

016050市ro30t0101010r0167040100195rE9r0蛇Pu0iatQ1040001D1010010r0100r010rt 0ei0100srntD¹00t01010otut ⁰019ur0r0106QF0101010T00rer0500191010i0e01910

iQ01010r010r0j0rorerG13700(0re70100201007001D0101010T00T0F¹010910101010

1T9708Q 0 1 0 0 0 0 1 0 任 19oreV0400101001019R100701D1011r0i0861QKCI010103010101010501S1000

**第** **7** **章** **文本分析**



**7.1** **文本分析的意义**

根据估算，各类组织(包括企业、政府)拥有的数据里80%是非结构化的数据，其中 大部分是文本的形式。非结构化的文本数据，包括电子邮件 (Email) 、 博客、微博、客户 反馈、医疗记录、合同文本等。这些文本里隐藏着潜在的价值。只有通过适当的分析方 法，才能从中提取这些有价值的信息。

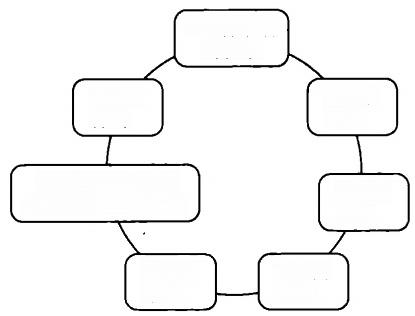
通过如下实例，我们来了解文本分析的意义。各个企业都非常关注它们的客户在想些 什么。企业希望借此改善产品和服务，保留已有的客户，开拓新的客户。通过结构化数据 分析(销售数据),企业可以了解已经卖出了什么产品和服务，哪一款产品经常被退回， 哪一款产品经常需要保修等。此外，客户还通过其他的渠道表达对产品和服务的看法，比 如在博客、产品论坛等网上虚拟空间，讨论产品和服务的优点和缺点，表达他们的喜好和 反感。通过适当的文本分析技术，企业可以深入了解这些信息，作为结构化数据分析的补 充，进而对产品和服务做出改进。

文本分析 (Text Analytics) 也称为文本挖掘 (Text Mining), 指的是从文本里分析 提取高质量信息的过程。文本分析的主要任务包括文本索引与检索 (Indexing and Search) 、 文本分类 (Text Classification)、文本聚类 (Text Clustering)、文档摘要 (Doc- ument Summarization)、主题抽取 (Topic Theme Extraction)、命名实体识别/概念抽取/ 关系建模 (Named Entity Recognition,Concept Extraction,Relation Modeling)、情感分 析 (Sentiment Analysis) 等。从文本中提取信息 (Information Extraction) 的方法和结构 化数据分析的方法是不一样的。文本分析如图7-1所示。

为了完成这些任务，需要结合使用语言学 (Linguistics) 、 统计学 (Statistics) 、 数据 挖掘和机器学习 (Data Mining &Machine Learning)、自然语言处理 (Natural Language Processing,NLP) 、 信息检索 (Information Retrieval) 等技术和方法。

文本分析的过程包括几个主要的步骤：(1)采集文本数据集；(2)运用文本分析方法 分析文本；(3)对分析结果进行可视化以及解释和评估分析结果等。





1.文件索引 与检索

2.文本 分类

6.命名实体识别/ 概念抽取/关系建模

4.文档 摘要

3.文本 聚类

7.情感 分析

5.主题 抽取

图 7 - 1 文本分析

**7.2** **文本分析的任务和方法**

**7.2.1** **句子切分、分词、词性标注、语法分析**

**1.** **句子切分和分词**

为了对文本进行分析，首先需要把文本切分成一个一个的句子。完成该功能的软件称 为语句切分器 (Sentence Detector,也 叫Chunker) 。 接着需要对句子进行分词，完成该功 能的软件称为分词器 (Tokenizer)。 句子切分和分词已经是一项成熟的技术。

**2.** **词性标注**

词性标注 (Part-of-Speech Tagger, 简 称POS Tagger) 软件分析某种语言的文本，然 后针对每个词 (Word 或 者Token) 赋 予 POS 标记，比如名词 (Noun)、 动 词 (Verb)、

形 容 词 (Adjective) 等 。 以 斯 坦 福 大 学 开 源 的 POS Tagger(Stanford Log-linear POS Tagger①) 为例，它使用了条件对数线性模型 (Conditional Loglinear Model) 实现词性的 标注。在这个软件中，已经为英语训练好了词性标注模型，还提供了阿拉伯语、中文、法 语、德语等语言的词性标注模型，方便用户使用。此外，用户可以针对某种目标语言重新 进行模型的训练，需要用户提供已经进行POS 标注的训练文本 (POS-Annotated Training Text) 。 斯坦福大学开源的 POS Tagger 利用了文本里的词法特征以及词性标记的上下文 信息，提高词性标注的正确率。

进行词性标注还可以使用基于条件随机场的方法以及基于最大熵 (Maximum-Entro-

py) 的方法等。在“命名实体识别” 一节，对条件随机场技术进行了简单介绍；在“情感 分析” 一节，对最大熵模型进行了简单介绍。

**3.** **语法树**

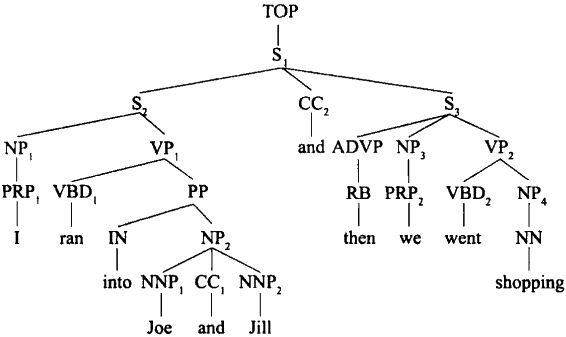
在自然语言处理中，语法解析器 (Parser) 接收语句，并且对句子的语法结构 (Gram-



①<http://nlp.stanford.edu/software/tagger.shtml.>

matical Structure) 进行分析，输出语法解析树 (Parser Tree) 。Parser首先对句子的文本进行 分词，然后进行POS 标注 (POS Tagging),根据 POS 标注结果以及句子成分信息，找出单 词/短语之间的依赖关系 (Dependency), 最后构建句子的语法解析树，结果以有向图或者树 的形式展示。

比 如 ， 句 子 I ran into Joe and Jill and then we went shopping, 经 过 Parser 的 语 法 分 析，输出 [TOP [S [S [NP [PRP I]][VP [VBD ran][PP [IN into][NP [NNP Joe][CC and][NNP Jill]]]][CC and][S [ADVP [RB then]][NP [PRP we]] [VP [VBD went][NP [NN shopping]]]]]]。表达成一棵语法树，更容易观察一些，如 图7-2所示，其中的NP,VBD,CC,ADVP 等表示具体的POS 标注，分别表示名词短 语、动词、连接词、副词等词性。①



**图** **7** **-** **2** **语法树**

实现语法树分析的具体技术包括：(1)概率型上下文无关语法分析技术 (PCFG- Probabilistic Context-Free Grammar),这个方法可以是考虑词汇的 (Lexicalized), 也可 以是非考虑词汇的 (Unlexicalized);(2) 移位归约语法分析技术 (Shift-Reduce Constitu- ency Parser);(3) 基于神经网络的语法分析技术 (Neural Network Dependency Parser)。 这些技术已经非常成熟，能够获得准确的分析结果。技术的具体细节，读者可以参考相关 资料 。

**7.2.2 文本索引和检索** **(Indexing and Search②**)

信息(这里主要指文本)检索是针对用户提出的信息需求， 一般是以关键字 (key word) 表达的查询，从文档集中查找和查询相关度高的文档或者文档片段，返回给用户。

信息检索系统一般包括四个主要部分，分别是数据预处理、索引生成、检索、结果排 序。数据预处理的目的是从网页、PDF 文件、Word 文件等文档中提取正文以及文档元信 息。索引生成是为这些文档的每个词项生成倒排索引(下文具体介绍)。检索则是根据用

① 其他更多的标记及其含义，可以参考 [https://www.ling.upenn.edu/courses/Fall\_2003/ling001/penn \_tree](https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_tree)- bank\_pos.html。

②Information retrieval 一般译作信息检索，search一般译作搜索，比如search engine译作搜索引擎。在本书中， 检索和搜索有时候交换使用，意思是从文档集中找出若干符合用户信息需求的文档或者文档片段。

户提交的查询，利用预先建立好的索引，从文档集里提取相关的文档。结果排序则根据文 档本身的重要性，以及文档和查询的相关度，对文档进行排序，把最有可能符合用户需求 的文档排在前面，尽早返回给用户(查询结果包含很多文档时， 一般一次仅返回少数几个 文档，用户可以通过翻页操作查看其他文档)。

主要的信息(文本)检索模型包括布尔模型、向量空间模型以及概率模型等。这里仅 介绍向量空间模型，另外两个模型，请读者参考相关资料。

**1. 向量空间模型与** **TF-IDF**

目前主要的文档表示模型是Gerard Salton 和 McGill 于1969年提出的向量空间模型 (Vector Space Model,VSM)。在向量空间模型里，文档表示为一个向量，向量的分量为 特征项的权重 (w₁,w₂, … ,wn), 其中w;为第i 个特征项的权重。一般选取单词作为特 征项，即一个单词是一个词项。

这里的权重用词频表示。词频分为绝对词频和相对词频。绝对词频用词项在文档中出 现的频率表示。我们可以基于绝对词频进行文档(网页)检索。比如查询的关键字为 “JohnBlog”, 当文档A 出现 “John”5 次，出现 “Blog”10 次 ，B 文档出现 “John”2 次 ， 出现 “Blog”8 次，我们基于绝对词频计算文档和查询的匹配度，文档A 的匹配度为5+ 10=15,文档B的匹配度为2+8=10,于是文档A 的匹配度高于文档B 。绝对词频具有 明显的不合理性，内容较长的文档，更有可能比内容较短的文档出现更多的关键字。①虽 然长文档出现更多的关键字，但是相对于文档长度来讲，关键字显得相当稀疏；短文档虽

然出现更少的关键字，但是相对于文档长度来讲，关键字可能显得相当密集。从这个道理 来讲，短文档和查询更加匹配，相对词频由此引人。相对词频是归一化的词频，其计算方 法主要是TF-IDF(Term Frequency-Inverse Document Frequency) 公式。

**TF** **的计算方法为：**



**这个值越大，表示这个词项越重要。比如，** **一篇文档进行分词以后，总共有500个词** **项，词项** **“World” 出现的次数是3次，那么其TF 值为** **TF=3/500=0.006。**

**IDF** **的计算方法为：**



这个公式的意义是一个词项出现的文档数越少，它越能够把文档区分出来，于是就越 重要。或者反过来说，一个词项如果在每篇文档里都出现，那么它就没有那么重要了。比 如一个文档集包含10000000个文档，“World” 这个词项在1000个文档中出现过，那么 它的IDF 为 log(10000000/(1000+1))=3.999。

TF-IDF 公式把TF 和 IDF 乘起来，计算词项的权重，即TF-IDF=TF×IDF。

为了对文档集进行索引，一般需要进行分词 (Tokenization) 、 词形还原 (Lemmatiza- tion) 或者词干提取 (Stemming) 。 词形还原和词干提取都是词形规范化的重要方式，都 能够达到有效归并词形的目的。词形还原是把任何形式的词汇还原为一般形式，还原以后 的形式能够表达完整语义。比如把 “driven”“drove”“driving” 等都处理为 “drive” 。 词



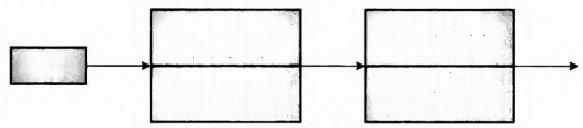
① 对 于key word, 有的译作“关键词”,有的译作“关键字”,我们采用后者。



干提取则是抽取词的词干或词根形式(不一定能够表达完整语义)。比如，将“cats”处理 为 “cat”, 将 “effective” 处理为 “effect” 等。

相对而言，词干提取是轻量级的词形归并方式，最后获得的结果为词干，并不一定具 有实际意义。词形还原处理相对复杂，获得结果为词的原形，能够承载一定意义，与词干 提取相比，更具有研究和应用价值。词干提取更多应用于信息检索领域，比如Lucene 检 索软件等，其粒度较粗，可以用于扩展搜索，提高召回率。词形还原用于更细粒度、更为 准确的文本分析和表达，主要应用于文本挖掘、自然语言处理等。

在上述处理之后，对每个词项建立倒排表，倒排表记录了这个词项在每个文档出现的 次数、出现的位置等信息。比如词项 “World” 在文档1出现了10次、在文档5中出现了 20次，那么倒排表从逻辑上具有如图7-3所示的结构。每个进行索引的词项，都具有如 该图所示的倒排表。



World在文档5

出现的次数

World

World在文档5

出现的位置列表

World在文档1

出现的次数 World在文档1 出现的位置列表

图7-3 倒排表的逻辑结构

如果我们现在需要了解同时出现 “Hello”“World” 两个单词的文档，那么我们只需 提取两者的倒排表，看看它们分别在哪些文档中出现，对两个文档集取交集，即可知道同 时出现这两个单词的文档有哪些。比如，出现 “Hello” 的文档子集为文档1、文档5、文 档7等，包含 “World” 的文档子集为文档1、文档5等，那么 “Hello”“World” 两个单 词同时出现的文档为文档1和文档5。

上述检索的处理方式并没有对结果进行任何的排序。我们知道，从Google 等搜索引 擎返回的结果，是按照相关度进行了排序的，它是如何做到的呢?这里介绍基于向量空间 模型的余弦相似度计算方法。首先，文档可以表示成一个权重分量(也就是很多的词项) 构成的向量。查询(关键字查询.Key Word Search) 表示为若干词项组成的查询文档， 于是也可以表示成一个权重分量构成的向量，只不过很多的分量为0。余弦相似度通过向 量夹角余弦，表示两个向量的相似度，夹角越小，相似度越高(见图7-4)。

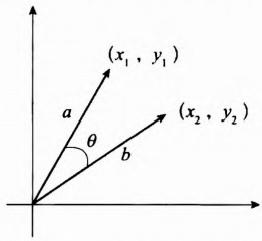


图7-4 两个向量的夹角(低维的二维平面)

余弦相似度的计算公式为 (q 为查询的向量表示，d 为文档的向量 表示)。余弦值 (cos) 越接近1,就表明夹角越接近0度，也就是两个向量越相似。

信息检索系统有两个重要的评价指标， 一个是准确率， 一个是召回率。表7-1的a,



b,c,d 分别表示被(检索系统)判断为相关的文档中的相关文档、被判断为相关的文档 中的不相关文档、被判断为不相关的文档中的相关文档、被判断为不相关的文档中的不相 关文档的数量。

**表7-1** **信息检索的混淆矩阵**

|  |  |  |
| --- | --- | --- |
|  | 实际上相关的文档 | 实际上不相关的文档 |
| 检索系统返回的判断为相关的文档 | a | b |
| 检索系统不返回的判断为不相关的文档 | C | d |

准确率 (Precision) 的计算公式为 ,召回率 (Recall) 的计算公式为

。除此之外，还有一个统一度量信息检索系统性能的指标为F 指标，它的

计算公式为 

准确率评价的是返回的结果中多少文档是相关的。召回率评价文档集中相关的文档， 检索系统返回了多少。F 值则是综合这两个指标的评估指标，用于反映信息检索系统的整 体性能。

不考虑词的相对位置关系的文档表示法也称为词袋 (Bag of Words) 表示法。使用词 袋表示法，每个文档表示为一个高维空间的一个稀疏向量，向量的维度是字典表 (Vocab- ulary) 的大小 |V|。 下面将要介绍的潜在语义分析，则把文档表示为一个k 维 (k 大大 小 于 |V|) 线性子空间 (Linear Subspace) 上的投影 (Projection)。

2. 潜在语义分析 (Latent Semantic Analysis,LSA)

潜在语义分析来源于如何利用查询 Query 搜索到相关的文档。如果我们仅仅通过关键 字匹配找到相关文档，则存在难以解决的局限性，某些没有这些关键字但是相关的文档不 能被检索出来。

潜在语义分析就是通过分析文档，去发现这些文档中潜在的意思和概念。在自然语言 的实际使用中，存在同义词和一词多义现象，造成概念混淆。我们在搜索中，需要去比较 的不是词项而应该是隐藏在词项背后的意义和概念。

潜在语义分析试图解决这个问题，它把词和文档都映射到一个“概念”空间，并且在 这个空间里进行比较，实质上这是一种降维技术，即把问题从高维空间降到低维空间 解决。

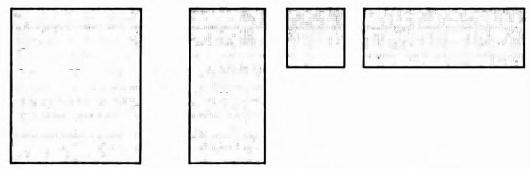
潜在语义分析依赖于奇异值分解 (Singular Value Decomposition,SVD)。奇异值分 解是因子分析 (Factor Analysis) 的一种形式，它建立了n 维的抽象语义空间，原有的词 项 (Term) 和文档 (Document) 都表示为其中的一个向量。其具体流程介绍如下。

首先，我们建立 Term-Document 矩阵，然后把该矩阵分解为三个矩阵 W,S 和 P 的 乘积。Term-Document 矩阵表达了哪个Term 在哪个Document 出现的信息(如图7 - 5 所示)。

其中，W 是一个标准正交 (Orthonormal) 矩阵，它的行对应X 的每一行，也就是各 个词项。它拥有 m 列，这些列向量之间是互相独立的，没有相关性。对应到m 个衍生的 奇异向量 (Derived Singular Vector),对应m 个维度。

P 矩阵是一个标准正交矩阵，它的每列对应X 的每一列，也就是各个文档，它拥有m 行，每行也对应衍生的奇异向量 (Derived Singular Vectors)。



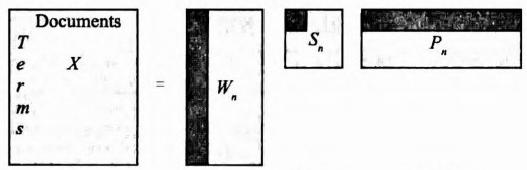


|  |  |  |  |
| --- | --- | --- | --- |
| **Documents**  T  e X r  m S | 二 | W | S P |
| W\*p |  | W\*m | *m\*m* ***m\*p*** |

**图** **7** **-** **5** **奇异值分解SVD**

S 矩阵是一个m\*m 的对角矩阵，在对角线上的元素是非零的奇异值 (Singular Val- ue) 。这些奇异值把W 和 P 矩阵的因子 (Factor) 关联起来，通过矩阵乘法，重建原有的 X 矩阵。S 矩阵从左上角到右下角，奇异值依次递减。大的奇异值表示该维度(维度从1 到 m) 在逼近的误差的平方和 (Sum Squared Error of the Approximation) 上的贡献。

进行奇异值分解以后，我们从m 个奇异值里面把n 个最重要的维度(因子)选择出 来，把其他的因子忽略掉。也就是仅仅保留S 矩阵的靠近左上角的较大的奇异值，同时W 矩阵的若干列(奇异向量， Singular Vector) 以及P 矩阵的若干行(奇异向量， Singular Vector) 一并删除，达到降维的效果。如图7-6所示。



W\*p W\*n n\*n n\*p

**图** **7** **-** **6** **奇异值分解以后的降维处理**

通过矩阵分解可以看出，原始矩阵中的t; 行只与W 矩阵的第i 行有关，我们称第i 行 为 t; 。 同理，原始矩阵中的d, 列只与P 中的第j 列有关，我们称这一列为 dj。

n 的大小的选择是一个至关重要的问题。理想情况下，n 应该足够大，以便拟合实际 的数据，同时n 又足够小，以便噪声、采样误差、无关紧要的细节都没有被包含到模型 里。在信息检索里，可以用检索的准确率、召回率等来做评价标准。

上述降维方案把每个文档从原来的文档所包含的词项的高维表示形式，缩减到了一个 n 维的向量表示，同时又保留了原来数据中的重要结构和模式。于是某些相关文档，虽然 并不包含查询的词项 (Query Term),也可以提取出来。

传统基于关键字匹配的索引与信息检索，有两个根本问题 (Fundamental Problem)。

(1)同义词 (Synonymy), 人们可以用多个词表示同一个概念，查询里的词项 (Query Term) 不一定和文档的描述匹配，比如我们查询 “Steve Jobs”, 包含 “iPhone” 的文档 有时候也是相关文档。(2)一词多义 (Polysemy), 也就是一个词语可以有多种意思，用 户的查询可能匹配无关的意思，于是把无关的文档提取出来。比如我们查询 “Apple”, 我 们希望把苹果公司的相关文档提取出来，但是系统可以提取关于水果的文档。 SVD 一 定 程度解决问题(1),但是它不能有效解决一词多义。它假设同样的词有同样的概念，于是 就解决不了需要根据语境才能确定其具体含义的词，比如 Bank, 如果跟贷款、存款联系 在一起，一般是银行的含义，如果和小河、小溪联系在一起，一般表示堤岸。

至此，我们看到LSA 通过对Term-Document 矩阵的奇异值分解，把Term-Document 关系缩减为一个近似表示。相对于原始矩阵，概念空间的维度大大降低，而且这些维度包 含了大量的信息和最少的噪声，有利于后续的分析。

(1)判断文档j 与 q 在低维空间的相似度。比较向量S,d; 与向量Sdg, 比如使用余 弦夹角，即可得出。

(2)通过比较 t₁Sn 与 iS。 可以判断词i 和 词p 的相似度。

(3)给定一个查询，计算其在语义空间内和已有文档的相似性，需要把文档和查询字 符串都映射到语义空间。对于原始文档，映射公式为 d;=S,-¹W,Td,(n\*1=n\*n×

n\*w×w\*1) 。 对于查询字符串，转换成相关词项对应的向量后，用公式 q=

S.-¹W.Tq(n\*1=n\*n×n\*w×w\*1) 进行映射。最后在语义空间里，比较查询和文档 之间的相似度。

(4)有了相似度，就可以对文档进行聚类。

我们可以在奇异值分解结果上执行凝聚层次聚类算法 (Agglomerative Hierarchical Clustering), 把语义接近的文档组织起来。首先，每个文档作为一个类簇。然后，最相似 的两个类簇合并成一个新的类簇。这个步骤不断迭代，直到所有文档都合并到一个大的类 簇中。

图7-7展示了5个文档的可能的层次聚类结果。左边显示聚类的层次，右边则显示 具体的各个类簇。在层次聚类过程中，计算文档类簇之间的相似度，可以使用全连接 (Complete Linkage) 算法。

|  |  |
| --- | --- |
|  | (d1,d2,d3,d4,d5) (d1,d2,d3,d4),(d5)  (d1),(d2,d3,d4),(d5)  (d1),(d2,d3),(d4),(d5)  (d1),(d2),(d3),(d4),(d5) |
| d2 d3 d4 d5 |  |

**图7-7** **文档的层次聚类**

我们通过一个实例①来理解 SVD 与 LSA 。 假设有9个标题 (Title, 每个标题作为一 个小的文档),出现2次以及2次以上的非停用词 (None Stop Word) 作为索引词。标注 如下。停用词是常见的对确定文档的含义没有多大意义的词项，比如a,the 等。

|  |
| --- |
| 1.The Neatest Little Guide to Stock Market Investing  2.Investing for Dummies,4th Edition  3.The Little Book of Common Sense Investing:The(nly Way to Guarantee Your Fair Share of Stock Market Returns  4.The Little Book of Value Investing  5.Value Investing:From Graham to Buffett and Beyond |



①<https://gist.github.com/vgoklani/1267632.>

6. Rich Dad's Guide to Investing :What the Rich Invest in,That the Poor and the Middle Class Do Not!

7. Investing in Real Estate,5th Edition

8. Stock Investing for Dummies

9. Rich Dad's Advisors:The ABC's of Real Estate Investing:The Secrets of Finding Hidden Profits Most Investors Miss

基于上述文档(标题)集，我们建立如下词项-文档矩阵。

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Index Words | Titles | | | | | | | | |
|  | T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 |
| book |  |  | 1 | 1 |  |  |  |  |  |
| dads |  |  |  |  |  | 1 |  |  | 1 |
| dummies |  | 1 |  |  |  |  |  | 1 |  |
| estate |  |  |  |  |  |  | 1 |  | 1 |
| guide | 1 |  |  |  |  | 1 |  |  |  |
| investing | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| market | 1 |  | 1 |  |  |  |  |  |  |
| real |  |  |  |  |  |  | 1 |  | 1 |
| rich |  |  |  |  |  | 2 |  |  | 1 |
| stock | 1 |  | 1 |  |  |  |  | 1 |  |
| value |  |  |  | 1 | 1 |  |  |  |  |

这个矩阵里的计数可以用TF-IDF 权重代替，计算办法请参见本节开始部分对向量空 间模型的介绍。对该 Term-Document 矩阵进行奇异值分解，并保留最重要的三个奇异值， 得到的结果如下：

**W**

|  |  |  |  |
| --- | --- | --- | --- |
| book | 0.15 | 一0.27 | 0.04 |
| dads | 0.24 | 0.38 | 一0.09 |
| dummies | 0.13 | 一0.17 | 0.07 |
| estate | 0.18 | 0.19 | 0.45 |
| guide | 0.22 | 0.09 | 一0.46 |
| investing | 0.74 | 一0.21 | 0.2 |
| market | 0.18 | 一0.30 | 一0.28 |
| real | 0.18 | 0.19 | 0.45 |
| rich | 0.36 | 0.59 | 一0.34 |
| stock | 0.25 | 一0.42 | 一0.28 |
| value | 0.12 | 一0.14 | 0.23 |

S

|  |  |  |
| --- | --- | --- |
| 3.91 | 0 | 0 |
| 0 | 2.61 | 0 |
| 0 | 0 | 2 |

P

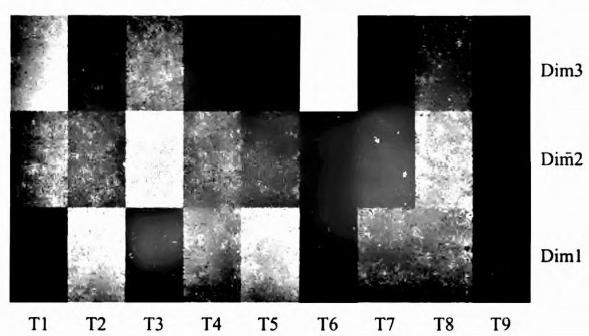
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| T1 | T2 | T3 | T4 | T5 | T6 | T7 | T8 | T9 |
| 0.35 | 0.22 | 0.34 | 0.26 | 0.22 | 0.49 | 0.28 | 0.29 | 0.44 |
| 一0.32 | 一0.15 | 一0.46 | 一0.24 | 一0.14 | 0.55 | 0.07 | 一0.31 | 0.44 |
| 一0.41 | 0.14 | 一0.16 | 0.25 | 0.22 | —0.51 | 0.55 | 0 | 0.34 |

我们把数字转换为颜色(图7-8中为不同的灰度),对矩阵 P 进行颜色表示，每一列 对应一个文档。从这个颜色表示来看，我们看到第一个维度上，各个文档的颜色差别不大 (见图7-8)。

我们可以在第二个维度和第三个维度上对文档进行聚类，聚类结果如图7-9所示。右 上角的聚类类簇包含了标题T7 和 T9, 是关于房地产 (Real Estate) 的。中间的聚类类簇包 含了标题T2,T4,T5 和 T8, 是讲价值投资 (Value Investing) 的。左下角的聚类类簇包含 了标题T1 和T3, 是关于股票市场 (Stock Market) 的。标题T6 则是一个孤立点 (Outlier)。



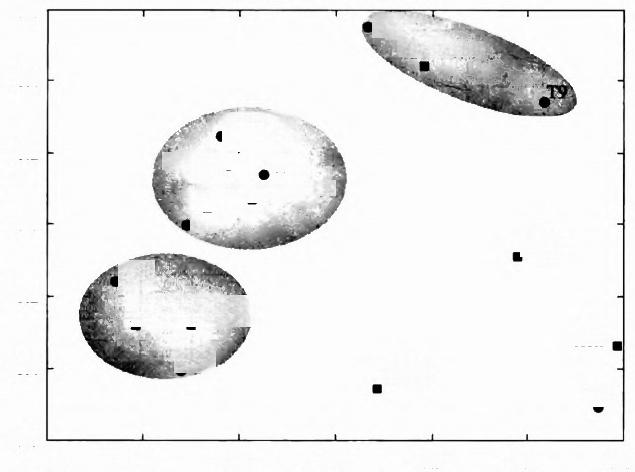
-101



Book Title

**图7** **-** **8** **文档降维以后的颜色表示**

***XY Plot of Wordsand Title***

0.6

*T7*

real estate

0.4

T4value

T5

0.2

T2

Dimension 3

investing

book■ T■8

dummies

0.0

dads

-0.2

T3

stock marke

rich

-0.4

T1

guide

T6

-0.6

-0.6

-0.4 -0.2 0.0 0.2 0.4 0.6 Dimension 2

**图7** **-** **9** **文档聚类结果**

**3. 开源搜索引擎Lucene**

Lucene 是一个开源的全文索引和检索软件包，它提供了完整的索引引擎和查询引擎， 以及部分文本分析引擎(英文、德文等)。理论上讲，使用Lucene 可以对互联网的网页进 行索引和检索，但是必须首先把互联网上出现的网页爬取下来，网页爬虫程序 (Web

Crawler) 实现这个功能。

Lucene 的原作者是Doug Cutting, 他是 一 位资深的全文索引、检索专家，也是大数 据处理工具Hadoop 项目的创建人。 Lucene 早先发布在作者自己的网站上 (<http://lu->



cene.apache.org/), 后来发布在Sourceforge 网站上。2001年底成为Apache 软件基金会 Jakarta项目的一个子项目，现在则是Apache 的顶级项目(<http://lucene.apache.org/)。>

Lucene 为各种应用程序构建全文检索功能。 Wikipedia,TheServerSide,jGuru 和 LinkedIn 等网站，以及 Apache 软件基金会的网站，都使用了Lucene 作为全文检索的 引 擎 。

目前Apache Lucene项目包含三个部分：(1) Lucene Core 。Lucene 核心子项目，用 Java 语言实现，提供了文本索引和检索 (Indexing &Search)、拼写检查 (Spellcheck- ing) 、查询词的高亮显示 (Hit Highlighting) 以及各种灵活的分词器 (Tokenization)。

Lucene 拥有强大、高效的搜索算法。通过编程，支持许多查询类型，比如短语查询 (Phrase Query)、通配符查询 (Wildcard Query)、范围查询 (Range Query)、模糊查询 (Fuzzy Query)、布尔查询 (Boolean Query) 等。此外，它能够解析人们输人的丰富查询 表达式。(2) Solr 。Solr 是建立在 Lucene Core 上完整的高性能的搜索服务器，提供了 XML/<HTTP> 以及JSON/Python/Ruby 应用程序编程接口 (API), 此外，还提供分面搜 索 (Faceted Search)、缓 存 (Caching) 、 复制 (Replication) 以及 Web 管理接口 (Web Admin Interface) 等高级功能。(3) PyLucene 。是 Core 子项目的一个Python 语言移植版 本 (Python Port)。

Lucene 项目的优点包括：(1) Lucene 的索引文件格式独立于应用平台。它定义了一 套以8位字节为基础的索引文件格式，使得不同平台的应用能够共享索引文件。(2) Lu- cene 在传统全文检索引擎的倒排索引的基础上，实现了分块索引。能够针对新的文件建立 小文件索引，提升索引速度。通过与原有索引的合并，达到优化的目的。(3) Lucene 的 查询引擎，默认实现了布尔操作、模糊查询 (Fuzzy Search)、分组查询等。通过 Lucene, 用户获得强大的文本查询能力。

**7.2.3** **文本分类**

文本分类 (Text Classification) 是把文档集合中的每个文档划分到一个预先定义的主 题类别 (Predefined Subject Category)。文本分类是文本分析和挖掘的一项重要工作。把 电子邮箱收到的邮件适当进行分类，分为正常邮件和垃圾邮件 (Spam), 就是文本分类的 一个应用实例。此外，确定文档的作者 (Authorship Identification)、确定文档作者的年 龄/性别 (Age/Gender Identification)、确定文档使用的语言 (Language Identification)、 确定整篇文档的情感 (Sentiment Analysis) 等，都可以看作一个文本分类问题。文档分 类①以后，用户可以通过类别来查看文档，方便浏览文档。

文本分类系统的主要功能模块包括：

(1)预处理器。把文档集合中的文档格式化为某种格式，方便后续处理。

(2)统计。进行词频统计，词项与分类的相关概率的统计。

(3)特征抽取。从文档中抽取反映文档主题的特征。

(4)分类器的训练。利用文档集部分文档的特征，对分类器进行训练。

(5)进行预测。利用分类器确定其他文档的类别。

由此可见，文本分类是典型的有监督学习的例子，训练集由已经明确分好类别的文档

①在下文中，文本分类和文档分类互换使用，表达同一个意思。

**组成，文档就是输入，对应的类别就是输出。人们尝试使用包括** **Rocchio 算法、k 最近邻** (**kNN)** **算法、朴素贝叶斯** **(Naive** **Bayes)** **算法、支持向量机** **(SVM)** **算法、决策树算法** 以及神经网络算法等，实现文本的分类。

Rocchio算法的基本思路是把一个类别里的样本文档各项取平均值，得到一个新的向 量，称为“质心”。“质心”成了这个类别最具代表性的向量表示。如果有新的文档需要判 断类别，比较新文档和各个“质心”的距离，就可以确定新文档的可能类别。 Rocchio算 法可以进行适当的改进，一种改进方法是，算法不仅考虑属于某个类别的文档的正例，而 且考虑不属于某个类别的文档的负例，计算出来的“质心”尽量靠近正例样本，同时尽量 远离负例样本。Rocchio 算法认为一个类别的文档聚集在一个“质心”的周围，实际情况 往往并非如此。其他算法的核心思想可以通过第5章“数据的深度分析”具体了解。下面 通过一个实例①,介绍如何使用朴素贝叶斯算法实现文本分类。

比如，现在有如表7-2所示的文档集，前4个文档是训练集，最后1个文档需要确 定类别。文档分为两类，分别是日本 (Japan) 和中国 (China)。

表7-2

|  |  |  |  |
| --- | --- | --- | --- |
|  | Doc | words | Class  c:China,j:Japan |
| training | 1 | Chinese Beijing Chinese | C |
| 2 | Chinese Chinese Shanghai | c |
| 3 | Chinese Macao | c |
| 4 | Tokyo Japan Chinese | j |
| testing | 5 | Chinese Chinese Chinese Tokyo Japan | ? |

根据表7-2,我们采用词袋模型 (Bag of Words) 的文档表示法获得如下先验概率，

P(c)=3/4,P(j)=1/4 (备注：P(c)=N./N)。 式 中 ，c 为 China;j 为Japan。

并且获得如下条件概率：

|  |  |
| --- | --- |
| P(Chinese | |c)=(5+1)/(8+6)=6/14=3/7 |
| P(Tokyo | |c)=(0+1)/(8+6)=1/14 |
| P(Japan | |c)=(0+1)/(8+6)=1/14 |
| P(Chinese | |j)=(1+1)/(3+6)=2/9 |
| P(Tokyo | lj)=(1+1)/(3+6)=2/9 |
| P(Japan | |j)=(1+1)/(3+6)=2/9 |

(备注：P(w|c)=(count(w,c)+1)/(count(c)+|V|),count(w,c) 表示类别c 里w 出现的次数，count(c) 表示类别c 的单词的数量，V 为词汇表， |V| 为词汇表大小。)

利用贝叶斯定理，P(c,I w₁,W₂,…,wn)=P(w₁,W₂,…,wn |c;)P(c;)/P(w₁,W₂, … , wn), 从这个式子可以看出，计算P(c|d5) 和 P(j|d5) 时，分母是一样的，只需比较分 子即可。

P(c |d5)∞P(Chinese |c)\*P(Chinese |c)\*P(Chinese |c)\*P(Tokyo c)\* P(Japan |c)\*P(c)=(3/7)³\*1/14×1/14\*3/4=0.0003,P(j|d5)αP(Chinese l



①<https://web.stanford.edu/class/cs124/lec/naivebayes.pdf.>

j)\*P(Chineselj)\*P(Chinese lj)\*P(Tokyo |j)\*P(Japan j)\*P(j)=(2/9)³\* 2/9\*2/9\*1/4=0.0001,0.0003>0.0001,于是文档5的类别更有可能是c(China)。

文本分类的一个需要注意的问题是特征选择。在机器学习以及分本分类中，特征选择 对于分类的效果起到至关重要的作用。特征选择是一种降维方法。由于文本数据的半结构 化，甚至无结构化的特点，当使用特征向量对文档进行表示时，特征向量通常会达到几万 维甚至几十万维。通过特征选择，降低特征空间的维数，提高分类的效率和精度。

可以选用的特征选择算法有互信息、文档频率、信息增益、卡方检验等。这些特征选 择算法的思想可以通过第5章“数据的深度分析”具体了解。

**7.2.4** **文本聚类**

文档聚类 (Text Clustering)① 是把相似度大的文档放在同一类簇(为了和分类任务 (Classification) 中的类别 (Category) 分开，在这里 Cluster 称为类簇)中，把相似度小 的文档放在另外的类簇中，它是一种无监督的机器学习方法。也就是，文档聚类无须用户 预先对文档进行类别标注，无须训练过程，聚类方法自动把文档归拢到不同的类簇中。文 档聚类应用于需要对文本信息进行有效组织 (Organization) 、 浏 览 (Browsing) 和摘要 (Summarization) 的场合。

为了对文档进行聚类，首先需要对各个文档进行表示，最流行的表示法是利用向量空 间模型对文档进行表示，每个文档表示成一个稀疏的向量，各个元素对应各个词项的权 重，这个权重可以使用TF-IDF 公式进行计算。

除此之外，还可以利用各种矩阵因子分解方法 (Matrix Factorization), 包 括 LSI (Latent Semantic Indexing)② 方法和 NMF(Non Negative Matrix Factorization)③ 方法 对文档进行表示转换。LSI 和 NMF 方法都是降维方法 (Dimensionality Reduction), 它们 把文档从原有的高维空间(每个维度对应一个词项)转换到一个新的特征空间 (Feature Space), 维度小很多。在新的空间里的特征 (Feature), 一般是原有空间的若干特征的一 个线性组合 (Linear Combination) 。LSI 和 NMF 凸显了文档的语义，减少了相似度计算 时 (Similarity Measure) 的噪声，有利于提高文档聚类的效果。人们研究发现，矩阵因子 分解技术和谱聚类 (Spectral Clustering) 技术是等价的 (Equivalent), 谱聚类基于图结 构 (Graph Structure) 对文档进行聚类。

在上述各种文档表示法基础上，可以利用如下三种类别的聚类技术对文档进行聚类。

**1. 基于距离的聚类算法** **(Distance-Based Clustering)**

基于距离的聚类算法，通过一个相似度函数来度量不同文档之间的相似性。在文本分 析领域，最常用的相似度函数是向量的余弦相似度函数。该类算法又分为凝聚层次聚类算 法 (Agglomerative and Hierarchical Clustering Algorithm) 和基于距离的分区算法 (Dis- tance-Based Partitioning Algorithm) 两类，后者包括 K-Medoid 聚类算法、K-Means 聚类 算法等。

其中，凝聚层次聚类算法的基本思想是开始把各个文档都看作独立的类簇，然后通过

①在下文中，文本聚类和文档聚类互换使用 ·表达同一个意思。

②潜在语义索引 (LSI), 即潜在语义分析 (LSA)。

③非负矩阵因子分解 (NMF) 的基本原理，请参考相关资料。

多次迭代，每一轮迭代把最相似的文档类簇合并成大的类簇，直到最后所有的文档形成一 个类簇。在进行文档类簇的相似度度量时，可以采用Single Linkage Clustering,Group- Average Linkage Clustering,Complete Linkage Clustering 等度量方法。 Single Linkage Clustering 的两个文档类簇的相似度是来自两个类簇的任意一对文档的相似度的最大者。 Group-Average Linkage Clustering的两个文档类簇的相似度是来自两个类簇的所有文档对 的相似度的平均值。Complete Linkage Clustering的两个文档类簇的相似度是来自两个类 簇的任意一对文档的相似度的最小者。

**2. 基于单词或者短语的聚类算法** **(Word and Phrase-Based Clustering**)

假设一个文档集有 n 个词项和d 个文档，我们可以用一个 n\*d 的词项-文档矩阵 (Term-Document Matrix) 对文档集进行建模，矩阵的<i,j> 元素的值对应词项i 在文档 j 中的权重。

对矩阵的行进行聚类分析，就是对词项进行聚类分析，对矩阵的列进行聚类分析，则 是对文档进行聚类。这两个问题是紧密关联的，寻找文档聚类和寻找词项聚类，本质上是 对偶问题 (Dual Problem)。好的词项聚类结果可以帮助寻找好的文档聚类，好的文档聚 类结果可以帮助寻找好的词项聚类。

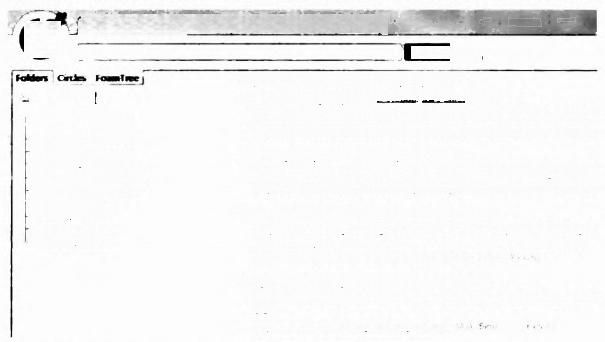
**3. 基于话题建模技术的概率聚类算法** **(Topic Modeling based Probabilistic Document Clustering)**

话题建模 (Topic Modeling) 是进行聚类和降维操作的更为通用的技术。人们利用LDA 模 型和PILSA模型等，对文档集进行话题建模，然后把每个话题 (Topic) 看作一个类簇 (Clus- ter)。话题建模的具体技术包括LDA 模型和PISA 模型，请参考7.2.6“主题抽取”一节。

话题建模的基本策略是为文档集建立一个概率生成模型 (Probabilistic Generative Model) 。在这个模型里，把文档集表示为隐藏随机变量 (Hidden Random Variable) 的 函 数 (Function)。从特定的文档集估计函数的参数，从而确定该模型。

传统的搜索引擎把所有的结果按照和查询的相似度不同，从高到低排序，返回给用 户，不同类别的信息混搅在一块儿。对搜索引擎返回的结果，通过话题建模进行聚类，可 以帮助用户快速定位到他们所需的信息。

通过话题建模对文档集进行聚类，以及对每个类簇给出一个简要的描述性标题(体现 了该类文档的主题),用户只需关心部分主题，于是可以有效地缩小检索范围。图7-10 是Carrot 2 搜索引擎对关键字查询 “Hadoop MapReduce” 返回的结果的聚类效果。



mru

Searth

Top 91realts of about 91for hdoop mupreduce

1 What sHdaop aplda2Webopedu Defintion

Hadoop Mapladuc (Hadoop Map/Reduce)s a softwore tramework for dstibu

htp/mww webopedacom/TERMHhadoop.maproduce htmd jA Bre

2 tadoop·MapReduce·Tutonalspont 4

Hadoop Mapraducn-Leam Hadoopin simple and eay steps starbng from itso

[http//mww.tutorialspoint com/hadoop/hadoop\_mapreduce htm tngC](http//mww.tutorialspointcom/hadoop/hadoop_mapreducehtmtngC)

3 MapRnduce ·Wikapedia.the free engydopeda: 4

Mapfeduce s aprogramming model and an assocated implementation for proe Hadoop's MpRoducr and RDEMS approeches on several specifk problems

[https//en wkapedia orywiMapleduxce .Eing G](https//enwkapediaorywiMapleduxce.EingG)

4 Apche Hdooo Moleduce-Hontonwors: 4

Apache tadoop M dce A framwort for writing applications that procesl thal.

htp//hortonworkscom/apache/muprduce/

Hdoopdud 图 Map and Radcn⑨ Sylorns B)

E Hadoog Tytoriw ⑥ 七 Coud Compuiog (5 E Hadoop Raiass(4)

L Leaning(4)

Wkipedis(4)

Aoyis3

nors|show all

elodws

hadoop mupreduce

Al Iops191

E Bg D (13)

C

w1

**图7-10** **Carrot 2搜索引擎的搜索结果聚类(基于话题建模)**



7.2.5 文 档 摘 要

文档摘要 (Document Summarization) 是为文档抽取或者生成一个简洁的版本。 Ra- dev 等把文档摘要定义为从一个文档或者多个文档生成的一个文档，这个文档能够表达 原有文档的重要信息，并且长度比原有文档的一半要小， 一般来讲要小很多。在互联网 信息时代，人们面对大量文本信息，文档摘要可以从一篇或者多篇文本生成一个简洁的 版本，帮助人们迅速把握文本里包含的主要信息。由此可见，文档摘要具有重要的应用 价值。

文档摘要的生成方法，按照使用的技术策略的不同，可以分为抽取法 (Extractive Summarization) 和摘要法 (Abstractive Summarization)。抽取法从原文本中抽取部分文 本(句子或者短语等)形成摘要。摘要法的目的是生成一个符合一 定语法规则的摘要 (Grammatical Summary), 一般需要用到一些语言生成技术 (Language Generation Tech- nique)。

按照原文档的数量，文档摘要也可以分为单文档摘要和多文档摘要两个类别。本章从 这个角度介绍几个具体的文档摘要方法。

1. 单文档摘要生成

Luhn 在1958年提出了如下单文档摘要方法，他的论文成为一篇经典论文，经常被引 用。首先，对文档里的单词进行词干提取 (Stemming), 还原到其词根形式 (Root Form), 同时把停止词 (Stop Word), 比 如a,the,this,that,and 等剔除掉。接着，把 文档里的单词按照其出现频率进行排序，给出各个单词的重要性 (Significance) 排名。然 后对每个句子计算一个重要性因子(Significance Factor), 反映各个重要单词 (Significant Word) 出现的次数以及它们的线性距离，也就是中间隔着多少非重要单词 (Non-Signifi- cant Word)。最后，对所有的句子根据重要性因子进行排序。把若干排在前面的句子选出 来构成文档摘要。

20世纪90年代以来，随着机器学习技术的发展，这些技术不断应用到自然语言处理 领域，包括文档摘要的生成。这些技术包括朴素贝叶斯方法、基于隐马尔可夫模型 (Hid- den Markov Model) 的方法、基于对数线性模型 (Loglinear Model) 的方法以及利用第三 方特征 (Third Party Feature. 比如搜索引擎查询中常用的一些查询关键字等)的神经网 络方法等。其中，基于朴素贝叶斯方法的文档摘要技术，简单介绍如下。

基于朴素贝叶斯方法的文档摘要技术，利用一个朴素贝叶斯 (Naive Bayes) 分类器， 把每个句子划分到两个类别——值得抽取/不值得抽取。

假设s 是某个特定的句子，S 是构成摘要的句子集合， F₁,F₂,…,F 是分类特征，

各个特征之间是互相独立的。



Kupiec 等 在Edmundson 等工作的基础上，提出了上述分类器。这个分类器使用的特 征包括高频词 (High Frequency Word)、线索词 (Cue Word 或 者Pragmatic Word)、出

现在标题中的词 (Title or Heading Word)、结构性指标(比如句子的位置 (Sentence Lo- cation)) 、 句子的长度 (Sentence Length)、是否出现大写的单词 (Upper Case Word) 等。根据上述公式，每个句子得到一个得分，最后保留得分最高的n 个句子构成文档 摘 要 。

**2.** **多文档摘要生成**

自20世纪90年代以来，从多个文档生成摘要的研究引起了人们的兴趣。大多数的研 究是面向新闻类文章 (News Article) 的多文档摘要生成的。不同的文档，其信息可能互 相补充、互相重叠(冗余),有时候还互相矛盾 (Contradictory) 。 多文档摘要生成，不仅 需要处理文档间的冗余问题，而且需要标定不同文档的新颖之处 (Novelty), 保证生成的 摘要是完整的和一致的 (Coherent) 。 主要的多文档摘要生成方法介绍如下。

(1)摘要和信息融合 (Abstraction and Information Fusion) 。McKeown,Barzilay 等 提出了此类文档摘要系统。在该系统里，首先标定主题 (Theme), 也就是标定相似的文 本单元(通常是段落)。这个问题可以建模为一个聚类问题。为了计算文本单元的相似度 指标 (Similarity Measure), 文本单元映射到一个特征向量，包括每个单词的 TF-IDF 得 分、名词短语 (Noun Phrase)、专有名词 (Proper Noun)、从 WordNet 数据库查询出来 的同义词 (Synset) 等。从数据中学习的决策规则 (Decision Rule) 确定每对文本单元是 相似的还是不相似的。聚类算法把最相似的文本单元放在一个主题里 (Theme)。

一旦确定了主题，系统进入第二个阶段，即信息融合 (Information Fusion), 其目的 是确定每个主题的哪个句子应该包含在摘要中。系统不是简单地从每个类簇 (Cluster) 里 选择一个句子作为代表，而是使用了更加复杂的算法。该算法首先把句子转换成依赖树 (Dependency Tree), 然后遍历这些 (Traverse) 依赖树的谓词-论元结构 (Predicate Ar- gument Structure), 考虑转述 (Paraphrase) 的冗余性，确定包含到摘要中的完整的一个 论述，即围绕某个动词展开的“主语-谓语(动词)-宾语”结构。

(2)基于最大边际相关性 (MMR) 的主题驱动的摘要生成 (Topic Driven Summari- zation) 。Carbonell 和Goldstein 于1998年提出了基于最大边际相关性 (Maximal Marginal Relevance,MMR) 的主题驱动的摘要生成技术 (Topic Driven Summarization)。该技术 的基本原理如下。假设Q 是一个查询 (Query) 或者用户画像 (User Profile),R 是从搜 索引擎返回的排序的文档列表(Ranked Document List)。现在构造一个增量式的 (Incre- mental) 选择文档的算法，每次选择一个文档到文档集 S 中。在每一步，对于R-S 文档集 的每个候选文档，D∈R-S,MR(D₁) 使用公式 MR(D₁)=λ sim(D,Q)-(1—λ)D₁es sim₂(D,D,) 进行计算，其中λ是一个取值在[0,1]之间的参数，在相关性 (Rele- vance) 和冗余度 (Redundancy) 之间做折中。siml 和 sim2 是两个相似度，通过向量空 间的余弦相似度公式进行计算(请参考信息检索一节)。R-S 集合中取得最大MR 的 D, 被 选出来，作为下一个进入S 的文档。

为了进行文档摘要，首先把这些文档分解成一系列的句子(或者段落)。用户提交一 个查询，对返回的排序靠前 (Top Ranking) 的段落运行MMR 算法。选择部分段落构成 文档摘要。由于该算法依赖于一个查询 (Query), 使得该算法特别适合为某个用户画像 (描述了用户偏好),生成个性化的文档摘要。

(3)基于图的激活扩散技术 (Graph Spreading Activation) 的摘要生成。 Mani 和 Bloedorn 于1997年提出该方法。这个方法并未生成一个文本性的摘要，而是把原来的文 档(句子和段落)表示成图，然后把图的重要区域 (Region) 通过图的激活扩散技术标定

**出来。**

该算法的基本流程是，首先，把文档表示成一个图，图的节点对应文档里出现的一个 单词或者短语(命名实体)。节点之间有若干连接类型，包括邻近连接 (ADJacency Link), 表示文本里单词间的相邻关系：相同连接 (SAME Link), 表示相同单词在不同地 方的出现 (Occurrence);ALPHA 连接 (ALPHA Link), 表示通过 WordNet 或者 Net Owl 捕抓到的语义关系 (Semantic Relationship),Net(wl是一个从结构化和非结构化大 数据中进行命名实体抽取和分析的软件包；短语连接 (PHRASE Link), 表示把属于同一 个短语 (Phrase) 的各个节点连接起来；名字和指代连接 (NAME and COREF Link), 表 示多次命名实体出现的指代关系 (Co-Referential)。

这个图建立好以后，通过 TF-IDF 得分，初始化突出的(得分高的)单词和短语，作 为人口节点 (Entry Node),也称为主题节点 (Topic Node)。然后使用激活扩散技术 (Spreading Activation),对语义相关文本的查找 (Search) 从这些节点开始，传播到其他 节点。根据如下原则，给相邻节点赋予权重，即在句子里的移动 (Travelling) 比在句子 之间更为容易，在句子之间移动则比在段落之间移动更为容易等。

对于两个文档对应的图来讲，公共节点 (Common Node) 指的是具有相同的词根 (Stem), 或者是同义词 (Synonym), 非公共节点称为相异节点 (Different Node)。对 于 两个文档的每个句子计算两个得分，分别是这个句子出现的公共节点的平均权重和这个句 子出现的相异节点的平均权重。最后，那些具有更高的公共节点得分和更高相异节点得分 的句子被突出强调 (Highlighted) 出来。用户可以指定两种类型的句子的最大数量，控制 输出结果。

(4)基于质心的文档摘要方法 (Centroid-Based Summarization) 。Radev 于2004年提 出了该方法，该方法是领域无关的 (Domain Independent), 也就是适用于所有领域。

首先，所有的文档以词袋模型 (Bag of Words) 进行表示。算法的第一阶段 (First Stage) 是主题检测 (Topic Detection), 目的是把描述同一事件的新闻文章 (News Arti- cle) 组织在一块儿。该系统在文档的 TF-IDF 向量表示基础上，使用凝聚层次聚类算法 (Agglomerative Clustering Algorithm) 实现主题检测。 一个文档类簇 (Cluster) 一般包 含2～10篇新闻文章，按照时间顺序排序，描述一个事件的发展历程。

第二个阶段使用每个类簇的质心，来标定类簇里关键(与整个类簇的主题紧密关联) 的句子。对于每个句子计算三个特征值，分别是句子的质心值 (Centroid Value,C₃)、位 置值 (Position Value,P₃)、与第一个语句的重叠值 (The First Sentence (verlap,F₃)。 C,是句子里每个单词的质心值的总和。P 的引入使得一个段落里领先的句子更加重要。F. 定义为这个语句的向量 (Word Occurrence) 和第一个句子的向量的内积。每个句子的得 分是上述得分的组合减去一个冗余惩罚因子 (Redundancy Penalty,R,), 描述其与排名较 高的句子的重叠度。

表7-3展示了某个类簇的质心。 Count 列表是每个单词在整个类簇的各个文档中出现 的平均次数。 IDF 从整个文集中进行计算。质心在这里是一个伪文档 (pseudo-docu- ment), 由 Count\*IDF 得分超过一定阈值的单词(在构成该类簇的文档中)组成。由此， 某个句子的Ci 值，即质心值，根据该句子的各个单词，在这张表中的Count\*IDF 的取值 进行累加得到。

**表7-3** **质心实例(Sample** **Centroid)**

|  |  |  |  |
| --- | --- | --- | --- |
| Word | Count | IDF | Count\*IDF |
| belgium | 15.50 | 4.96 | 76.86 |
| gia | 7.50 | 8.39 | 62.90 |
| algerian | 6.00 | 6.36 | 38.15 |
| hayat | 3.00 | 8.90 | 26.69 |
| algeria | 4.50 | 5.63 | 25.32 |
| islamic | 6.00 | 4.13 | 24.76 |
| melouk | 2.00 | 10.00 | 19.99 |
| arabic | 3.00 | 5.99 | 17.97 |
| battalion | 2.50 | 7.16 | 17.91 |

7.2.6 主 题 抽 取 (Topic Theme Extraction)

文档的主题是文档所描述的事物、概念等。比如一篇文档是关于美国大选的，另外 一篇文档是关于欧洲冠军杯的，我们可以把这两篇文档归入政治和体育两个大的主 题中。

在文档建模的主题模型中， 一般来讲，可以把一个主题看作一个桶。这个桶里面装 载了出现概率比较高的单词，这些单词和该主题具有很强的相关性。比如，关于2016 年美国大选的政治主题类文章中，“选票”“希拉里”“特朗普”等单词和短语出现的概 率很高。

对于主题和文档之间的关系，我们可以通过生成模型 (Generative Model) 来进行建 模。生成模型就是文章中出现某个单词，是通过这样的过程得到的：第一步，以一定概率 选择某个主题；第二步，从这个主题出发，以一定概率选择某个词语。

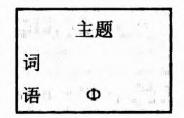
按照这个模型，当我们要生成一篇文档，文档里每个单词出现的概率为：



用矩阵来表达，其形式为：

二

|  |
| --- |
| 文档 词  语 C |



×



其中，“文档-词语”矩阵表示每个文档中每个单词的词频，即出现的概率。“主题-词 语”矩阵表示每个主题中每个单词出现的概率。“文档-主题”矩阵表示每个文档中每个主 题出现的概率。给定一个文档集，通过计算各个文档中每个单词的词频，就可以得到“文 档-词语”矩阵C 。主题模型就是通过左边这个矩阵进行训练，学习出右边两个矩阵。

在文本分析中对主题进行建模 (Topic Modeling) 的主要技术有概率潜在语义分析模 型 (Probabilistic Latent Semantic Analysis,PLSA) 和潜在狄利克雷分配模型 (Latent Dirichlet Allocation,LDA)等，下面分别给予介绍。

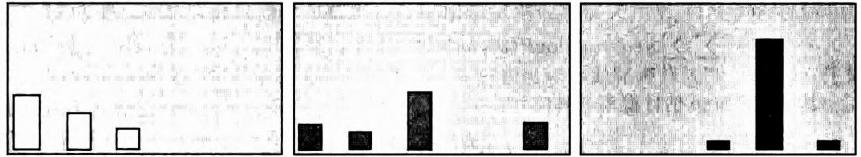


**1.PLSA 模型**

Hofmann 在1999年的信息检索国际会议 (SIGIR) 上提出了基于概率统计的PLSA

模型，并且用EM 算法学习模型参数。在实际应用中， 一篇文章往往有多个主题，只是各 个主题在文档中出现的概率不一样。比如，介绍一个国家的文档中，往往会分别从政治、 经济、文化等多个主题进行介绍。 Hoffmann 认为一篇文档 (Document) 可以由多个主题 (Topic) 混合而成，每个Topic 都是词汇上的概率分布，文章中的每个词都是由一个固定 的 Topic 生成的。图7- 11是英语中几个Topic 在词汇表 (Vocabulary) 的词汇上的概率 分布。

Math Computer Science Wine



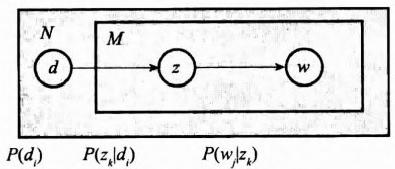
Proofinduction/objectbouquet/memory Proof/induction/objectbouquet/memory Proof/induction/objectbouquetmemory

(柱状表示每个单词在各个主题中出现的概率)

**图7-** **11** **各个Topic 在词汇表的词汇上的概率分布**

PLSA 是从概率分布的角度建模的一种方法，它假设在词和文档之间有一层语义隐藏 层，即主题，主题符合多项分布。 一个主题中的词项也符合多项分布。由这两层分布的模 型生成各种文档。在主题模型中，每篇文档d 都可以以概率P(z|d) 属于某个主题。在给 定主题以后，每个词以一定的概率P(w|z) 生成。

PLSA 的概率图如图7- 12所示。



**图7-12** **PLSA模型的概率图**

我们看到，在这个主题 (Topic) 模型中，Document 不 和 Term 直接关联。 Document 是有 Topic 的，通过Topic 和 Term 关联，但由于Topic 是不可观测的变量，它是一个隐 藏变量，Document 和 Term 都可以看成该隐藏变量的分布。

假设d 代表文档，z 代表隐含类别或者主题， w 为观察到的单词。 P(w; |di) 表示词 w; 在文档d; 中出现的概率， P(zk|d;) 表示某个主题zk在给定文档d; 下出现的概率，

P(w;|zk) 表示某个词w; 在给定主题zx下出现的概率，也就是，在P(w;l d)含有隐藏 变量。

生成模型具体如下：

(1)按照概率P(d) 选择一篇文档di。

(2)按照概率P(zk |d;) 选择一个隐含的主题类别zk。

(3)按照概率 P(w;|zk) 生成一个词w;。



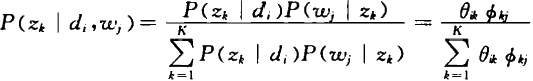
这样就可以得到文档中每个词的生成概率。换句话说，重复扔“文档-主题”色子和 “主题-词项”色子.重复M 次(产生M 个词),生成一篇文档。产生一篇文档的方法重复 N 次，则生成N 篇文档。

现在我们利用看到的文档，推断其隐藏的主题(分布)的过程，其实就是产生文档的 逆过程，即主题建模的目的就是自动地发现文档集中的主题分布。换句话说，P(zkld) 和 P(w;|zk) 分布对应了两组多项式 (Multinomial) 分布，我们需要估计这两组分布的参 数。文档d 和单词w 是可被观察到的，但主题z 是隐藏的。我们可以利用EM 算法得到 P(ze|d,),P(w; |zk) 的概率。

EM 算法的目标是Document 和 Term 的联合概率的似然函数最大化。

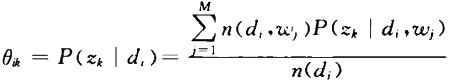


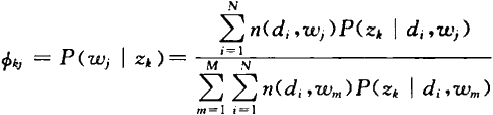
其中，E-Step 估计概率，具体为：



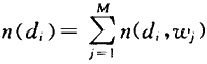
在这个步骤中，假设所有的 P(zk|d;) 和P(w;|zk) 都是已知的，因为初始时随机赋 值，后面迭代过程中，每轮能够从M 步骤得到参数值。

M-Step 更新参数，具体为：



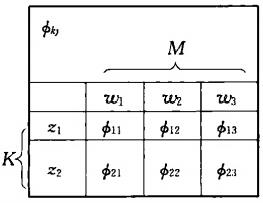
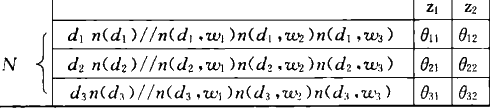


n(d,,w;) 表示词项w; 在文档d; 中的词频，n(d;) 表示文档d, 中词的总数，显然有



下面给出一个假想的实例，总共有3个文档，2个主题，词汇表有3个词。读者可以 基于如下表格，具体考察 P(z₁ ld₃,w₂),θ12,φ23 等是如何计算的。

|  |  |
| --- | --- |
| P(zxld,,u;) | |
| 21 | P(z₁ ldi,w₁)P(z₁ ldi,w2)P(z₁ ld,w3) |
| P(z₁ Id₂,wy)P(z₁ Id2,w₂)P(z₁ Id2,ws) |
| P(z₁ ld₃,u)P(zild₃,uz)P(z₁ Ids,ws) |
| Z2 | P(z2 |di,wy)P(z₂ |di.w₂)P(zz Idi,ws) |
| P(z2 |dz,w)P(z₂ l d2,u2)P(z₂ I d2,us) |
| P(z₂ |d₃+u)P(z₂ |d₃,02)P(z₂ |d₃,w₃) |



|  |  |  |  |
| --- | --- | --- | --- |
|  | B次  K | | |
|  |  |  |
|  |  |  |  |
|  |  |  |
|  |  |  |

2.LDA 模型

LDA 是一种非监督机器学习技术，可以用来识别大规模文档集 (Document Collec- tion) 或语料库 (Corpus) 中潜藏的主题信息。

LDA 是一种文档主题生成模型。文档的生成过程可以形象化地描述如下：(1)生成 器有两个装满色子的坛子，第一个坛子装的是Doc-Topic 色子，第二个坛子装的是 Topic- Word 色子。(2)生成器随机从第二个坛子中，独立地抽取 K 个 Topic-Word 色子，编号 为1～K 。(3) 每次生成一篇新的文档时，生成器从第一个坛子随机抽取一个Doc-Topic 色子。然后，重复如下过程生成文档中的词： (a) 投掷这个 Doc-Topic 色子，得到一个 Topic 编号z 。(b) 选择 K 个 Topic-Word 色子中编号为z 的那个色子，投掷该色子，得到 一个词。

具体来讲，LDA 模型是一个三层贝叶斯概率模型，包含单词、主题和文档三层结构， 如图7 - 13所示。在图7 - 13表示的 LDA 模型中，定义了如下生成过程 (Generative Process)。

(1)对于语料库中的每篇文档，首先选定一个主题向量 θm。整个θ是一个M\*K 的矩 阵，即M 个 Doc, 每个 Doc 对应一个 K 维Topic 分布向量。

(2)重复如下过程，直到遍历文档中的每一个单词。

(a) 从主题分布向量θ中选择一个主题z。

(b) 按主题z 的单词概率分布生成一个单词。



模型中的α参数为K 维向量；P(θmlα) 为选择某个文档概率分布的概率；β参数为各个

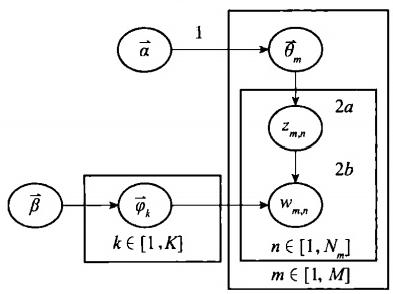


图7-13 LDA 模型

主题对应的单词概率分布矩阵。

换个角度，上述概率图(见图7-13)可以分解为两个主要的物理过程：

(1)a→0m→zm,n, 这个过程表示在生成第 m 篇文档时.先从第一个坛子中抽一个 Doc-Topic 色子0m, 然后投掷这个色子生成了文档中第n 个词的Topic 编号 zm,n。

(2)β→→wm,nIk=zm 、n, 这个过程表示生成语料中第 m 篇文档的第n 个词，过程 是从K 个 Topic-Word 色 子 中 挑 选 编 号 为k=≈m,n 的那个色子进行投掷，然后生成 Word wm,n。

LDA 的学习过程是利用给定的语料库，训练出两个控制参数a 和β。具体是把w 当 作观察变量，θ和z 当作隐藏变量，通过 EM 算法，学习出a 和β(具体过程请参考相关资 料)。LDA 以文档集 D 作为输入，首先进行分词、去掉停用词 (Stop Word)、提取词干 等处理，然后训练出如下两个结果向量。假设文档集聚类成K 个 Topic,D 中所有单词 构成的词典包含 N 个词。

(1)对每个 D 中的文档，对应到不同 Topic的概率分布， θa=<P₁,…,p₄>, 其 中 ，p, 表示 d 对应 T 中第i 个 Topic 的概率。(2)对于每个T 中的Topict, 对应生成不 同单词的概率φ,=<Pu₁, … ,Puy>, 其中，pu, 表示 t 生成 Vocabulary 中第 i 个单词的 概率。

比如，我们有如下几个小的文档①,

Docl:I like to eat broccoli and bananas.

Doc2:I ate a banana and spinach smoothie for breakfast.

Doc3:Chinchillas and kittens are cute.

Doc4:My sister adopted a kitten yesterday.

Doc5:Look at this cute hamster munching on a piece of broccoli.

LDA 算法可以发现这些文档中的主题。如果我们限定主题数量为2个，那么LDA 产 生如下结果。

(1)各个文档的主题分布是：

Doc 1:100%Topic A。

**Doc 2:100%Topic A。**

Doc 3:100%Topic B。

Doc 4:100%Topic B。

Doc 5:60%Topic A,40%Topic B。

(2)各个主题的不同单词的概率分布是：

Topic A:30%broccoli,15%bananas,10%breakfast,10%munching, …。

Topic B:20%chinchillas,20%kittens,20%cute,15%hamster, …。

由此，我们可以把主题A 解释为“食物”主题，主题B 解释为“可爱的动物”主题。

**David Blei 提出的LDA 生成模型** **(Generative Model) 看起来相当简单，但它的建模** 能力**是非常强的。LDA** **算法获得的结果非常直观，它产生的文档主题和我们感受到的文** 档主题是如此相似。



①<http://blog.echen.me/2011/08/22/introduction-to-latent-dirichlet-allocation/.>



**7.2.7** **命名实体识别、概念抽取和关系抽取、事实抽取**

**1.** **命名实体识别** **(Named** **Entity** **Recognition,NER)**

命名实体(简称实体)识别的目的是发现文档里的各种实体。这些实体包括人物、地 理位置、组织、日期时间、数字(比例、金额、重量等)、邮件以及各种特定领域的实体， 比如药品、疾病、河流、山川等。

命名实体识别根据使用的技术分为三种类型，分别是基于正则表达式的命名实体识 别、基于字典的命名实体识别，以及基于机器学习模型的命名实体识别。(1)基于正则表 达式的命名实体识别，把预先定义的正则表达式和文本进行匹配，把符合正则表达式的文 本模式都定位出来。基于正则表达式的命名实体识别一般用于日期时间、金额、电子邮件 等规则的文本。(2)基于字典的命名实体识别、把文本和字典里的<短语，类别>对进行 匹配，对匹配的短语进行实体标注， 一般用于人名、地名的识别。(3)基于机器学习模型 的命名实体识别，则需要预先对一部分文档进行实体标注，即产生一系列的<短语，类 别>,然后利用这些文档进行机器学习模型的训练，用这个模型对没有遇到过的文档进行 命名实体识别和标注。

用于命名实体识别的条件随机场 (Conditional Random Field) 技术属于第三类技术。 条件随机场由 Lafferty 于2001年提出，他在最大熵模型和隐马尔可夫模型的基础上提出 了新的概率无向图学习模型。该模型是用于标注和切分有序数据的条件概率模型，目前已 经成功应用于自然语言处理 (NLP) 、 生物信息处理、机器视觉等领域。

首先，随机场是一组随机变量的集合，一般这些变量之间有依赖关系。马尔可夫随机 场 (Markov Random Field.MRF) 是一类特殊的随机场，它对应一个无向图。在这个无 向图上，每个节点对应一个随机变量；节点之间的边表示随机变量之间的概率依赖关系。 马尔可夫随机场反映了人们的先验知识，即哪些变量之间有依赖关系。在马尔可夫随机场 中。随机变量的概率分布只和它的邻居节点有关而与其他节点无关，体现了马尔可夫 性 质 。

如果给定的马尔可夫随机场中每个随机变量对应一个观察值.我们需要确定在给定观 察值集合下，马尔可夫随机场的分布。这个马尔可夫随机场 (MRF) 的分布是一个条件 分布，那么这个马尔可夫随机场称为条件随机场 (CRF), 或者说CRF 本质上是给定了观 察值 (Observation) 集合的MRF。①

使用条件随机场来识别实体，需要：(1)选择特征函数(也就是无向图上的各个节点 对应的随机变量的分布函数);(2)利用标注好的训练数据，估计条件随机场模型的参数， 也就是各个特征函数的权重向量；(3)利用已经学习的模型，对新数据进行标注。

**2.** **指代消解**

指代消解 (Co-Reference Resolution) 是自然语言处理中和命名实体识别关联的一个 重要问题。比如下面这段话，“比尔 ·盖茨是微软公司的首任总裁，他于20世纪70年代 就提出让每个家庭都拥有一台计算机的口号”,“他”表示比尔 ·盖茨。又比如，在对某位 专家学者的一个访谈中，除了第一次提到其姓名职务之外，之后提到这位专家，文本中可 能使用“某博士”“某教授”“他”等代称，或者以其所担任的职务相称，比如“所长”等

① 关 于MRF 和CRF 的更加详细的信息，请参考相关资料。



代称。如果访谈中还提及其他人物，并且使用了类似的代称，那么把这些代称对应到正确 的命名实体上，就是指代消解。

指代消解的应用包括信息抽取、信息检索、文本摘要生成、机器翻译、问答系统等。 常见的指代消解根据照应词 (Anaphor) 的词性，分为代词消解(比如上述实例中的 “他”指的是该专家)和名词消解(比如上述实例中的“所长”指的是该专家)。

人们尝试多种方法实现指代消解，包括：(1)基于语法的指代消解，是较早使用的方 法。这种方法以启发式的方式利用语法层面的知识，实现指代消解。(2)基于大量语料库 的统计方法、机器学习方法(比如聚类、决策树等),已经有研究人员开始使用深度学习 方法，实现代词的指代消解。

**3. 命名实体歧义消除** **(Named Entity Disambiguation)**

在自然语言处理中，经常遇到的一个问题是命名实体的歧义问题。比如人名的重名问 题，全国叫王涛的人有很多，有时候重名问题甚至在一个单位出现。为了让计算机能够正 确分析自然语言书写的文本，对命名实体的歧义需要进行消除，也就是把具有歧义的命名 实体唯一地标识出来。

人们研究了许多方法进行命名实体的歧义消除。近年来，随着Freebase 等语义知识库 的出现，大量基于知识库的命名实体歧义消除方法被提出来。命名实体歧义消除的任务是 将文本中出现的每个实体指称去除歧义，唯一地链接到知识库的唯一实体。可用的知识 库，除了Freebase, 还有 Wikipedia.DBpedia 等。有的研究者从Wikipedia 抽取命名实体 的上下文信息，并且从文档中获得命名实体的上下文信息，然后进行比对，找出匹配度最 高的Wikipedia 页面，从而对命名实体完成歧义消除。

**4. 关系抽取** **(Relation Extraction)**

从上文的介绍我们知道信息抽取是将结构化或者半结构化形式的自然语言文本，转化 成结构化数据。关系抽取是信息抽取的一个重要的子任务，它负责从文本中识别出实体 (Entity) 之间的语义关系 (Relation)。

比如，从“郭富城、梁家辉、周润发三大影帝合作出演了电影《寒战2》”文本中，我 们可以抽取出几个重要的实体，以及实体间的关系，具体如表7-4所示。

**表7-4**

|  |  |  |
| --- | --- | --- |
| 实体1 | 实体2 | 关系 |
| 郭富城 | 梁家辉 | 联合出演(co-star) |
| 梁家辉 | 周润发 | 联合出演(co-star) |
| 郭富城 | 周润发 | 联合出演(co-star) |
| 郭富城 | 寒战2 | 出演(staring) |
| 梁家辉 | 寒战2 | 出演(staring) |
| 周润发 | 寒战2 | 出演(staring) |

关系抽取得到的结构化数据，可以用于自动问答系统、文档摘要、知识库建设与知识 推理等重要的应用场合。关系抽取的主要方法介绍如下。

(1)有监督的机器学习 (Supervised Learning)。有监督的机器学习方法主要分为两 类，包括基于特征向量 (Feature-Based) 的学习方法和基于核函数 (Kernel-Based) 的学 习方法。有监督的机器学习方法，把关系抽取任务当作分类问题解决。首先。需要人工标

注大量训练语料库，然后在已经标注好的语料库上进行特征抽取和选择，利用不同的机器 学习算法，训练分类模型，用于抽取新的实体对及其关系。这里讲的特征，包括实体对的 上下文的各种词法、语法、语义等信息以及背景知识。

(2)半监督的机器学习 (Semi-Supervised Learning)。半监督的机器学习方法，无须 人工标注语料库，但是需要根据预定义好的关系类型，人工构造出关系实例，作为种子集 **(Seed Set), 然后利用Web 或者大规模语料库信息的高度冗余性，充分挖掘关系描述模** **式，通过模式匹配，抽取新的实体关系实例。请参考第9章“语义网与知识图谱”中关于** **知识图谱构建的介绍。**

**(3)无监督的机器学习** **(Unsupervised Learning)。无监督的机器学习方法是一种自** **底向上的信息抽取策略。它基于这样的假设：拥有相同语义关系的实体对的上下文信息较** **为相似，其上下文集合代表该实体对的语义关系**。

关系抽取过程分为三个阶段，包括实体对及其上下文信息的提取，根据上下文信息对 实体对进行聚类，对各个类簇的语义关系进行标注，也就是对关系类型进行描述。这个方 法产生的聚类结果描述了比较宽泛的语义关系，定义合适的关系类别有些困难。另外，对 于低频的实体对及其关系，其处理能力有限。

目前，关系抽取技术并未做到尽善尽美、存在很多挑战、包括：(1)自动发现关系类 型。目前的研究工作一般都是基于定义好的关系类型抽取关系实体对。如何自动或者半自 动地建立一套合理的关系类型体系仍然有待解决。(2)关系推理与冲突消解。目前的研究 工作将每种关系类型作为独立的处理对象，并未考虑其潜在的关系。比如我们可以在父子 和母子关系基础上推断出夫妻关系，在父子和父子关系基础上推断出祖孙关系等。夫妻关 系有一对一的约束，朋友关系可以是一对多等。(3)领域自适应的关系抽取。目前的研究 工作主要面向特定的关系类型或者特定的领域，使用特定的语料库，很难在不同领域之间 进行迁移。跨领域的关系抽取需要更多的研究努力。(4)篇章级的关系抽取。目前的研究 工作主要以句子级实体间的关系为研究对象，丢失了大量的代词参与的关系。利用共指消 解处理结果、在篇章级别上实现关系抽取，实现对篇章的更好理解。

**5. 概念抽取** **(Concept Extraction) 与建立概念层次关系** **(Concept Hierarchy)**

概念层次关系是用以组织和展现((rganize and Present) 知识的有用方法。概念层次 关系也可以称为概念分类系统 (Concept Taxonomy)。

概念抽取任务包括各个概念的抽取，以及建立概念之间的层次关系。它把信息组织到 分类系统里，对数据里体现的关系进行抽象和泛化处理。概念抽取的结果成为一个领域本 体 (Ontology) 的主干内容，本体是一个主题下不同概念及其关系的总和。

为了抽取概念，人们研究了不同的方法，简单介绍如下。

(1)利用字典信息。字典里的条目具有一定的规律性 (Regularity) 。 比如，对于 tiger 的解释有 “a tiger is a mammal (哺乳动物)”。这句话指出老虎属于哺乳类动物，哺乳动 物是一个概念。Terml is a Term2 包含了Is-a 关系，指出了Terml 和 Term2 的上位词关 系 (Hypernym), 即 Term2 是上位词。利用字典信息抽取概念，准确度较高，缺点是和 领域相关 (Domain Independent)。

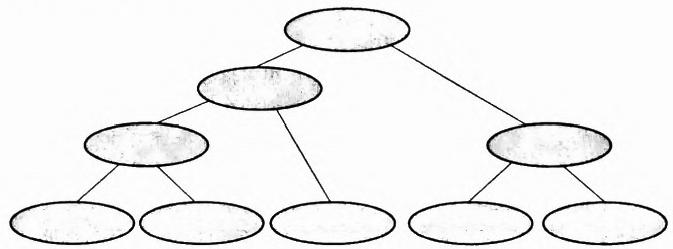
(2)使用词汇语法模式 (Lexico-Syntactic Pattern)。我们可以利用词汇语法模式，从 文档集里自动学习到上下位关系 (Hyponym Relation)。比如 “Such injuries as bruises, wounds and broken bones”包含了如下抽象概念到具体概念的上下位关系，即 Hyponym (bruise,injury),Hyponym(wound,injury),Hyponym(broken bone,injury) 等。

Hearst 总结了若干词汇语法模式，称为Hearst 模 式 (Hearst Pattern)。从文档里匹配这 些模式，就可以学习到上下位关系。

|  |
| --- |
| Hearst 1:NP such as {NP,}\*{(and |or)}NP  Hearst 2:such NP as {NP,}\*{(and |or)}NP  Hearst 3:NP{,NP}\*{,}or other NP  Hearst 4:NP{,NP}\*{,}and other NP  Hearst 5:NP including{NP,}\*NP{(and |or)}NP  Hearst 6:NP especially{NP,}\*{(and |or)}NP |

注：NP 为 Noun Phrase,即名词短语

(3)简单凝聚层次聚类。利用文档里各个名词或者名词短语的相似性，进行凝聚层次 聚类，如图7-14所示。在这里，名词或者名词短语间的语义相似度可以用其上下文的相 似度来度量 (Similarity of the Context)。



car **bike** apartment excursion **trip**

**图7-14** **概念以及概念上的凝聚层次聚类**

无监督的聚类算法产生的聚类树，由于数据的稀疏性，名词或者名词短语间的相似度 有可能是偶然发生的 (Accidental), 导致我们得到的概念之间的相似度是错误的。此外， 我们很难给各个类簇加上合适的标签 (Label), 于是聚类树缺乏清晰和正式的解释。

(4)指导下的聚类。利用从WordNet 获得的上下位关系 (Hypernym), 以及从文档 集里匹配的Hearst 模式、从WWW 上匹配的Hearst 模式所表现出来的上下位关系，指导 聚类过程。指导下的凝聚层次聚类 (Guided Agglomerative Clustering,GAC) 的基本过 程是：(a) 输入一个Term 列表；(b) 计算每对Term 之间的相似度，并且按照从高到低 进行排列；(c) 对于可能聚集到一个类簇中的每对Term, 从上述三个来源获得上下位关 系的提示，如图7-15所示。

GAC 抽取的结果相当好，它提供了每个类簇的标签，同时，它比单纯的凝聚层次聚 类要快，避免了数据稀疏性导致的虚假的相似性问题 (Spurious Similarity)。

(5)通过机器学习确定Term 之间的关系。比如要确定Terml 和 Term2 是否具有Is-a 关系，可以构造一个分类器，如图7-16所示。

分类器可以基于决策树 (Decision Tree)、朴素贝叶斯 (Naive Bayes) 网络、神经网 络等技术构建(多层感知机， Multi-Layer Perceptions)。为了训练分类器，需要各个 Term 的特征值，这些特征值包括：

(a) 给定t1 和t2, 计算文档集里能够匹配多少次Hearst 模式，这些Hearst 模式显示 t1 和t2 具有Is-a 关系。即isaHearst =I patterns(t1,t2)I/I patterns(t1,')|。比如，

**WordNet** Google

Hyponym

Hints

Clustering

Concept Hierarchy

图7- 15 指导下的凝聚层次聚类

Features

Is-a(T1,T2) Yes/No

Classifier

Corpus

Similarity Calculation



**图7-** **16** **用于判断是否具有** **Is-a关系的分类器**

我们在文档集里匹配到如下的 Hearst 模 式 ， 于 是 conference 是 一 个 event 的 可 能 性 比 较大。

|  |
| --- |
| isararst(conference,event)0.36  isararat(conference,body)0.18  isararst(conference,course)0.09  isarort(conference,weekend)0.09  isanorst(conference,meeting)0.09 isatarst(conference,activity)0.09 |

(b) 在 WWW 上匹配一些模式，这些模式是特定的表达式 (Expression), 可以使用 Google 搜索引擎 API 来获得特定模式的数量。具体的表达式具有如下形式：

|  |
| --- |
| π(t₁)such asπ(t₂)  suchπ(t₁)as π(t2)  π(t₁),includingπ(t₂)  π(t₁),especiallyπ(t₂)  π(t₁),and other π(t₂) π(t₁),or otherπ(t₂) |

然后利用公式 isawww=I patternswww(t1,t2)|/I patternswww(t1,")| 计算本特征。最后 的结果如下，表示conference 是 一 个event 的可能性较大。

|  |
| --- |
| isa(conference,event)0.27  isamm(conference,activity)0.17 |



isa (conference,initiative) 0.03 isamm(conference,function)0.03

(c)WordNet 特征。通过是否存在从t1 的 Synset 到 t2 的 Synset 的多条路径，来 判断t1和t2之间是否具有上下位关系(Hypernym Relationship)。具体的计算公式为isaws (t1,t2)=min(1,Ipaths(senses(t1),senses(t2》|/|senses(t1)| 。Senses(t1) 表 示t1 的各个sense (意思)的Synset。

(d) 一 个 Term 出现在另外一个Term 之前，也可以给我们一些关于Is-a 关系的启 发。比如， t1=“international conference”,t2=“conference”, 表示11和t2 之间存在 isahead(t1,t2) 的关系，也就是international conference是一种 conference。

(e) 基于文档集的包含关系 (Corpus-Based Subsumption)。t1 是12的 一 个子类 (Subclass), 如果整个文档集中出现t1 的所有的上下文都出现t2, 具体来讲，isacopus(t1, t2)=|features(t1)∩features(t2)|/|features(t1)|。

(f) 基于文档的包含关系 (Document-Based Subsumption) 。t1 是 t2 的 一 个子类 (Subclass), 如果t1 出现的文档都出现t2, 具体来讲，isadocumen(t1,t2)=|#doc both t1 and t2 occurs l/|#doc tl occurs|。

**6. 事实抽取** **(Fact Extraction)**

Web 包含了大量关于实体的事实性信息 (Factual Information),这些实体包括历史人 物 (People) 、 名人 (Celebrity) 、 地理位置 (Place) 、 电影、书籍、产品、重大事件 (E- vent) 等。如果有办法把这些事实性信息收集起来，并且提供搜索这些信息的机制，就可 以利用这些事实，提高问答系统 (Question Answering) 的答案质量，以及搜索引擎 (Search Engine) 的搜索结果质量。搜索引擎能够利用事实抽取结果， 一定程度上理解网 页的内容。

主流的搜索引擎包括Google,Bing,Yahoo 等，以前都是根据用户查询.返回一个网 页链接列表，每个链接旁边附带一个摘要。用户需要查看这些摘要，或者从链接导航到目 标网页，才能查找到答案。利用事实性信息，以及加深对用户查询意图的理解，搜索引擎 就可以针对用户的查询，返回简短的答案.这些答案正好是用户所期望的。我们在Google Q &.A,Product Search,Local Search (用商家名称 (Name) 、 关键字 (Key Word)、卫 星图片上的位置信息等搜索本地商业目录)等服务中，看到抽取出来的事实发挥了重要 作用。

关于实体的事实性信息，可以用<属性，值>对的形式来表示。比如对于名人安吉丽 娜 ·朱莉 (Angelina Jolie),“Birthday:June 4,1975” 和 “Birth Name:Angelina Jolie Voight” 是她的两个事实。关于实体Angelina Jolie的事实信息，如表7-5所示。

**表7-5** **关于实体Angelina** **Jolie的事实**

|  |  |
| --- | --- |
| 属性名(Attribute Name) | 属性值(Attribute Value) |
| Entity Name | Angelina Jolie |
| Birth Name | Angelina Jolie Voight |
| Date of Birth | June 4.1975 |
| Place of Birth | Los Angeles,California,United States |
| Academy Awards | Best Supporting Actress |

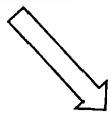
资料来源：所有的事实来源于<http://en.wikipedia.org/wiki/Angelina_Jolie.>



事实抽取试图确定和某个命名实体 (Named Entity) 相关的所有事实(即属性及其取 值)。由于抽取事实的来源一般是 Web 网页.所以抽取了事实以后，还需要把原网页的网 址记录下来。抽取的事实一般存放到数据库里.以便后续对其做规范化处理 (Normaliza- tion) 以及清洗 (Cleanup) 等。

我们通过Google 公司提出的GRAZER 系统，来了解事实抽取的基本过程。GRAZER 是从 Web 学习和证实事实 (Corroborate and Learn Facts from the Web) 的一套软件 系统。

首先，我们介绍该系统用到的一些重要概念。(1)实体：包含一系列的事实。(2)事 实：是关于某个实体的<属性，值>对，附带提及该事实的源网址 (URL) 。(3) 相关网 页：和某个实体相关的网页。(4) HTML 模式 (Pattern): 在一个网页上重复超过两次以 上的连续的 HTML 标记序列 (Contiguous HTML Tag Sequence)。

GRAZER 从半结构化的网页中 (Semi-Structured Text),基于已知的关于某个实体的 事实，抽取更多的关于该实体的事实，并且对其进行证实。如图7-17所示。

|  |  |
| --- | --- |
| Known Facts | |
| Entity Name | Angelina Jolie |
| Date of Birth | June 4,1975 |

|  |  |
| --- | --- |
| More Facts | |
| Entity Name | Angelina Jolie |
| Date of Birth | June 4,1975 |
| Academy Awards | ? |
| Place of Birth | ? |

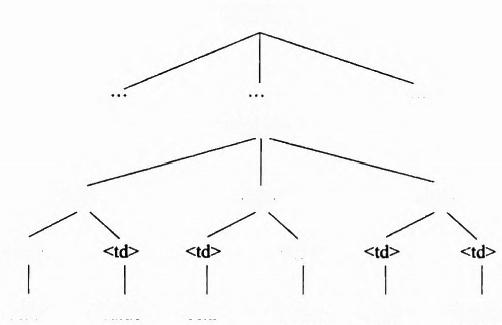
**图7-17** **GRAZER 事实抽取**

GRAZER 的工作原理，详细叙述如下：

(1)首先，系统为实体建立种子事实 (Seed Fact) 集合。在该系统里，种子事实集合 是通过手工编写的脚本从en.wikipedia.org 网站上爬取相关信息而创建的。这些网页可以 通过判定指向它的链接的锚文字 (Anchor Text) 是否和命名实体的名字 (Entity Name) 相匹配 (Match), 来确定是否和某个实体相关。还可以使用两个启发式规则保证网页的质 量：一个是实体名出现在网页的标题 (Page Title) 里；另一个是实体名出现在网页的突 出位置上，比如大标题 (Heading)。

(2)提取其他相关网页并分析。寻找提及目标实体以及种子事实 (Seed Fact) 的其他 相关网页和第一步里提取初始的种子事实集的网页，方法是类似的。当系统查找到一个新 的网页里提及已知的事实时，它开始在网页上已知事实及其周围区域 (Surrounding Area) 寻找重复的HTMI 模式。GRAZER 抽取符合 HTML 模式的所有事实(套用该HTML 模 式)。如果该网页包含一定数量的已知事实，GRAZER 用新的事实来丰富已知的事实集合 (Known Fact Set)。

人们在编写网页时， 一般把一个实体的事实性信息组织成一张表格的形式。图7-18 展示了这样的一个表格(这里显示网页的 HTML 标记)以及由其组织起来的关于某个实 体的事实。其中，表格的每一行对应一个 HTML 模 式 ( 即 HTML Tag序列),包含关于 实体的一个事实。



<html>

·

<table>

<tr>

<td>

Attr Value

Attr Value

Attr Value

<td>

<tr>

<tr>

**图7** **-** **18** **HTML** **表格与事实(<属性，值>对)**

利 用HTML 模 式 (Pattern) 来寻找新事实，它的假设是如果在 一个 HTML 模式里 包含了一个事实，附近类似的HTML 模式极有可能包含对同一个实体的另外一个事实的 描述。如图7 - 19所示，关于Angelina Jolie的奖项、出生地等事实，出现在出生日期 (包含在 一 个HTML 模式中)附近。





<tr>

<td>Date ofBirth</td><td>June 4,1975</td> </tr>

<tr>

<td>Academy Awards</td><td> … …</td> </tr>

<tr>

<td>Place of Birth</td><td> … …</td> </tr>

..

|  |  |
| --- | --- |
| More facts | |
| Entity name | Angelina Jolie |
| Date of birth | June 4,1975 |
| Academy awards | ? |
| Place of birth | ? |

**图7-** **19** **新事实就在已知事实的附近**

对于已经抽取的事实，如何评价其可靠性是个问题。 GRAZER 认为，经过证实的事 实， 一般有两个以上的来源，未经证实的事实，则只有一个来源。拥有多个来源的事实.

一般来讲是可靠的。还可以使用其他信号 (Signal) 来确定事实的质量 (Quality), 包 括 来源网页的可靠性 (Reliability) 以及多样性 (Diversity) 等 。

事实的抽取和证实的过程是一个自我改进的过程 (Bootstrapping Process), 经过少数 几轮学习就可以结束，并获得较好的抽取结果。扩大的事实集合，用在下一轮学习中，于 是已知事实集合不断增长，直到满足终止条件 (Stopping Criterion) 为止。

**7.2.8 情感分析** **(Sentiment Analysis)**

情感分析的应用很广泛，包括对问卷调查 (Survey)、产品的用户评论 (Review)、 新 闻 (News) 、 博客 (Blog) 、 论坛 (Forum) 、 呼叫中心日志记录 (Call Center Logs) 等进 行情感分析。

对社会媒体 (Social Media)的情感分析，可以实现互联网舆情监控。互联网舆情监 控使得政府、公益组织和企业及时分析海量的文本数据，了解舆论倾向，及时应对。互联 网使得广大网民可以自由地表达观点，逐渐成为话题产生和传播的重要渠道。由于互联网 信息量庞大，很难依靠人工方法及时进行数据收集和处理。依靠情感分析技术，可以自动 对舆情信息进行监控。有些研究者通过对微博内容进行情感分析，了解青少年的心理健康 状况。这些分析结果可以帮助社会工作者对过激的思想和行为倾向进行干预。在金融市场 上，有些研究者通过分析 Twitter 用户的情感变化来预测股票价格的涨跌。

情感分析包括情感信息抽取、情感信息分类，以及情感信息的检索及归纳三个层次的 任务。情感信息抽取的目的是抽取情感文本中有意义的信息单元，包括观点持有者、评价 对象、具体评价、情感评价单元(主要形式是<评价对象，评价词语>搭配)等信息。比 如“我认为M 品牌汽车的油耗真高”,“我”是观点持有者，评价对象是 “M 品牌汽车的 油耗”,具体评价是“真高”,情感评价单元是“油耗\_真高”。

情感信息分类按照分类粒度，可以分为篇章级情感分类、语句级情感分类以及短语级 (具体到某个对象以及对象的某个方面，比如照相机以及照相机的对焦速度)情感分类。 从分类的目的来划分，情感信息分类分为主/客观分析和正面/负面分析。

情感信息的检索是针对用户的查询请求，返回和某个主题相关的包含情感信息的文 档。情感信息的归纳则针对某个主题的大量情感文档，汇总出情感分析结果，同时基于时 间分片进行情感汇总，了解情感的变化趋势。

在这里我们不深入介绍情感信息抽取、检索和归纳而是重点介绍情感信息分类。情感 信息分类，首先要完成主/客观信息的分类，然后对主观信息进行细致的情感分类，包括 常见的褒贬二元分类，以及更加细致的类别划分。

比如，有三句话：“品牌M 汽车的油耗很高”“品牌J 汽车的性价比相当高”“品牌Z 汽车有1米多高”。第一句表达了贬抑的情感倾向，第二句表达了褒扬的情感倾向，第三 句则表述了一个事实，没有情感倾向。主/客观信息分类，把第一句、第二句分到主观信 息类，把第三句分到客观信息类。研究者使用了Naive Bayes 等分类器进行二元分类。为 了提高这些分类器的性能，需要对分类特征 (Feature) 进行选择，包括从标点符号、人 称代词、数字等角度，区分主/客观文本的异同。

在主/客观信息分类的基础上，对于主观信息还需要进一步进行具体的情感分类，主 要的策略有两种。第一种策略，依靠情感词典或者领域词典，对主观文本中带有情感极性 的组合评价单元进行评价，获得主观文本的情感极性(正面或者负面，即褒义或者贬义)。 需要注意的是，人工创建的情感词典，往往出现不能囊括文本中的所有情感词的情况，需 要人工或者自动的方法进行调整和扩充。第二种策略，则选取文本中对情感判别有意义的 特征 (Feature), 交给机器学习方法来完成分类。比如，有些研究者使用了n-Gram 词语特 征和词性特征 (n-Gram 是多个单词或者汉字的组合，比如 I am happy包含两个2-Gram, 分 别为I am 和 am happy). 然后使用Naive Bayes、最大熵模型 (Maximum Entropy Model)、

支持向量机 (Support Vector Machine,SVM) 等分类器来实现分类。基于机器学习的情 感极性分类，需要事先准备经过人工标注的训练语料。

最大熵原理是1957年由Edwin Thompson Jaynes 提出的。其主要思想是在只掌握关 于未知分布的部分知识时，应该选取符合这些知识但熵值最大的概率分布。基于该思想建 立的最大熵模型 (Maximum Entropy Model)① 是一种机器学习方法，在自然语言处理的 许多领域，包括句子边界识别、中文分词、词性标注、浅层句法分析、文本分类等应用 中，都取得较好的效果。

在这里举一个拼音转汉字的例子。②假如，输人的拼音是 “wang'xiao’bo”, 根据 语言模型以及有限的上下文信息，可以给出两个常见的名字“王小波”和“王晓波”,但 要进一步确定唯一的名字则有些困难。我们根据文档的主题进行判断，如果文章是关于文 学的，作家王小波的可能性较大，如果文章主要讨论两岸关系，学者王晓波的可能性比较 大。根据语言模型以及主题信息，我们就可以建立一个最大熵模型，同时满足这两种约束 信息。利用这个模型，就可以把上述拼音较为准确地转换为汉字。

**7.2.9** **其他文本分析任务与方法**

**1. 话题检测与跟踪** **(Topic Detection and Tracking,TDT)**

话题检测与跟踪 (Topic Detection and Tracking,TDT) 指的是从新闻专线 (News- wire, 比如 Reuters) 和新闻广播 (Broadcast News, 比如CNN) 等数据流中识别不同的 事件，并且对其进行跟踪的自动化技术。在话题检测和跟踪任务中，“话题”也称为“事 件”(Event)。

话题检测与跟踪技术具有广泛的应用，它使得人们能够及时有效地 (Timely &.Effi- ciently) 对信息进行存取。比如，不同新闻来源的新闻包含大量有用的信息，但是人们没 有精力和时间去仔细查看、阅读不同新闻来源的众多新闻。如果有一个软件能够从这些新 闻数据流中确定出每个事件的边界 (Story Boundary), 发现一些新的事件，并把每个事件 的发展历程组织在一起，那么对于用户是非常有帮助的。这些功能正是TDT 软件所应该 具备的功能。

TDT 的基本任务包括三个，分别是报道切分 (Story Segmentation)、话题检测 (Topic Detection) 和话题追踪 (Topic Tracking)。报道切分的任务是把新闻数据流切分 成关于某个统一主题的报道。话题检测的任务是检测新出现的事件 (New Event)。话题 追踪则是对已知的事件(已经检测过)进行后续报道的追踪。如图7-20所示。

对于某个事件第一次出现的检测称为首次报道检测 (First Story Detection),它 从 具 有时间顺序的报道数据流中自动锁定未知话题的第一篇相关报道。此后，话题追踪则负责 把某一个话题的报道集合都找出来。

从上述对这些任务的描述，我们可以看出，话题的检测和追踪依赖于各个报道之间的 相似度，把不同报道划分到不同的话题中，本质上可以把这些任务看作文本上的聚类。人 们一般采用基于统计原理的文本表示形式，比如向量空间模型来表示各个报道。事件或者 报道之间的相似度计算，则采用余弦夹角和 Hellinger 距离公式等。在此基础上，为了刻



①关于最大熵模型的细节信息，请参考相关资料。

②<http://blog.csdn.net/liqiming100/article/details/70168873.>

|  |  |  |  |
| --- | --- | --- | --- |
|  | Segmentation | Stories-disjoint,homogeneous regions |  |
|  | Detection  Tracking |  |  |
|  |  | More stories about the same event |  |

**图7-20** **话题检测与跟踪的任务**

画每个报道的特点，人们利用自然语言处理技术提取相关特征(比如文本里提及的命名实 体 (Named Entity) 等)。在此基础上，利用常用的聚类算法比如 K-Means 算法，进行报 道的聚类处理。

为了提高话题检测和追踪的性能，算法的设计要尽量做到和新闻数据源无关 (Source Independent) 、 和具体领域无关 (Domain Independent) 以及和语言无关 (Language Inde- pendent)。

与 TDT 任务相关的任务包括：(1)根据事件的重要性进行排序，以便展示给用户；

(2)提取事件的相关信息，包括时间、地点、实体(人物)、实体关系等；(3)标定一系 列报道中 (a Set of Stories) 的冗余信息 (Redundant Information);(4) 为一系列报道生 成摘要信息 (Summary)。

**2.Kleinberg 突发检测算法** **(Burst Detection Algorithm)**

邮件、新闻可以看作流式数据，因为邮件和新闻是随着时间不断到达的。在邮件和新 闻中包含话题，这些话题第一次出现，持续一段时间，然后逐渐衰退以致消失。文档数据 流 (Document Stream) 里的话题的出现和消失，以某些特征 (Feature, 比如单词)的频 率突然上升和逐渐衰弱作为信号。

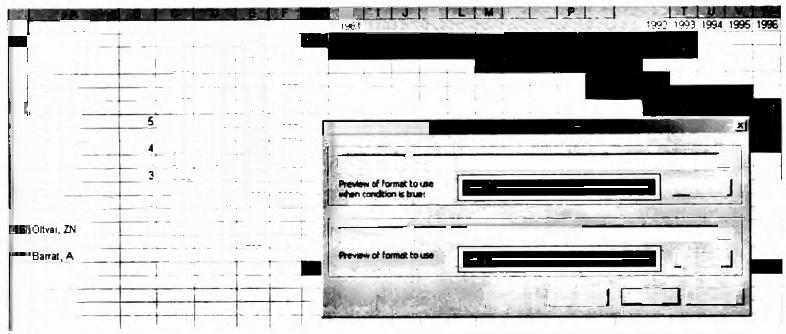
突发检测可以针对这些特征，计算其活跃 (Activity) 的开始时间、持续时间，并且 对这些特征的活跃强度进行排序。通过查看突发检测的结果，人们可以大致了解文档数据 流里的话题变迁。

Kleinberg 于2002年提出的针对文档数据流的突发检测算法，用一个无限状态自动机 (Infinite State Automata) 对数据流进行建模，通过状态的迁移 (State Transition) 检测 特征的突发活动。图7 - 21是对三个社交网络研究者 (Wasserman,Vespignani 和 Barabasi) 的相关论文成果进行突发检测的结果，形象展示了在这些成果中显示出来的各 个主要作者(包括3个研究者的合作者)的崛起和活跃度以及持续时间。

突发检测算法还可以应用到点击流数据 (Click Stream) 以及搜索引擎的查询日志数 据 (Query Log) 上。算法检测出来的突发词汇能够反映人们对特定事件、主题或者网站 的关注程度。

**3.** **机器翻译**

机器翻译 (Machine Translation) 是利用计算机程序，将 一 种语言(称为源语言) 文本转换为另外一种语言(称为目标语言)的文本。机器翻译系统具有重要的应用价 值，它在促进各国人民在政治、经济、文化的交流方面发挥重要作用。 IBM 、微 软 、 Google 、Apple 、 百度、科大讯飞等公司都在机器翻译上投入巨大的资源，进行研究和开



K N 0

Word

Wasseman,S 3 Galaskiewcz,J

lacobucci,D 同 Vicsek,T

6 Pietronero,L

Z Stanley.HE

B Hain.S

Zappen.S

1D Loreto,V

N AJber,R

12 Daruka,I

1Jeong.H

KPastor-satorras,R

16 Vazquez,A

17Barthelemy,M

2004

Femat..

酬 》Dbte OK caned

ondtonalformattimg CondionL

Faxmdats FNG ANCYG51>-年2.0P(41<-5F2.152-7)⁵02>=19)

End 1980

1993

1993

1991

1992

1995

1997

1996

1999

1998

2000

1999

2003

2004

Length Weight Strength

12 1433 14.33 11 3.85 385 6 3.80 380

3 3.56 356

Stant

1980 1981 1986i 1990 1990 1992 1992 1995 1995 1997

1997

1999

2000

2002

2002. 2004

703

738

478

8.30 391

644 360 574 717. 3.57 5.32 497

6.60

7.03

738i 47B

830

391

644

360

574

717

3.57

5.32 497

660

198219831994 1985 198611987 198819891

6

6

5.

4

5. 5

6 6

4L 4

C R|S

19901991

3

Ermm. …

*When condtons bum:*

Candkion2 F

< 一F2.F2-"))

口

一F2.0Q(G1

-AMC541>

**图** **7** **-** **2** **1** **Kleinberg 突发检测实例**

资料来源：Network Workbench Tool User Manual 1.0.0.

发。文献①②对机器翻译技术进行了详细介绍，在此简述如下。

机器翻译系统的处理过程大致可以分为以下几个步骤：(1)对源语言的分析和理解；

(2)进行转换；(3)按照目标语言的语法规则，生成目标语言文本。

根据使用的技术，机器翻译系统可以分为基于规则的 (Rule-Based) 机器翻译系统和 基于语料库的 (Corpus-Based) 机器翻译系统两大类。基于规则的机器翻译系统，依赖于 词典和规则库进行机器翻译，这类系统进一步可以划分成语法型、语义型、知识型和智能 型等。早期的机器翻译系统，大多是基于规则的系统。近年来，基于语料库的机器翻译系 统获得了较大的发展，表现出更加优异的翻译性能。基于语料库的机器翻译系统，无须词 典和规则库而是利用统计规律，在经过标注的语料库上构建翻译模型。这类方法利用庞大 的 语 料 库 ， 获 得 了 非 常 好 的 翻 译 效 果 。

基于语料库的机器翻译系统，包括基于统计的机器翻译系统、基于实例的机器翻译系 统，以及基于人工神经网络系统的机器翻译系统等。

基于统计的机器翻译系统，把机器翻译过程看成是一个信息传输的过程，用一种信道 模型对机器翻译进行建模。机器翻译问题被分解成模型问题、训练问题以及解码问题三个 子问题。其中，模型问题是为机器翻译建立概率模型，它描述了源语言语句到目标语言语 句的翻译概率及其计算方法。训练问题则是利用语料库对模型进行训练，获得模型的参 数。解码问题则是在已知模型和参数的基础上，对任意输入的新句子产生概率最大的翻译 文 本 。

日本机器翻译专家长尾真 (Makoto Nagao) 提出了基于实例的机器翻译系统的基本 思想。他研究发现，人们初学外语时，总是记住最基本的英语句子和对应的日语句子，然 后进行替换。基于实例的机器翻译系统，不需要字典和语法规则库，但是需要双语对照的 实例库。它利用已有的经验知识，通过类比原理实现翻译，并没有经过对自然语言的深层 分析。其翻译过程是，首先将源语言文本分解为句子，再分解为短语片段，接着通过类比 的方法把这些短语翻译成目标语言短语片段，最后把这些短语合并成长句。由于受限于语 料库规模，基于实例的机器翻译系统很难达到较高的准确度。一般在一些专业领域，其翻



①<http://36kr.com/p/533801.html.>

②<https://amta2010.amtaweb.org/AMTA/papers/6-02-DillingerIntroMT.pdf.>



译效果才能够满足实际使用的要求。

2013年以来，随着深度学习技术研究的进展，基于人工神经网络的机器翻译研究迅 速兴起。基于人工神经网络的机器翻译系统的核心是一个拥有大量神经元的深度神经网 络，它可以从语料库中自动学习翻译模型。在机器翻译中， 一般结合使用长短时记忆网络 (Long Short Term Memory,LSTM) 和循环神经网络 (Recurrent Neural Network, RNN) 技术。比如，Google 公司的研究人员使用一个多层的ISTM 神经网络，把输入的 序列 (Sequence, 也就是源语言的 一 个语句)映射到固定维度的向量(称为词向量 (Word Vector)), 然后用另一个LSTM 神经网络，直接从该向量解码 (Decode) 出目标 语言序列。他们的模型在 WMT’14(Shared Tasks of the 2014 Workshop on Statistical Machine Translation) 数据集上，执行英语到法语的翻译任务，获得比统计机器翻译模型 更高的性能。

**4. 问答系统** **(Question Answering System,QA)**

传统的信息检索系统能够返回包含用户查询关键字的文档和段落，但是它缺乏对问题 的深入理解，不能以一定的置信度 (Confidence) 直接返回正确的答案。问答系统是一种 高级的信息检索系统。问答系统接收用户用自然语言提出的问题，经过对问题的分析和理 解，它能够用准确、简洁的自然语言给出答案。问答系统适应了人们快速、准确获取信息 的需求。文献①②③对问答系统进行了详细介绍，在此简述如下。

根据处理问题与答案的形式不同，问答系统可以分为两类，包括：(1)定型的问答系 统；(2)开放领域 (Open-Domain, 也称不限定领域)的问答系统。在定型的问答系统 中，系统需要回答的问题是关于特定事实的，比如“首都国际机场在哪里”,或者是专业 性的.比如“霍乱有什么症状”。

开放领域的问答系统可以进一步细分为如下几类：(1)能够处理事实问题的问答系 统。这类系统从若干文献构成的集合中抽取文本片段作为回答。(2)能够从不同文献中融 合出答案的问答系统。这类系统能够提取散落在若干文献中的局部的信息，然后形成一个 融合的答案。(3)具有推理能力的问答系统。这类系统需要在不同的文本片段中找出答 案，并且用简单的推理形式找出问题与这些答案之间的关系，并且把它们关联起来。有些 问答系统具有类比推理的能力。比如问题是“安然公司遇到了什么麻烦”,要回答这个问 题。需要从各种文本中提取证据的碎片，然后进行类比推理，构造出答案。(4)交互式问 答系统。这类问答系统，人与计算机之间可以进行交互。计算机能够在前期与用户互动形 成的情境的基础上，提出新问题而不是提出孤立的一些问题。

2011年12月， IBM 的人工智能计算机系统“沃森”(Watson) 在电视智力问答游戏 节目《危险边缘》 (Jeopardy!) 中战胜两位人类冠军选手，获得第一名，给人们留下了深 刻的印象。 Watson 是一个不限定领域的高级问答系统。Watson 问答系统的工作过程分为 两个阶段：(1) Watson 需要针对一个主题 (Subject) 进行学习。用户把与该主题相关的 文档装载到 Watson 系统中，包括Word 文档、PDF 文件、网页等，用户同时提供一些 “问题/回答”示例给Watson 系统，在该主题上对Watson 系统进行训练。(2) Watson 回 答用户提出的问题。当用户提出问题， Watson 开始分析和理解该问题，并且搜索大量的

①<https://my.oschina.net/apdplat/blog/420370.>

②<https://my.oschina.net/apdplat/blog/420720.>

③<http://blog.csdn.net/yzzky/article/details/49252509.>

文档(上百万级),找到上千个候选答案。然后它收集证据，用一个评分 (Scoring) 算法 对这些证据进行评分。最后基于这些证据评分，对所有备选答案进行排序，选出最优的答 案返回给用户。

传统的基于知识 (Knowledge-Based) 的问答系统，根据领域知识和问题的逻辑表示， 试图证明某个答案是正确的，然后返回。这类问答系统的主要缺点包括：(1)需要大量的 人力去获取知识，并且对知识进行正式的编码，以便计算机程序能够理解和使用这些知 识；(2)这类系统缺乏深入理解以自然语言表达的问题的能力，要利用经过编码的知识， 这是必要的前提。在特定领域，这类问答系统能够获得非常高的准确率，但是对于开放领 域的问题，则显得力不从心。IBM 在传统的基于知识的问答系统技术的基础上，应用了先 进的自然语言处理、信息检索、知识表示和推理、机器学习等技术，研发了DeepQA 核心 技术。Watson 使用该核心技术，在问答中完成假设生成 (Hypothesis Generation)、大量 证据的收集 (Massive Evidence Gathering)、证据的分析和评分 (Analysis and Scoring), 最后给出系统认为最优的答案。

除 了Watson 问答系统，还有很多开源和商用的问答系统。开源的问答系统有 Ques- tion Answering System(Java 开源)、(penEphyra(Java 开源)、 Watsonsim(Java 开 源)、YodaQA(Java 开源)、(penQA(Java 开源)。商用系统有 START 、Apple Siri、 Wolfram|Alpha 、Evi 、 微软小冰、Magi Semantic Search等。

其中，Question Answering System是一个用Java 语言实现的人机问答系统，它能够 自动分析问题并给出备选答案。目前它支持5种问题类型(答案类型),包括人名、地名、 机构名称、数字、时间等问题类型。它的工作过程是：(1)判断问题类型(答案类型), 目前它使用模式匹配的方法来实现，将来会支持更多的方法，比如朴素贝叶斯分类器；

(2)提取问题关键字；(3)利用问题关键字搜索多种数据源，当前的数据源主要是人工标 注的语料库以及谷歌、百度等搜索引擎的搜索结果；(4)从搜索结果中，根据问题类型 (答案类型)提取备选答案；(5)结合问题以及搜索结果对备选答案进行打分；(6)返回 得分最高的前 N 项 (Top N) 备选答案。

在 Question Answering System的系统架构中包含四个主要模块，具体如下：(1)证 据获取。从本地数据库或互联网上获取支撑问题的证据。若本地数据库存储有该问题的证 据，则直接返回这些证据，否则，利用搜索引擎(如百度、谷歌)从互联网上抓取与该问 题相关的网页，并抽取出其中的正文作为该问题的支撑证据。(2)证据评分。采用基于词 频的、Bi-gram 的和Skip-Bigram 的三种评价方法，以及基于上述三种方法加权的组合方 法，评价不同证据对问题的支撑度。(3)问题分类。该系统可识别的问题类别包括人名、 地名、机构名、数字、时间、定义和对象七类(目前仅支持前五类),并预先定义这几类 问题的匹配模式。对问题所属的类别进行判定。分类过程为： (a) 提取问题的模式。 (b) 和预定义的问题类型模式进行正则匹配。这些模式可以分为三大类.包括直接匹配模 式、基于问题分词的词与词性的匹配模式和基于问题主谓宾的词与词性的匹配模式。在建 立模式之后，先按选定的某类模式提取问题的模式，再与所有问题类型模式正则匹配，最 后将得票最多的类型作为问题的类别。(c) 根据匹配的结果确定问题的类别。(4)备选答 案评分。评价备选答案的可信度。评分过程为：(a) 根据问题类型确定答案类型，然后从 证据词集中筛选出命名实体标记与答案类型一致的词，作为备选答案。(b) 针对每个备选 答案，利用评分模型进行打分。系统包含七个评分模型，即基于词频的、基于词距的、基于 最短词距的、基于文本对齐的、基于宽松文本对齐的、基于回带文本对齐的和基于热词的



模型①,以及一个组合模型(即前七个模型的线性加权)。在组合模型的打分过程中，每类 评价方法都有一个权值。备选答案的得分是评价方法的评分与该评价方法权值的乘积的和。

Question Answering System 系统的证据及备选答案评分模型，参考了IBM Watson 系统的文本证据收集与分析方法。该系统的文本预处理(分词、词性标注及依存句法分 析)功能，则使用开源工具stanford-parser-3.3.1来实现。

**7.3** **文本分析可视化**

**7.3.1** **标记云**

标记云 (Tag Cloud, 也叫 Word Cloud) 是文本数据可视化的一种形式。一般用来描 绘一个网站的元数据 (Metadata) 的关键字，或者对自由文本里的词汇进行可视化。显示 的单元一般是一个个单词(也称标记),每个标记 (Tag) 的重要性(比如频率),通过大 小或者颜色来进行区分。Tag Cloud 可以使得用户迅速对文本里最重要、最突出的单词获 得一个直观的感受。

Tag Cloud可以和话题建模技术组合，首先对一个文档集进行话题建模，把文档划分 成不同话题，然后对每个话题的文档子集提取重要的Tag, 把这些Tag 在一个Tag Cloud 里面展示，颜色则表示不同的话题，其大小表示重要性。

Wordle 网站 (<http://www.wordle.net/>) 提供了免费的 Word Cloud生成功能，图

7- 22是基于Wikipedia 最重要的1000个文档生成的Word Cloud。



**图7-22** **Word** **Cloud实例**

资料来源：<https://en.wikipedia.org/wiki/File:Wikipedia_Wordle_> 一 \_Top\_ 1000\_vital\_article\_hits png.

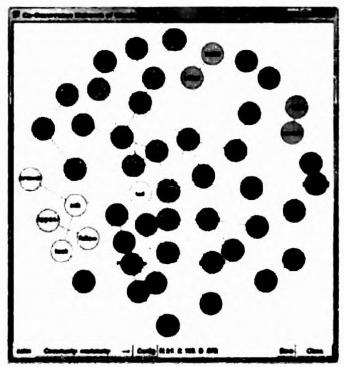
**7.3.2 词共现分析与可视化** **(Co-Word Analysis &Visualization)**

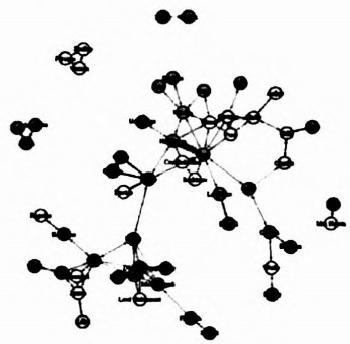
词共现分析指的是对文章的段落、语句里每对词汇的共现 (Co-Occurrence) 进行统

①关于这些模型的详细信息.请参考相关资料。



计，记录其共现的频率，进行后续分析。对我们要研究的词汇列表，可以记录每对词的共 现频率并记录在一个矩阵C 里 。C; 表示 Word, 和 Word, 的共现频率，这些频率可以进行 规范化处理，称为共现系数 (Coefficient)。

在此基础上，我们可以使用凝聚层次聚类算法 (Agglomerative Hierarchical Cluste- ring Algorithm)、多维尺度分析方法 (Multi-Dimensional Scaling,MDS) 以及社交网络 社区检测方法 (Community Detection) 进行分析。(1)凝聚层次聚类算法，首先把每个 词汇看作一个类簇，然后不断地合并两个最相似的类簇，直到指定的类簇个数，从而把词 汇分解到不同的类簇中。(2)多维尺度分析方法，是一种降维方法，它把高维空间的向 量映射到低维(比如2、3维)向量，使得低维空间中向量间的距离，和高维空间中向 量间的距离，保持一定比例，方便可视化和观察(下文详述)。(3)社交网络社区检测 方法则把词汇看作图的节点，词汇间的共现系数决定节点间的连接强度，这些词汇构成 一个带权重的无向图 (Weighted Undirected Graph)。利用社交网络社区检测方法，可以 把词汇划分到不同的社区，进而观察和分析。图7-23是利用社区检测方法对词汇的社 交网络进行分析的结果。



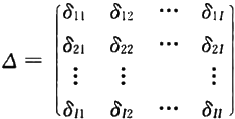
(a) (b)

**图7-23** **词汇社交网络与社区检测算法结果**

资料来源：(a):<https://sites.google.com/site/casualconc/.>

(b):<http://khc.sourceforge.net.>

下面，我们对上述方法中的多维尺度分析方法作进一步介绍。比如，要分析的句子集 包含i 个对象(句子)。我们定义一个距离函数，并且计算第i个对象和第j 个对象之间的 距离δ:.;。然后，把这些距离组织成一个矩阵，如下：



MDS 算法的目的是根据这个△寻找I 个向量x₁,x₂,…,x₁∈R, 使得|x;一x; | ≈8;,

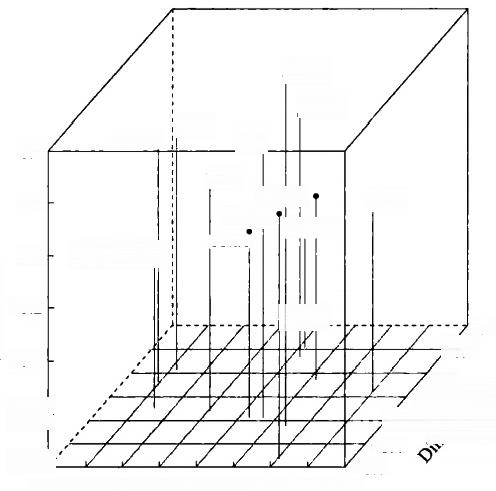
Vi,j∈I。 这里| · |是向量之间的范数。在经典的MDS 算法中，该范数为欧式距离。



从这里可以看出，MDS 试图把I 个对象嵌入到子空间RN 中，并且保留这些对象之间 在高维空间的相似度。当子空间的维数 N 为2或者3,我们就可以画出对象相似性的一个 可视化结果。为了得到向量xi,一般把MDS 看作一个优化问题，寻找{x₁,x₂,…,x₁}, 并

且最小化目标函数  。这个问题可以通过数值优化的方法得 到最优解。

Dimension 3



.think

make Red

tell

say school house

0.0

-0.2

room

-0.4

0.2

0.0

-0.6

-0.8

-0.8-0.6-0.4-0.20.00.20.40.60.8

teacher come

-0.2 

0.6 0.4

-0.4 -0.6

0.4

0.2

Porcupine

Shirt。

time

Dimension 1

图7-24 MDS 实例

资料来源：<https://sourceforge.net/p/khc/wiki/Multi-Dimensional%20Scaling/.>

KH Coder是一款免费的内容分析和文本挖掘软件 (Content Analysis &Text Min- ing), 它能够处理日语、英语、法语、德语、意大利语、葡萄牙语以及西班牙语等不同语 言的文本。KH Coder 可以实现 KWIC 分 析 (Keywords in Context)、搭配统计分析 (Collocation Statistics,Term 之间带顺序关系)、共现网络分析 (Co-Occurrence Net- work,Term 之间不带顺序关系)、 SOM 分 析 (Self-Organizing Map)、多维尺度分析 (Multi-Dimensional Scaling)、聚类分析 (Cluster Analysis)、社区检测 (Community De- tection) 、 对应分析 (Correspondence Analysis) 等。

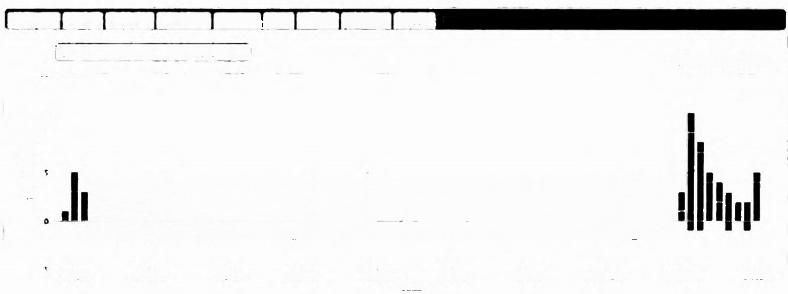
除了上述实例，其他文本分析结果可视化实例包括情感分析可视化、话题分析可视 化、聚类分析可视化等。图7-25和图7-26分别是情感分析可视化和话题分析可视化的 实例。

图7-25显示的是关于大峡谷 (Great Canyon) 的 Tweet 里体现的情感随时间变化的 趋势。x 轴显示时间刻度，y 轴叠加显示各类情感的比例。

图7-26显示1790—2014年美国年度国情咨文 (Annual State of the Union Address, SOU) 主题的变迁， x 轴表示各个年度， y 轴表示各个年度SOU 讨论各个话题 (Dis- course Cluster) 的比例，虚线表示不同的历史时期。



1 哺 C a u



s Ta ~m MI. Marrat Tvest

|  |  |  |  |
| --- | --- | --- | --- |
| ■Rrlaed■Happv■Unhapoy | ■Upvet |  |  |
|  |  | grand | camyon(169) |
| 15 |  |  |  |

10

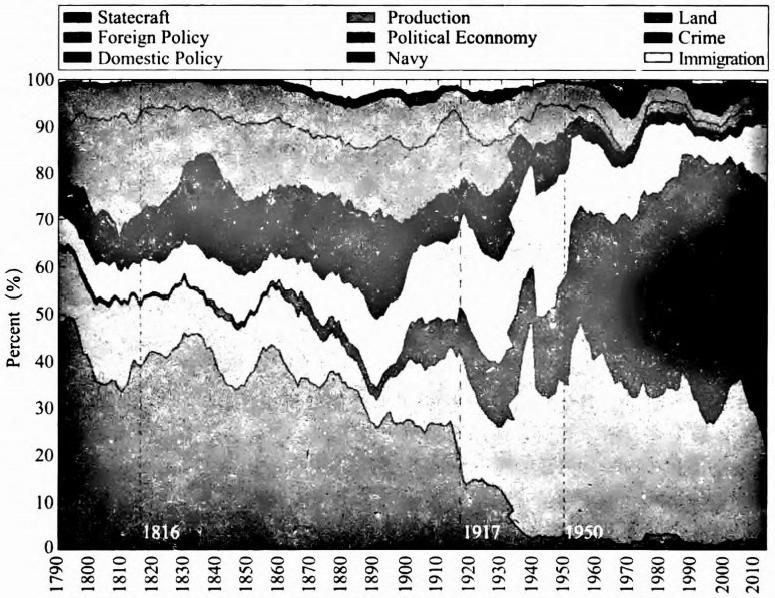
ali h. …… u |1.

May 19 200m 400m 6.00u *800am* 10.90am 1200pm 260pm *Time*

800pm 1000pm

**图7** **-** **25** **情感分析可视化**

资料来源：<https://www.csc.ncsu.edu/faculty/healey/tweet_viz/.>



Years

**图7** **-** **26** **话题分析可视化**

资料来源：<http://www.pnas.org/content/112/35/10837.>

图7-27是文本可视化软件系统TIARA 的话题分析结果的可视化效果。从文本集分 析得到的话题，用堆叠的河流 (Stacked River) 的形式展示出来。河流的宽度对应每个话 题 的 相 对 权 重 ( 或 者 强 度 (Strength)), 每个话题的Top-K 关键字也显示在各个话题的河

流 中 。

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | |  |  |  |  | |  | l |
|  | | |  | | |  |  |  |  | |  |
|  | | |  | | |  |  |  | asia as | | ias build center china chinese  ting aweek fire hebei help host jounna local open range review storage teehnol |
| stronq  upbeat  fourth | | |  | | |  |  |  |  | compu  largest |
|  | | |  |
| past partner  qqpl power  stock  low | | | | | | nounoed  hipsets ibmer  chips films  ductor happy tweeted  tnerinq centennial ies  rate akianastasiou check cen nics achie  res face  slashdot savs arcticstoat  enqadget move graphene  zdnet revealed replace  tusphere qhz full | | tury  vements  achievemen  silicon on cpus | access  voutube  unveils simart  celebrates  100×100  chanqed  filmibm100  Ist vdi rolls  cha liked 3rd  virtual desktop offerinq    packaqe ablewont writes  ram ioffer svmposium  ibmpartners event aqenda | | |
| qrowth deals  billion stronq beats  street | nyse rise  earninqs  stocks unix  results blow  manaqer  quarter | | | mobile  an  arm c  advanced  semicon  partn par | |
| net fourth  posts rises income  percent  29b revenue | | quarterlv  delivers  record  released quarter | | | compan  collabo  electro teams |
| dvd pentium drive  intel forsale hard lotusknows Is11 lo | | | | | |

2011-01-182011-01-19 2011-01-20 2011-01-21 2011-01-22 2011-01-23 2011-01-24 2011-01-252011-01-26

**图7-27** **TIARA 话题分析可视化**

资料来源：<http://cgcad.ths.tsinghua.edu.cn/shixial/talk/Visual%20Text%20Analytics.PacificVis.2012.pdf.>



**7.4** **文本分析软件和工具**

**7.4.1 NLTK**

NLTK 是使用Python 语言编写的自然语言处理软件库。其具体信息以及实例，请参 考第14章 “Python 与数据科学”。

**7.4.2 OPEN NLP**

Apache 开源项目OpenNLP 是一个基于机器学习技术的软件库，用于自然语言处理。 OpenNLP 支持最常见的自然语言处理任务，包括句子切分、分词、语法解析、词性标注、 命名实体识别、指代消解 (Co-Reference Resolution)、文档分类等。这些基本任务用于构 建更加复杂的文本分析程序。为了实现这些任务。(OpenNLP 使用了包括最大熵、基于感 知机的机器学习等先进的算法。

(penNLP 的目标，一方面是创建完成上述任务的成熟的软件库；另一方面是提供大 量的预训练模型，方便用户使用这些模型，快速建立文本分析应用程序。

**7.4.3 Stanford NLP**

斯坦福大学的NLP(Naturual Language Processing) 团队把他们的自然语言处理软 件开源出来，供研究人员使用。这些工具采用了统计方法、深度学习方法以及基于规则的 方法，解决常见的自然语言处理问题。Stanford NLP使用Java 语言编写，同时提供了其 他语言的接口，包括 Python,Ruby,Perl,JavaScript,F# 等语言编写的程序，都可以 调用该工具软件的功能。Stanford NLP已经被工业界、学术界以及政府部门采用。

Stanford NLP 提供了句子切分、分词、语法树分析以及词性标注、命名实体识别、 关系抽取、指代消解、情感分析以及文本分类等功能。其中，文本分类器把文本数据划分 到预先制定的K 个类别中的某一个类别。该分类器是一个最大熵 (Maximum Entropy) 分类器，它是一个概率分类器，能够给出某个文本数据在各个类别上的软分类。

**7.4.4 LingPipe**

LingPipe 是 Alias 公司开发的一款自然语言处理开源软件包，使用Java 语言进行开 发。它提供了详细的文档，并提供大量经过预先训练的模型，方便用户使用。LingPipe 提 供丰富的功能，包括语言标识 (Language Identification)、句子检测 (Sentence Detec- tion) 、词性标注 (Part-of Speech Tagging)、聚类 (Clustering) 、 主题分类 (Topic Clas- sification) 、情感分析 (Sentiment Analysis) 等。LingPipe 支持中文分词功能 (Chinese Word Segmentation), 方便中文信息的处理。

**在命名实体识别方面，LingPipe 提供三类技术，包括基于正则表达式的命名实体识** **别、基于字典的命名实体识别，以及基于统计模型的命名实体识别。基于正则表达式** (**Regular Expression) 的命名实体识别，找出文本中符合正则表达式的符号、单词或者短** **语，一般用于邮件地址、数值、货币等实体的识别。基于字典的命名实体识别则利用一个** **字典(字典里包含<短语、类别>列表),找出文本里匹配的单词或者短语，** **一般用于人** **名、地名的识别。基于统计模型的命名实体识别，则事先利用一系列的<短语、类别>** 对，训练一个模型，然后用这个模型去预测新文本里的单词或者短语的类别，这些单词或 者短语，有可能在训练集里并没有见过。

**7.4.5 GATE**

GATE 是 General Architecture for Text Engineering的缩写，它是文本信息抽取和处 理的开放式软件框架。GATE 为用户提供图形化的开发环境，被许多自然语言处理项目尤 其是信息抽取项目采用。目前，应用GATE 的自然语言处理应用程序种类繁多，包括 Web 信息挖掘 (Web Mining)、信息抽取 (Information Extraction)、语义标注 (Semantic Annotation) 、 癌症研究 (Cancer Research)、药物研究 (Drug Research)、招聘 (Recruit- ment)、客户情感分析 (Voice of the Customer) 等。

GATE 是一个开放的软件架构。它有三个主要目标，首先，它为自然语言处理提供一 个基础框架；其次，它提供可以重用的组件，嵌入到不同的应用程序中使用；最后，它提 供一个开发环境，为自然语言处理软件的研究者和开发者提供便利的图形化环境，进行可 视化开发和调试。

自然语言处理的功能被分解成几种主要类型的组件 (Component, 也称为资源 (Re- source)) 。GATE 组件有三种类型，具体形式为Java Bean。这三种类型是：(1)语言资 源 (Language Resource,LR), 包括词典 (Lexicon) 、 文献库 (Corpora) 或者本体库 (Ontology) 等。(2)处理资源 (Processing Resource,PR), 是执行某种功能的软件模 块，比如各种解析器 (Parser)、生成器 (Generator) 以及n-Gram 模型等。(3)可视化资 源 (Visual Resource,VR), 是图形用户界面上的可视化组件或者可以编辑的用户界面 控件。

所有的资源(组件)打包成Java 归档文件 (Jar), 附 带 XML 配置信息。集成到 GATE 平台的一批资源(组件),也称为CREOLE(a Collection of Reusable Objects for Language Engineering) 。GATE平台已经包含常用的自然语言处理的数据结构和算法组件 (资源),包括文档、文集及各种标注，信息抽取和语言分析的组件，以及数据可视化和可 以编辑的用户界面控件，方便用户使用。

**7.4.6 UIMA**

Apache UIMA 是 Apache 基金会对UIMA 规范的开源实现。 UIMA 是 Unstructured Information Management Applications的简称，指的是分析大规模非结构化信息以发现知 识的软件系统。比如，有的 UIMA 软件从文本中标定各类实体，包括人物、地理位置、 组织等，或者标定各种关系.比如人物和组织的“为其工作”关系等。这里的非结构化信

息，不仅包括文本，而且包括音频、视频等信息。

Apache UIMA是一个软件框架，它把非结构化信息的处理功能分解成一系列的组件， 这点和GATE 使用的策略是类似的。软件框架负责管理和运行这些组件，以及组件间的 数据流动和交换。组件可以用Java 或者C++ 语言来编写。系统已经提供了分词、命名实 体识别、词性标注 (Hidden Markov Model based POS Tagger) 等组件，方便用户调用。 用户可以按照规范，自己编写各种特殊用途的组件。

UIMA-AS 和 UIMA-DUCC 是 UIMA 的附加组件。通过UIMA-AS, 可以实现基于消 息队列 (Java Messaging Services.ActiveMQ 等)的系统扩展。通过 UIMA-DUCC, 则 可以把 UIMA 应用程序的处理流水线，扩展到集群环境上运行。

**7.4.7 Netlytic**

Netlytic([www.netlytic.org](https://www.netlytic.org)) 是开源社区开发的文本分析和社交网络分析工具，它 能够从社交网站的在线会话 (Online Conversation) 中自动发现社交网络。

Netlytic提供了如下功能：(1)从Twitter,Facebook,YouTube,Instagram,RSS

Feed 等社交媒体网站或者 txt/csv 文件中抓取数据；(2)自动发现流行的话题 (Popular Topic);(3) 发现会话或者讨论中正在涌现的主题 (Emerging Theme);(4) 从在线会话 数据中自动发现社交网络关系，并对其进行可视化以及分析；(5)对社交媒体数据进行地 理位置标注，在地图上进行可视化以及汇总分析。

**7.4.8 WordNet 和** **SentiWordNet**

WordNet 是普林斯顿大学的计算机科学家、语言学家和心理学家共同设计的英语词 典，它把单词按照语义组织成一个单词的网络。名词、动词、形容词和副词，各自被组织 成一个同义词的网络。每个同义词集合 (Synset) 都代表了基本的语义概念。这些集合间 有各种连接关系。主要的关系包括上下位关系 (Hyponymy) 、 整体部分关系 (Merony- my) 、 一词多义关系 (Polysemy) 、 同义关系 (Synonymy) 等。

SentiWordNet 是一部词典，用于情感分析和观点挖掘 (Opinion Mining)。它 为 WordNet 的每个同义词集合赋予了三个情感打分信息，包括正面情感评分 (Positivity) 、 负面情感评分 (Negativity) 、 客观评分 (Objectivity)。

**7** **.** **5** **思** **考** **题**

(1)举例说明文本分析的意义。

(2)文本分析的主要任务有哪些，文本分析的过程主要有哪些步骤。

(3)Part-of-Speech Tagger。

(4)文本索引和检索的向量空间模型和TF-IDF 方法。

(5)文本分类及主要方法。



(6)文本聚类及主要方法。

(7)文档摘要及主要方法。

(8)主题抽取及主要方法。

(9)命名实体识别/事实抽取/概念抽取和关系建模及主要方法。

(10)情感分析及主要方法。

(11)话题检测与跟踪 (TDT), 首次报道检测任务 (FSD)。

(12)突发检测算法 (Burst Detection Algorithm)。