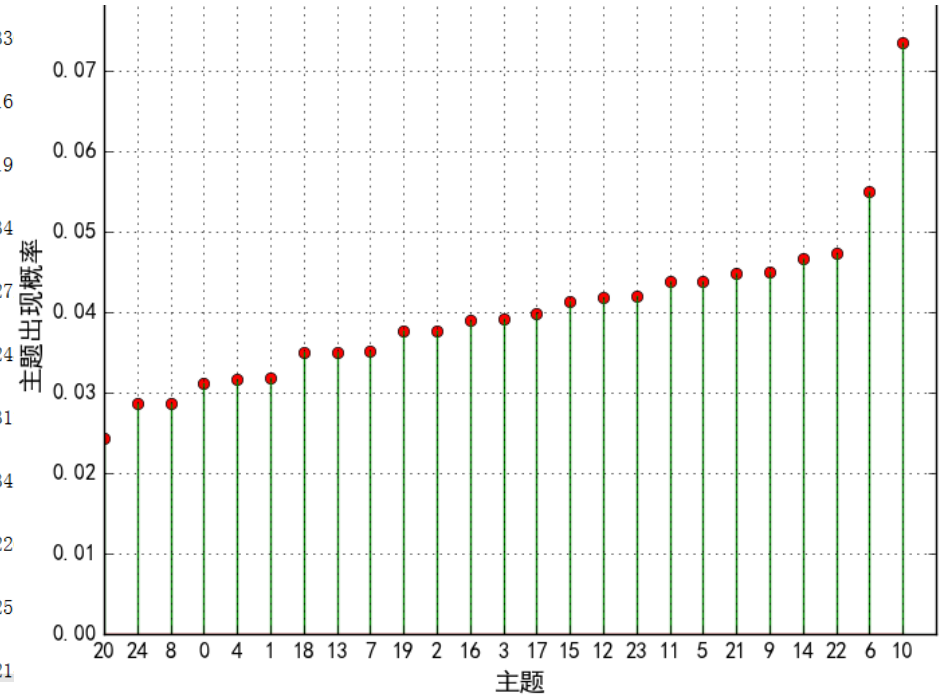
主题#14： 坏 炉 压力表 润滑油 泵 安全阀 清理

概率： [0.04789169 0.03913219 0.03672254 0.03227133 0.03054908 0.02618981 0.025

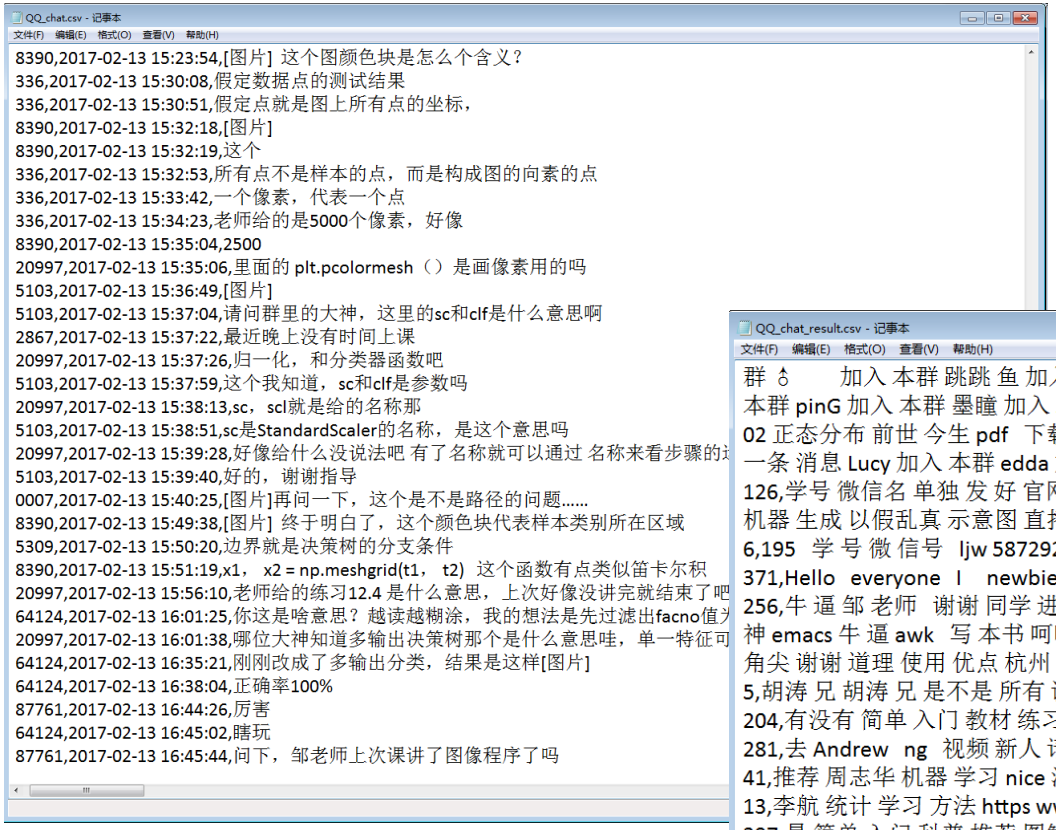
主题#15： 缺陷 新增 无 本月 交接班 胶带机 日志

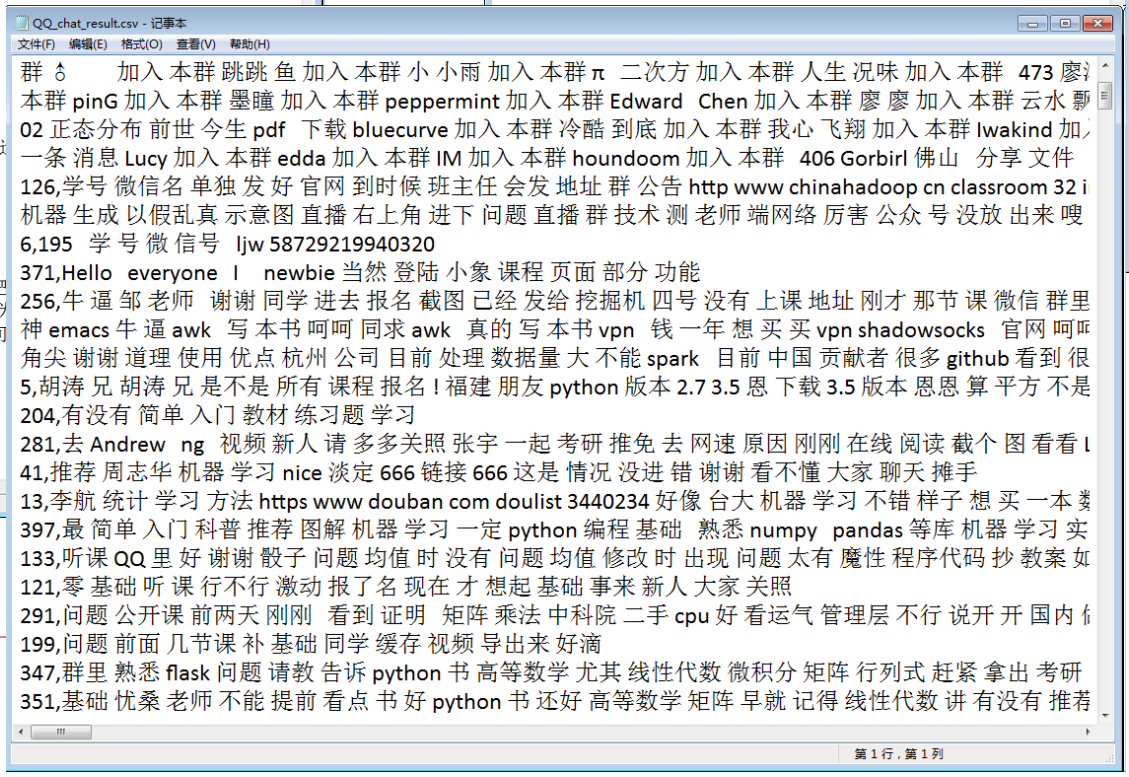
概率： [0.07364203 0.07360244 0.06751279 0.06417833 0.03489421 0.02254108 0.021





聊天记录分析感兴趣话题

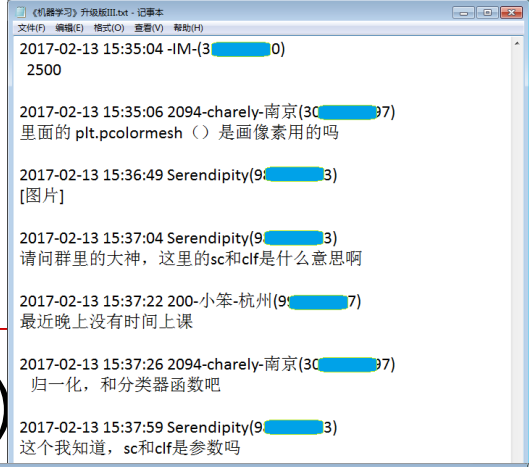


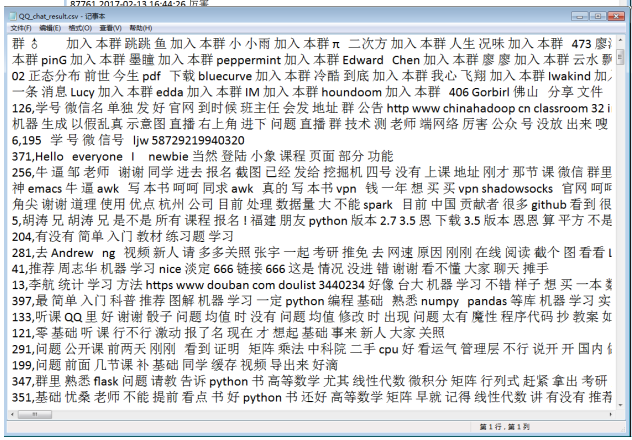


互联网新技术在线教育领航者

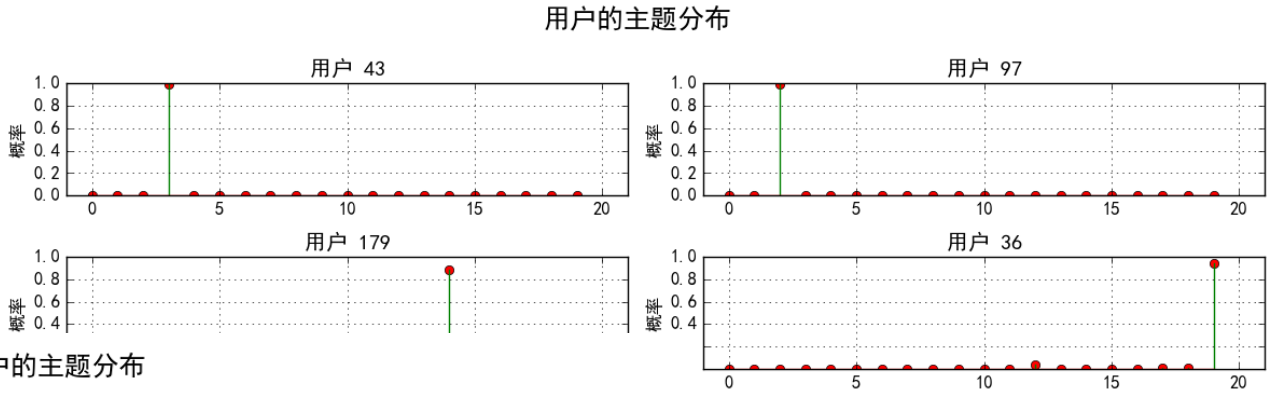
数据处理流程

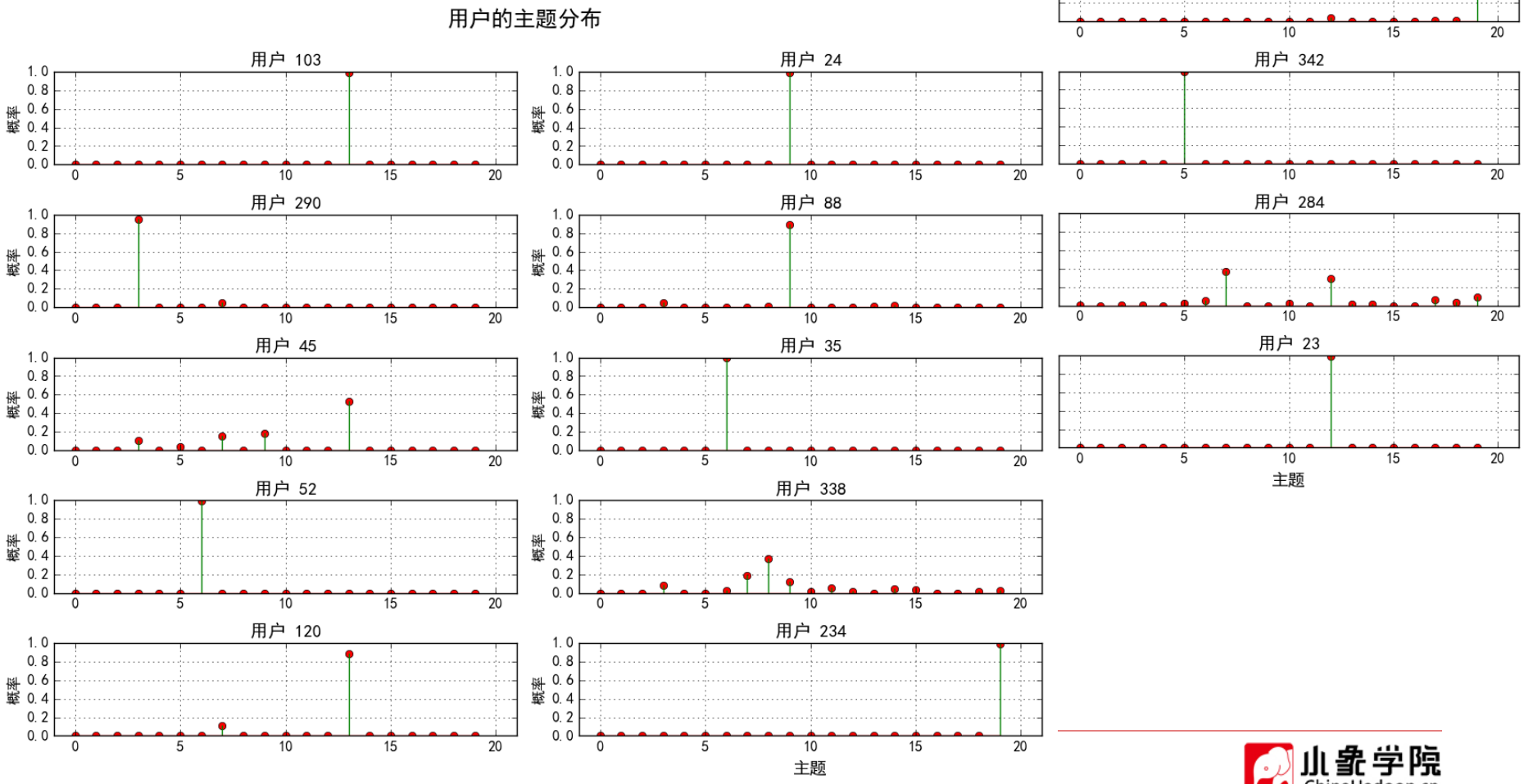
- 获取QQ群聊天记录：txt文本格式(图1)
- 整理成“QQ号/时间/留言”的规则形
  - 正则表达式
  - 清洗特定词：表情、@XX
  - 使用停止词库
  - 获得csv表格数据(图2)
- 合并相同QQ号的留言
  - 长文档利于计算每人感兴趣话题(图3)
- LDA模型计算主题
  - 调参与可视化
- 计算每个QQ号及众人感兴趣话题





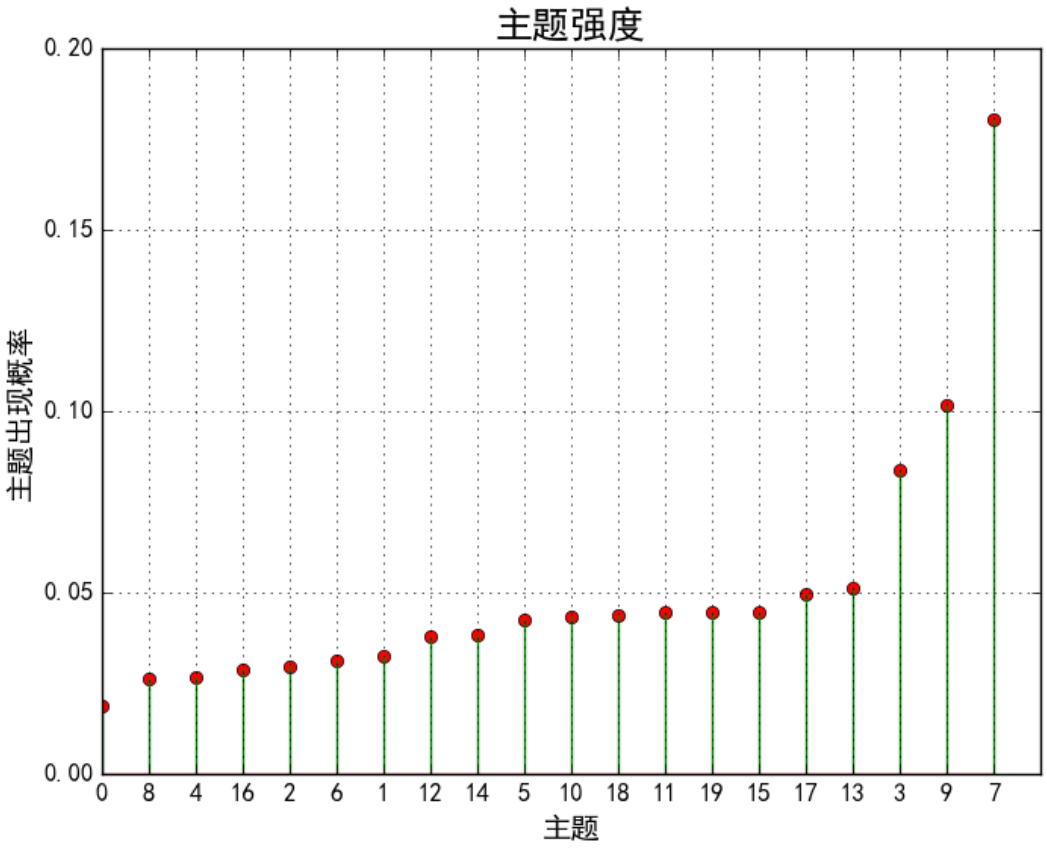
主题分布





感兴趣话题

主题#1: 下周 用到 绑定 matlab 手册 详细描述 666
概率: [0.00386318 0.00379716 0.00275161 0.00267257 0.0025779 0.002
主题#2: 决策树 下载 代码 新入 记得 无法 if
概率: [0.00459593 0.00368178 0.00332368 0.00322059 0.00282207 0.002
主题#3: 下载 视频 app 中 cn 电脑
概率: [0.01650318 0.00643555 0.00534376 0.00493769 0.00477131 0.004
主题#4: 互扫 上装 停留 一份 层次 缺 找点
概率: [0.00457096 0.00322019 0.00314435 0.00283468 0.0027911 0.002
主题#5: 同问 变量 极限 训练 毕竟 问
概率: [0.00799138 0.00796053 0.00353821 0.00350646 0.00323556 0.003
主题#6: 上网 元素 等待 白屏 中国 在线看
概率: [0.00678803 0.00387909 0.00340132 0.00335809 0.00303858 0.002
主题#7: 好 大家 老师 学习 没有 谢谢
概率: [0.02435813 0.01137977 0.01116381 0.01048067 0.01000842 0.008
主题#8: 参考书 需不需要 离散 型 基础知识 组成
概率: [0.00909812 0.00337894 0.00242403 0.00229328 0.00220396 0.00215262 0.00209999]
主题#9: com 91pintuan wxe9c061d2aed9ae09 group https 请
概率: [0.01010167 0.00642439 0.00517232 0.00517232 0.00495921 0.00485321 0.00467808]
主题#10: 画面 端 找回 QQ 群 账号
概率: [0.00641685 0.00384681 0.00359311 0.00355134 0.00343251 0.00288641 0.00272948]
主题#11: 直播 今天 一片空白 没有 进不去 问题
概率: [0.01035459 0.00352562 0.00291557 0.00276692 0.00275323 0.00259631 0.00257422]
主题#12: 太过分 编程 讲完 闪照 配置 这部分
概率: [0.00510845 0.00290508 0.00289446 0.00273205 0.00265899 0.00251825 0.00249154]
主题#13: pydotplus 学号 积分 浏览器 弄 请问
概率: [0.01348479 0.00478899 0.00342937 0.00260794 0.00236339 0.00226667 0.00213521]
主题#14: 分享 两个 满足 话 京东 需求
概率: [0.0143761 0.00353596 0.00270139 0.00264514 0.00255197 0.00249422 0.00244258]
主题#15: 相等 听不见 openCV 版 数据库 设置
概率: [0.01212347 0.00449718 0.0030489 0.0030018 0.00285451 0.00281434 0.00263035]
主题#16: 回放 课程 OpenStack 牛云 投放 点哇
概率: [0.01010629 0.00425028 0.0026812 0.00238546 0.00221883 0.00221883 0.00221883]
主题#17: 福 敬业 早 断 厉害 歌



正则表达式

语法	说明	表达式实例	完整匹配的字符串
字符			
一般字符	匹配自身	abc	abc
.	匹配任意除换行符"\n"外的字符。 在DOTALL模式中也能匹配换行符。	a.c	abc
\	转义字符，使后一个字符改变原来的意思。 如果字符串中有字符*需要匹配，可以使用*或者字符集[*]。	a\.c a\\c	a.c a\c
[...]	字符集（字符类）。对应的位置可以是字符集中任意字符。 字符集中的字符可以逐个列出，也可以给出范围，如[abc]或[a-c]。第一个字符如果是^则表示取反，如[^abc]表示不是abc的其他字符。 所有的特殊字符在字符集中都失去其原有的特殊含义。在字符集中如果要使用]、-或^，可以在前面加上反斜杠，或把]、-放在第一个字符，把^放在非第一个字符。	a[bcd]e	abe ace ade
预定义字符集（可以写在字符集[...]中）			
\d	数字：[0-9]	a\dc	a1c
\D	非数字：[^\d]	a\Dc	abc
\s	空白字符：[<空格>\t\r\n\f\v]	a\sc	a c
\S	非空白字符：[^\s]	a\Sc	abc
\w	单词字符：[A-Za-z0-9_]	a\wc	abc
\W	非单词字符：[^\w]	a\Wc	a c
数量词（用在字符或(...)之后）			
*	匹配前一个字符0或无限次。	abc*	ab abccc
+	匹配前一个字符1次或无限次。	abc+	abc abccc
?	匹配前一个字符0次或1次。	abc?	ab abc
{m}	匹配前一个字符m次。	ab{2}c	abbc
{m,n}	匹配前一个字符m至n次。 m和n可以省略：若省略m，则匹配0至n次；若省略n，则匹配m至无限次。	ab{1,2}c	abc abbc
*? +? ?? {m,n}?	使 * + ? {m,n}变成非贪婪模式。	示例将在下文中介绍。	

边界匹配（不消耗待匹配字符串中的字符）			
^	匹配字符串开头。 在多行模式中匹配每一行的开头。	^abc	abc
$	匹配字符串末尾。 在多行模式中匹配每一行的末尾。	abc$	abc
\A	仅匹配字符串开头。	\Aabc	abc
\Z	仅匹配字符串末尾。	abc\Z	abc
\b	匹配\w和\W之间。	a\b!bc	a!bc
\B	[^\b]	a\Bbc	abc
逻辑、分组			
|	|代表左右表达式任意匹配一个。 它总是先尝试匹配左边的表达式，一旦成功匹配则跳过匹配右边的表达式。 如果|没有被包括在()中，则它的范围是整个正则表达式。	abc|def	abc def
(...)	被括起来的表达式将作为分组，从表达式左边开始每遇到一个分组的左括号'('，编号+1。 另外，分组表达式作为一个整体，可以后接数量词。表达式中的|仅在该组中有效。	(abc){2} a(123|456)c	abcabc a456c
(?P<name>...)	分组，除了原有的编号外再指定一个额外的别名。	(?P<id>abc){2}	abcabc
\<number>	引用编号为<number>的分组匹配到的字符串。	(\d)abc\1	1abc1 5abc5
(?P=name)	引用别名为<name>的分组匹配到的字符串。	(?P<id>\d)abc(?P=id)	1abc1 5abc5
特殊构造（不作为分组）			
(?:...)	(...)的不分组版本，用于使用'|'或后接数量词。	(?:abc){2}	abcabc
(?iLmsux)	iLmsux的每个字符代表一个匹配模式，只能用在正则表达式的开头，可选多个。匹配模式将在下文中介绍。	(?i)abc	AbC
(?#...)	#后的内容将作为注释被忽略。	abc(?#comment)123	abc123
(?=...)	之后的字符串内容需要匹配表达式才能成功匹配。 不消耗字符串内容。	a(?=\d)	后面是数字的a
(?!...)	之后的字符串内容需要不匹配表达式才能成功匹配。 不消耗字符串内容。	a(?!\d)	后面不是数字的a
(?<=...)	之前的字符串内容需要匹配表达式才能成功匹配。 不消耗字符串内容。	(?<=\d)a	前面是数字的a
(?<!...)	之前的字符串内容需要不匹配表达式才能成功匹配。 不消耗字符串内容。	(?<!\d)a	前面不是数字的a
(?(id/name) yes-pattern |no-pattern)	如果编号为id/别名为name的组匹配到字符，则需要匹配yes-pattern，否则需要匹配no-pattern。 |no-patern可以省略。	(\d)abc(?(1)\d|abc)	1abc2 abcabc

常用正则表达式

- 匹配中文字符：[\u4e00-\u9fa5]
- 匹配双字节字符(包括汉字在内)：[^\x00-\xff]
- 匹配空白行：\n\s*\r
- 匹配HTML标记：<(\S*?)[^>]*>.*?</\1>|<.*? />
- 匹配首尾空白字符：^\s*|\s*$
- 匹配Email地址：\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*
- 匹配网址URL：[a-zA-z]+://[^\s]*
- 匹配帐号合法(5-16位，字母开头，允许字母数字下划线)：^[a-zA-Z][a-zA-Z0-9_]{4,15}$
- 匹配国内电话号码：\d{3}-\d{8}|\d{4}-\d{7}
- 匹配腾讯QQ号：[1-9][0-9]{4,}
- 匹配中国邮政编码：[1-9]\d{5}(?!\d)
- 匹配身份证：\d{15}|\d{18}|\d{17}[xX]
- 匹配ip地址：\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}

常用正则表达式

- 匹配特定数字：
  - 匹配正整数：^[1-9]\d*$
  - 匹配负整数：^-[1-9]\d*$
  - 匹配整数：^-?[1-9]\d*$
  - 匹配非负整数(正整数 + 0)：^[1-9]\d*|0$
  - 匹配非正整数(负整数 + 0)：^-[1-9]\d*|0$
  - 匹配正浮点数：^[1-9]\d*\.\d*|0\.\d*[1-9]\d*$
  - 匹配负浮点数：^-([1-9]\d*\.\d*|0\.\d*[1-9]\d*)$
  - 匹配浮点数：^-?([1-9]\d*\.\d*|0\.\d*[1-9]\d*|0?\.0+|0)$
  - 匹配非负浮点数(正浮点数 + 0)：^[1-9]\d*\.\d*|0\.\d*[1-9]\d*|0?\.0+|0$
  - 匹配非正浮点数(负浮点数 + 0)：^(-([1-9]\d*\.\d*|0\.\d*[1-9]\d*))|0?\.0+|0$
- 匹配特定字符串：
  - 匹配由26个英文字母组成的字符串：^[A-Za-z]+$
  - 匹配由26个英文字母的大写组成的字符串：^[A-Z]+$
  - 匹配由26个英文字母的小写组成的字符串：^[a-z]+$
  - 匹配由数字和26个英文字母组成的字符串：^[A-Za-z0-9]+$
  - 匹配由数字26个英文字母或下划线组成的字符串：^\w+$

Code

```
def clean_info(info):
    replace_str = (('\n', ''), ('\r', ''), (',', ', '), ('表情', ''))
    for rs in replace_str:
        info = info.replace(rs[0], rs[1])

    at_pattern = re.compile(r'(@.* )')
    at = re.findall(pattern=at_pattern, string=info)
    for a in at:
        info = info.replace(a, '')
    idx = info.find('@')
    if idx != -1:
        info = info[:idx]
    return info


def regularize_data():
    time_pattern = re.compile(r'\d{4}-\d{2}-\d{2} \d{1,2}:\d{1,2}:\d{1,2}')
    qq_pattern1 = re.compile(r'([1-9][0-9]{4,})')    # QQ号最小是10000
    qq_pattern2 = re.compile(r'(\w+([-+.]\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*)')
    f = open(u'《机器学习》升级版III.txt')
    f_output = open(u'QQ_chat.csv', mode='w')
    f_output.write('QQ,Time,Info\n')
    qq = chat_time = info = ''
    for line in f:
        line = line.strip()
        if line:
            t = re.findall(pattern=time_pattern, string=line)
            qq1 = re.findall(pattern=qq_pattern1, string=line)
            qq2 = re.findall(pattern=qq_pattern2, string=line)
            if (len(t) >= 1) and ((len(qq1) >= 1) or (len(qq2) >= 1)):
                if info:
                    info = clean_info(info)
                    if info:
                        info = '%s,%s,%s\n' % (qq, chat_time, info)
                        f_output.write(info)
                        info = ''
                if len(qq1) >= 1:
                    qq = qq1[0]
                else:
                    qq = qq2[0][0]
                chat_time = t[0]
            else:
                info += line
    f.close()
    f_output.close()
```

```
def segment():
    stopwords = load_stopwords()
    data = pd.read_csv('QQ_chat.csv', header=0)
    for i, info in enumerate(data['Info']):
        info_words = []
        words = jieba.cut(info)
        for word in words:
            if word not in stopwords:
                info_words.append(word.encode('utf-8'))
        if info_words:
            data.iloc[i, 2] = ' '.join(info_words)
        else:
            data.iloc[i, 2] = np.nan
    data.dropna(axis=0, how='any', inplace=True)
    data.to_csv('QQ_chat_segment.csv', sep=',', header=True, index=False)


def combine():
    data = pd.read_csv('QQ_chat_segment.csv', header=0)
    data['QQ'] = pd.Categorical(data['QQ']).codes
    f_output = open('QQ_chat_result.csv', mode='w')
    f_output.write('QQ,Info\n')
    for qq in data['QQ'].unique():
        info = ' '.join(data[data['QQ'] == qq]['Info'])
        str = '%s,%s\n' % (qq, info)
        f_output.write(str)
    f_output.close()
```

参考文献

- David M. Blei, Andrew Y. Ng, Michael I. Jordan, *Latent Dirichlet Allocation*, 2003
- Gregor Heinrich, *Parameter estimation for text analysis.* 2008
- Matthew D. Hoffman, David M. Blei, Francis Bach. *Online learning for Latent Dirichlet Allocation.* 2010
- http://en.wikipedia.org/wiki/Dirichlet_distribution
- http://en.wikipedia.org/wiki/Conjugate_prior

我们在这里

- http://wenda.ChinaHadoop.cn
  - 视频/课程/社区
- 微博
  - @ChinaHadoop
  - @邹博_机器学习
- 微信公众号
  - 小象
  - 大数据分析挖掘

感谢大家！

恳请大家批评指正！