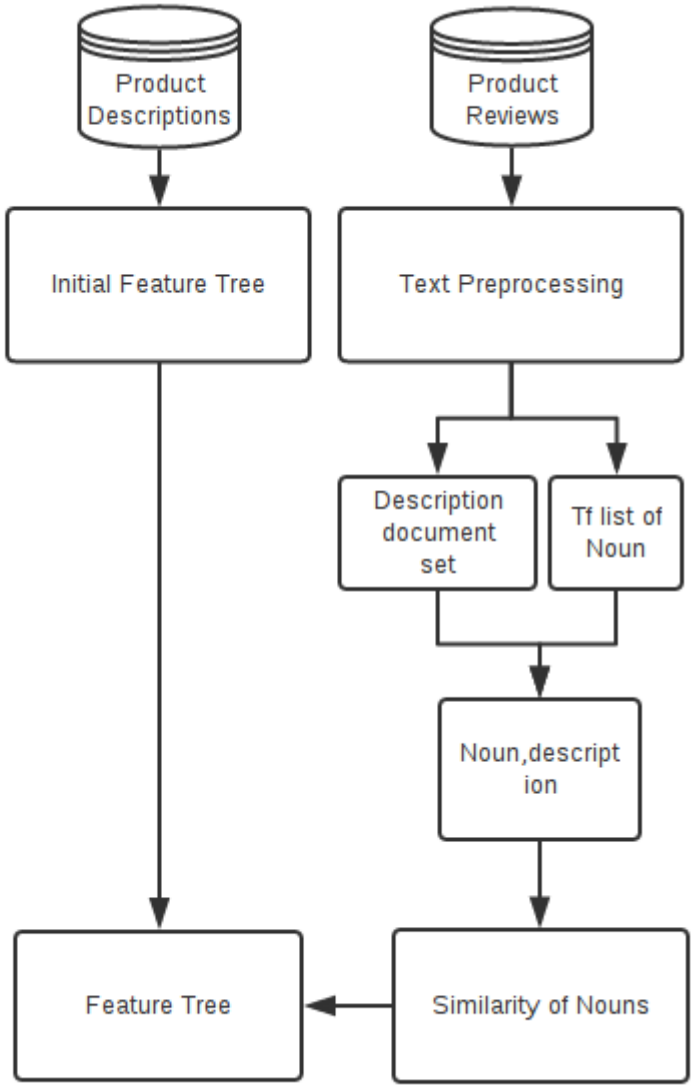
数据来源与方法框架

一，叙论：

近年来，电子商务行业发展迅速，业类各大公司都积累了大量的数据，其中包括顾客的商品评论。同时，有大量的研究指向商品评论。关于商品评论研究的文献回顾。有关于语义挖掘的文献综述。

在这里我们给出来一种方法，对评论文本内容进行概念建模。生成了顾客在给出评价时所依照的概念树。

研究框架如下：



二，数据来源：

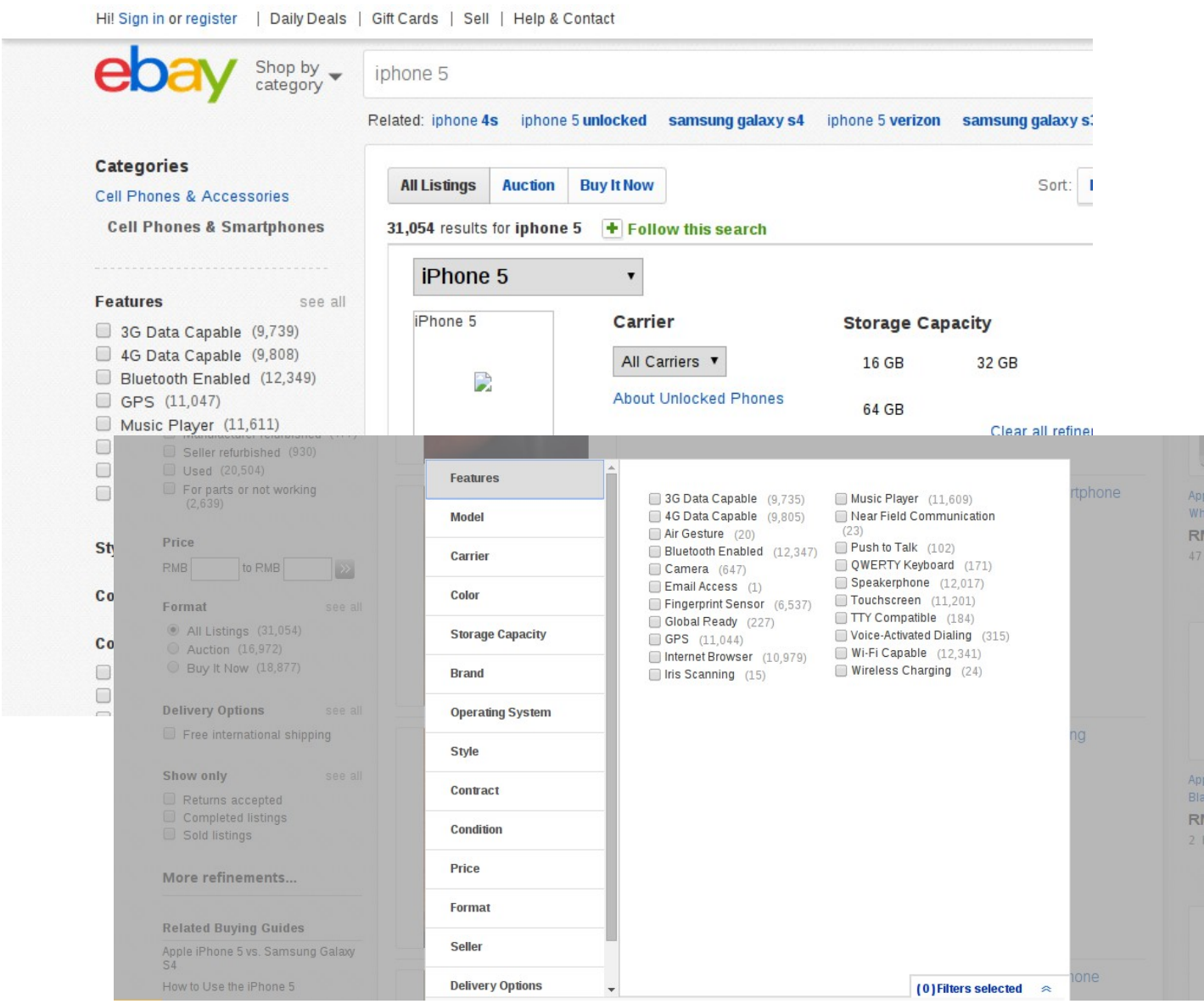
在线商品与商品描述（eg，与手机有关）。

１，京东商城商品，描述，评论。

2，亚马逊网站的商品，商品描述

3，淘宝网的商品，描述。

4，ebay 网商品，描述



三，生成概念树

以京东商城 IPHONE5S 商品为例。包括其商品描述和商品评论。

1,基于商品描述构造初始概念树。

根节点：京东

第一层：手机

第二层：品牌－价格－网络－系统－机身，颜色－显示－主体－存储－感应器－摄像－娱乐－传输－其它

第三层：品牌：华为－苹果－三星－酷派－努比亚－htc-诺基亚－魅族－联想－索尼－中兴。。。

价格：

网络：移动－电信－联通－2G-3G-4G-双卡-网络-制式-数据-业务-频率

系统：安卓-苹果-微软-IOS-windowsphone

机身，颜色：白色-灰色－黑色－金色－银色－红色－蓝色－粉色－黄色－绿色－橙色－紫色

显示：屏幕-材质-尺寸-触摸屏-分辨率-字体-色彩

主体：型号-输入-智能机-视频通话-版本-CPU-频率-核数

存储：内存-运行-存储卡-类型-16G-ROM

感应器：gps-重力-光线-距离

摄像：摄像头-闪光灯-对焦

娱乐：收音机-音乐-视频-电视

传输：WIFI-蓝牙-热点

其它：SIM 卡-尺寸-电池-型号-类型-通话-时间-待机-数据线-耳机-接口-尺寸-重量

讨论点（1，每一个结点的名词也是该结点的特征。而结点“其它”并不是一个有效的名词特征。在评论文本中，“其它”作为一个词很少出现，而位于“其它”下的特征名词则能作为该类的特征，因而，对于结点“其它”并不会对概念树的生成造成恶劣的影响。2,每个结点可能是一个词，也可能是 N 个词。）

在实验中，我们以根节点到第二层树的生成为例。如果构造的初始概念树有 N 层，我们的方法也可以用于生成 N 层的概念树。

2,计算数据集中名词与名词之间的相似性。

按句子隔断后一般结果构成（目标名词+其他名词＋动词＋形容词+其他词）。

句子的主题是名词，其他词性作为名词的描述词出现。将每一个名词的描述词集合作为一个文档，训练 TopicModel。根据每个文档的 topic 分布，利用 KL 距离，度量名词之间的距离。

语义上的假设：同义词往往对应相似的描述词，描述词的差异越大，其语义差距越大，在语义 topic 上的分布，差异越大。

- 文本预处理与名词词频统计

  分词（词性标记）-去停用词

  预处理并按照句号，问号，感叹号（通常认为的语义切换符号）。对文档进行分段处理。

词频统计	名词词频统计	其他词性词频统计
不错/a 8787 手机/n 4643 买/v 3890 京东/ns 3319	手机/n 4643 京东/ns 3319 苹果/n 3065 正品/n 2658	不错/a 8787 买/v 3890 不/d 2512 喜欢/v 2055

苹果/n 3065 正品/n 2658 不/d 2512 喜欢/v 2055 东西/ns 1786 速度/n 1452 好好/d 1442 说/v 1318 5s/n 1254 感觉/n 1252 送货/v 1077 满意/v 1058 很快/d 1046 质量/n 986 给力/a 965 系统/n 957	东西/ns 1786 速度/n 1452 5s/n 1254 感觉/n 1252 质量/n 986 系统/n 957 价格/n 803 土豪/n 746 朋友/n 724 4g/n 698 行货/n 680 物流/n 664 信号/n 565 电池/n 537 评价/n 533 屏幕/n 510	好好/d 1442 说/v 1318 送货/v 1077 满意/v 1058 很快/d 1046 给力/a 965 送/v 869 很好/a 843 值得/v 825 很好/d 714 支持/v 701 太/d 693 快递/v 693 发现/v 627 流畅/a 621 赞/v 572

- 生成 名词-名词描述文档。
  1,文档中不存在名词时，对该句不理睬。
  2,句子中不只一个名词。可能的假设：其他名词是某一个名词的附属词，按词频排序，只计入高频名词的描述文档。

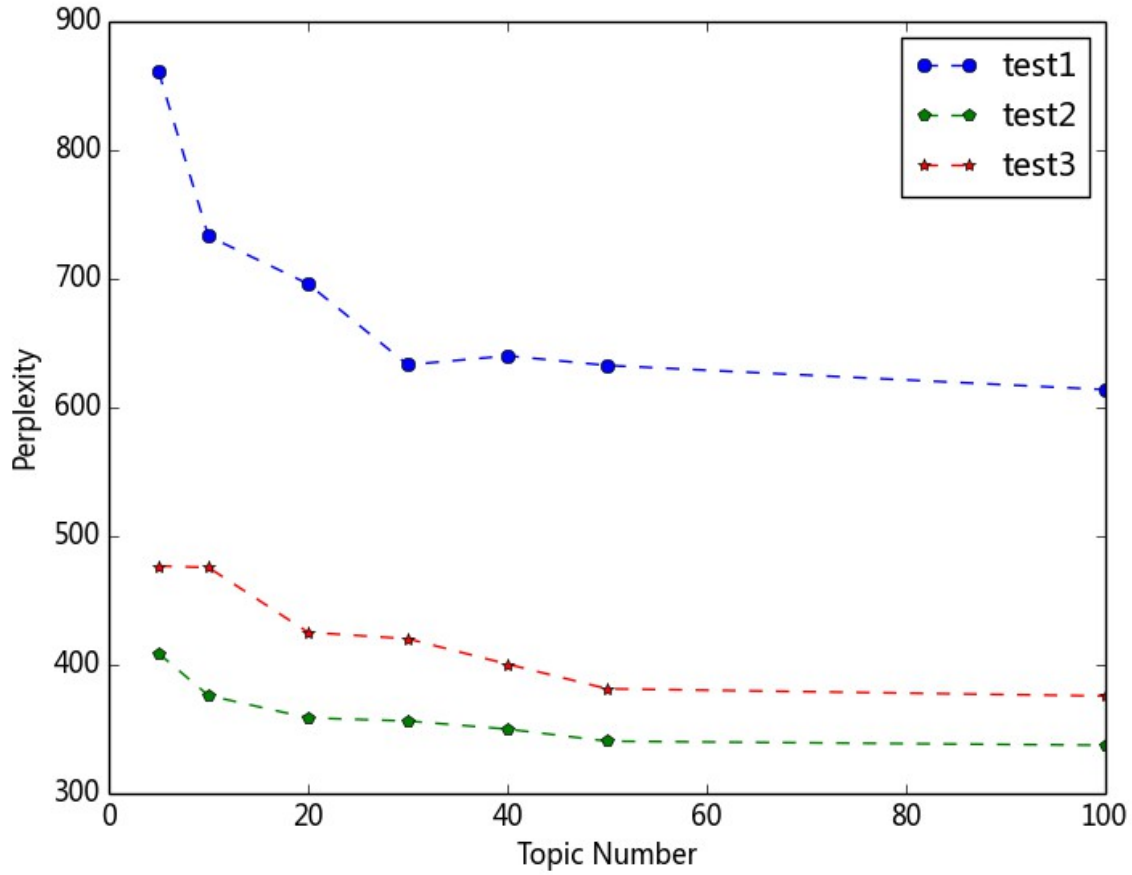
algorithm

```
Def find_affiliate(word,content) :          #定义一个从分段文档集中获取名词附属词的函数。
    new_content=[word,“   ”]                #预生成的附属词文档集，列表的第二项为附属词文档集
    content_copy=content[:]                 # 生成一个分段文档的副本
    for line in content:                    #循环读取分段文档集中的文档
        if word in line:                    # 如果目标名词在该分段文档中
            new_content[1]+=line            #则该分段文档所有词都是该词的附属文档，包括其本身
            content_copy.remove(line)       # 在分段文档集的副本中删除该文档
    return new_content,content_copy         #返回词－附属文档以及下一个词的待查找分段文档
content=[]                                  #第二步中，根据标点生成的分段文档集合
Noun=[]                                     #名词列表，根据名词排序生成的名词列表
Noun_affiliate=[]
for word in Noun:
    if len(content)==0:
        break                               #如果待查找分段文档集长度为空，则终止循环
    if word in content:
        new_content,content=find_affiliate(word,content)
                                            #如果 word 在 content 中，引用函数，生成该 word 的附属文档集。同时将分段文档集重置
return Noun_affiliate[word,word_affiliate]  #得到所需要的词-附属文档集合
```

手机/n	iphone/eng 现时/t 性价比/n 一款/m 手机/n 不错/a 果/ng apple/eng iphone/eng 5s/n 16/m g/eng 版/n 4g/n 手机/n 金色/n fdd/eng lte/eng td/eng lte/eng td/eng scdma/eng wcdma/eng gsm/eng 攒/v 物流/n 当天/t 下单/v 当天/t 到货/v 手机/n 盖/v 两个/m 黑点/n 接受/v 到手/d 三天/m 降价/n 吭/zg 京东/ns 物流/n 值得/v 肯定/v 下单/v 签收/v 时间/n 手机/n 耍/v 两天/m 死机/n 重启/v 电量/n 时间/n 无故/n 掉/v 很快/d 手机/n 很好/d 很漂亮/n 质量/n 不错/a 手机/n 情况/n 不错/a 苹果/n 手机/n 信赖/n 苹果/n 手机/n 没话说/l 系统/n 给力/a 期待已久/i 土豪/n 金/nr 终于/d 到手/d 。。。
机子/n	拿到/v 机子/n 不错/a 发现/v 换/nz sim 卡/n 跑/v 好几个/m 地方/n 差点/n 动手/n 剪/v 还好/v 机子/n 当日/t 激活/a 有时候/l 发热/v 正规/a 发票/n 未/d 拆封/v 机子/n 通用/v 网络/n 正版/v 机子/n 不错/a 机子/n 实用/v 机子/n 还行/v 自动/vn 卡死/nr 自动/vn 跳/v 回主/v 屏/n 反正/d 保修/v 一会/m 机子/n 热/a 暂时/d 发现/v 评论/n 担心/v 运气/n 还好/v 一点/m 查/v 序列号/n 10/m 月/m 20/m 号/m 生产/vn 机子/n 机子/n 新/a 顺畅/a 机子/n 不错/a 发现/v 全新/d 未激活/i 不错/a 新/a 机子/n 14/m 年/m 快递/v 给力/a 。。。
价钱/n	还行/v 听/v 顺手/d 价钱/n 高/a 一分/m 价钱/n 一分货/n 高大/a 价钱/n 降低/v 星期/t 降/v 100/m 块/q 不错/a 奥/nr 掉/v 太/d 价钱/n 不错/a 一分/m 价钱/n 一分货/n 不错/a 价钱/n 优惠/vn 服务到位/n 送货/v 到货/v 很快/d 价钱/n good/eng 不错/a 价钱/n 公道/n 实惠/vn 草/n 显示/v 无卡/ns 花/v 价钱/n 买/v 不/d 舒心/a 一分/m 价钱/n 一分货/n 不错/a 上个月/t 入手/v a1518/eng 4799/m a1530/eng 4799/m 20/m 抵用/vn 券/n 4779./m。。。
价格/n	价格/n 确实/ad 不错/a 商场/n 便宜/a 价格/n 实惠/vn 不错/a 价格/n 合适/a 送/v 贴膜/n 完美/a 送给/v 老婆/n 天/q 暂时/d 发现/v 考察/v 送货/v 很快/d 价格/n 实惠/vn 买/v 第三个/m 价格/n 实惠/vn 性价比/n 高/a 价格/n 合适/a 土豪/n 金/nr 高大/a 价格/n 给力/a 幸好/a 双十/m 价格/n 价格/n 算/v 土鳖/n 买得起/v 快递/v 效率/n 高/a 昨天/t 下单/v 支持/v 买来/v 送/v 价格/n 浮动/vn 很大/a 不错/a 价格/n 算是/v 优惠/vn 第二次/m 京东上/n 买/v 类似/v 价格/n 电子产品/n 还好/v 失望/v 看着/v 评论/n 犹豫/a 很久/m 买/v 到手/d 查/v 。。。
外观/n	外观/n 漂亮/a 唯一/b 音量/n 键/n 松动/a 整体/n 不错/a 网上/s 查/v 新机/n 确实/ad 不错/a 喜欢/v 外观/n 超/v 喜欢/v 通话质量/n 外观/n 漂亮/a 功能齐全/i 通话音质/l 功能齐全/i 外观/n 漂亮/a 值得/v 拥有/v 外观/n 磨损/vn 外观/n 磨损/vn 外观/n 磨损/vn 外观/n 磨损/vn 外观/n 磨损/vn 外观/n 很漂亮/n 好看/v iphone5s/n 外观/n 不错/a 外观/n 漂亮/a 喜欢/v 一惯/n 喜欢/v 外观/n 不/d 解释/v 软件/n 不错/a 外观/n 漂亮/a 待机时间/n 长/a 顺手/d 外观/n 很漂亮/n 分辨率/n 高/a 外观/n 漂亮/a 铃声/n 外观/n 漂亮/a 外观/n 颜色/n 通话/n 清晰/a 外观/n 漂亮/a 操作/v 真心/d 值/n 钱/n 性价比/n 外观/n 不錯/a 用起/v 麻烦/an 外观/n 向往/d 值得/v 赞/v 外观/n iphone5/eng 指纹/n 还好/v 不错/a 外观/n 很漂亮/n 。。。
金色/n	帮/v 同事/n 买/v 额/n 两个/m 几个/m 金色/n 黑色/n 不错/a 满意/v 网购/n 金色/n 黑色/n 贵/a 几百块/m 办法/n 不/d 稀饭/n 黑色/n 金色/n 16/m g/eng 喜欢/v 摸索/v 金色/n 神机/n 过段时间/n 入手/v 小苹果/nr 金色/n 确实/ad 细致/a 送货/v 金色/n 漂亮/a 好看/v 不错/a 外面/f 便宜/a 一点/m 金色/n 好看/v 金色/n 小苹果/nr 不用说/l 超赞/v 查过/v 正宗/nz 国行/n 放心/v 金色/n 太/d 好看/v 不错/a 喜欢/v 金色/n 外壳/n 太潮/nr 金色/n 不是太好/l 买/v 送/v 老爸/n 喜欢/v 金色/n 不俗/a 买/v 金色/n 后悔/v 黑色/n 没用/v 小时/n 盖/v 花/v 金色/n 弱/a 金色/n 。。
...	…

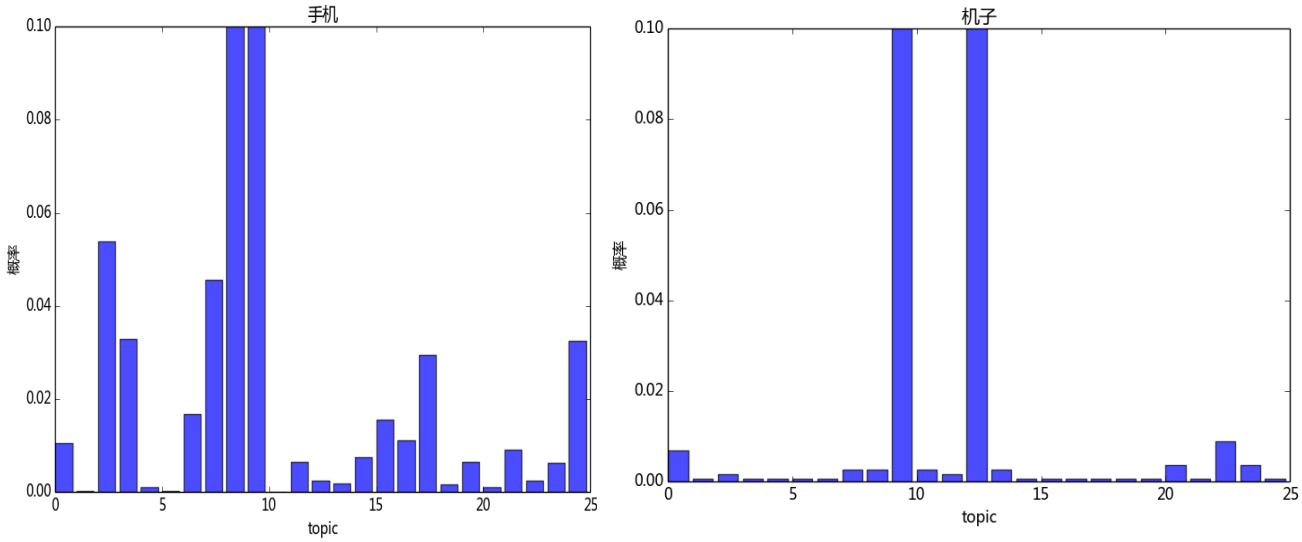
- GibbsLDA 主题抽取

在评价语言模型时候，通常使用困惑度。因而用 lda 抽取文档的 TOPIC 时，我们选择用困惑度来确定 TOPIC 数量，在数据集上，困惑度曲线如图所示。困惑度最小值位于 T=25 处，因此，参数 T 取 25。

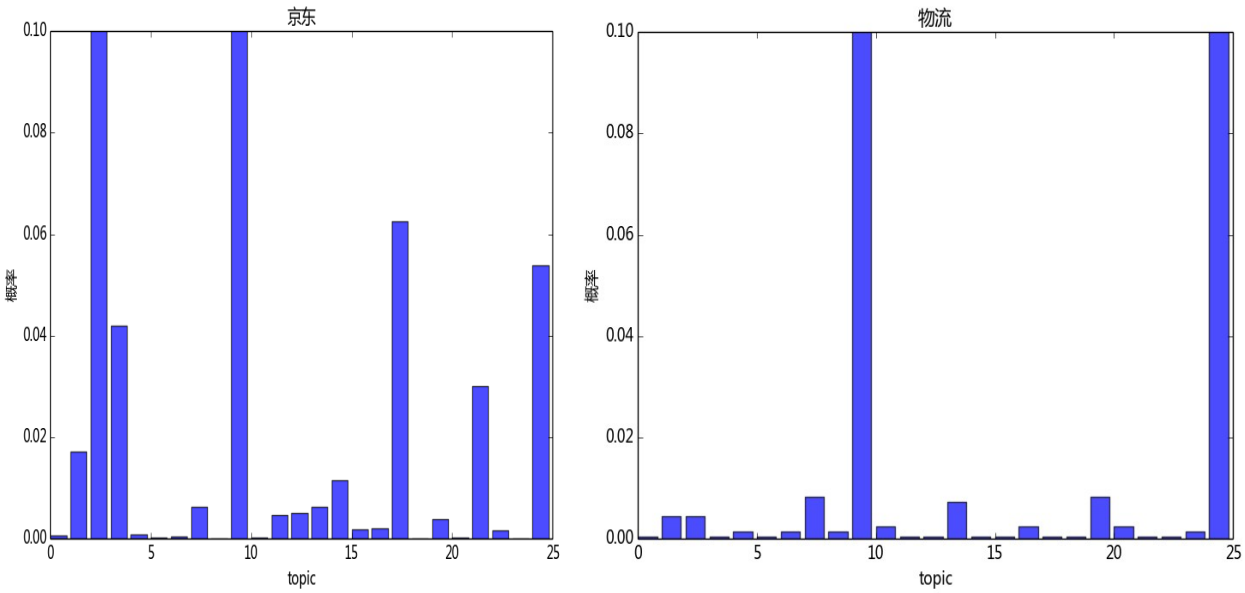


利用 LDA 抽取出的语言模型，对每一个名词进行推断，得到各名词的语义分布。

手机 VS 机子



京东 VS 物流



- 名词与名词间的相似度

通过 Gibbs 抽样，我们得到了共 170 个名词的语料集合的 TOPIC 分布。对每一个名词进行 TOPIC 推断，我们得到了 170 个名词以及该名词的语义分布。要利用名词的语义分布计算名词之间的相似程度，即计算离散分布之间的相似度，有两种较好的方法。一是余弦相似度[Steyvers M, Griffiths T. Probabilistic topic models[J]. Handbook of latent semantic analysis, 2007, 427(7): 424-440.]，二是 KL-距离[Rosen-Zvi M, Griffiths T, Steyvers M, et al. The author-topic model for authors and documents[C]//Proceedings of the 20th conference on Uncertainty in artificial intelligence. AUAI Press, 2004: 487-494.]。

A ,余弦相似度。将每个点的概率值作为权重。

$$Similarity_{cosine}=\frac{\sum(Xi*Yi)}{\sqrt{(\sum(Xi^2)+\sum(Yi^2))}}$$

B .KL 距离。分别将每个名词的 TOPIC 分布视为基础模型 Q，将其他名词的 TOPIC 分布 P 与其比较。

$$Similarity_{KL}(N1,N2)=-\sum(P(x)*logQ(x)+Q(x)*logP(x))$$

在实验中，选取其中一种相似度。用标记数据测试相似度准确率，按正确率和召回率来选择哪一种相似度。（待实验。）

3 ，生成概念树。

---

1 ，对于初始树上的每一个结点 G （w_1,w_2,..w_k）

$p(w_k)=\frac{TF-IDF_K}{\sum(TF-IDF_i)}$ #把每一个结点当作一篇文档，结点上的词按 TF-IDF 概率化。

2 ，for Gi in InitialTree:

for n in noun:

$p(n|w_k)=\frac{similarity(n,w_k)}{\sum(similarity(n_i,w_k))}$

$P(n|G)=p(n|w_1,w_2..w_k)=\sum p(n|w_k)*p(w_k)$

3,更新树，结点为 $G=(l_1,l_2,l_3,...,l_k)=(p(l_1),p(l_2),p(l_3)...p(l_k))$

4,评论语料的所有词　word=$(w_1,w_2,w_3...w_m)$

名词的附属文档 $content_{Ni}=(p_1,p_2,p_3...p_n)$

以词频作为概率 $content_{Ni}=(p_{i1},p_{i2},p_{i3}...p_{in})=(TF_{i1},TF_{i2},TF_{i3}...TF_{in})$

for w in word:

$p(w|G)=p(w|l_1,l_2,l_3,...,l_k)=\sum p(w|l_k)*p(l_k)$

5,更新树，生成的树，每一个结点包含评论语料中所有的词以及词的概率。

步骤 3 更新树如下表（部分结点）：

结点	词-概率
根结点	
第一层结点	
第二层结点 1	
第二层结点 2	
第三层结点 3	

最终生成的概念树如下表（部分结点）：

结点	词-概率
根结点	
第一层结点	
第二层结点 1	
第二层结点 2	
第三层结点 3	

四，利用概念树对产品评论进行推断。