文本预处理

文本挖掘的背景

1. 文本挖掘：将数据挖掘的成果用于分析以自然语言描述的文本，这种方法被称为**文本挖掘**或**文本知识发现**。
    - 与数据挖掘的区别：文本挖掘的对象是半结构化或非结构化的，无确定形式；数据挖掘的对象以数据库中的结构化数据为主。因此，数据挖掘的技术不适用于文本挖掘，或至少**需要预处理**。

分词

1. 文本特征：关于文本的元数据。
    - 描述性特征：文本的名称、日期、大小、类型等；
    - 语义性特征：文本的作者、标题、机构、内容等。
2. 特征抽取：预处理 -> 文本表示 -> 降维技术。
3. 词语标记（Tokenization）和词性还原（Lemmatization）。
    - 词语标记：输入一段文本，输出单词串。
    - 词形还原：所需的知识库：（1）词典；（2）前缀表；（3）后缀表；（4）有关屈折词尾变形的规则。
4. 汉语词法分析面临的问题：（1）重叠词、离合词、词缀；（2）汉语词语的切分歧义（交集型歧义、组合型歧义、混合型歧义）；（3）汉语未登录词。
5. 分词的基本算法：最大匹配法、概率方法。

文档模型

1. **布尔模型**：建立在经典的集合论和布尔代数的基础上，每个词在一篇文档中是否出现，对应权值为0或1。
    - 特点：将文档检索转化为布尔逻辑运算；
    - 优点：简单、易理解、形式简洁；
    - 缺点：信息需求的能力表达不足。
2. **词袋模型（Bag-of-Words, BoW）**：忽略掉文本的语法和语序等要素，将其仅仅看作是若干个词汇的集合，文档中每个单词的出现都是**独立的**。
    - 术语权重（Term weight）：（1）0/1形式；（2）词频（TF）形式；（3）词频-逆文档率（TF-IDF）形式。
3. n**-gram模型**：也称N元语法模型，是一种基于统计语言模型的算法。n表示n个词语，n元语法模型通过n个词语的概率判断句子结构。
    - 基本思想：对文本内容按照字节按照**大小为**N**的滑动窗口**进行划分，形成长度为N的字节片段序列，每个片段称为**gram**。统计对所有gram的频度，且按照事先设定好的阈值进行过滤，形成关键gram列表，也即该文本的向量特征空间。列表中的每一种gram就是一个特征向量维度。

- 理论依据：**马尔科夫假设**，即第N个词的出现只与前$N-1$个词相关，而与其他任何词都不相关。整句的概率就是各个词出现概率的乘积。

> w作为第i个单词出现的概率，只取决于前$t(t \geq 1)$个单词：
>
> $$P(w_i|w_0w_1\cdots w_{i-1}) = P(w_i|w_{i-t}w_{i-t+1}\cdots w_{i-1})$$

- 应用场景：输入法提示、搜索引擎等。

4. bi/tri-gram模型：即二/三元语法模型，每个单词出现的概率只与前两/三个单词有关。

> $P(S)$为整个句子出现的概率，应等于句子中每个单词出现概率的乘积：
>
> $$P(S) = P(w_1, w_2, \cdots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_2)\cdots P(w_n|w_{n-1})\ (N=2)$$
> $$P(S) = P(w_1, w_2, \cdots, w_n) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2)\cdots P(w_n|w_{n-2}w_{n-1})\ (N=3)$$
>
> 对于一般的n-gram模型而言，条件概率可由**极大似然估计**得到：
>
> $$P(w_i|w_1,\cdots,w_{i-1}) = \frac{c(w_1,w_2,\cdots,w_i)}{\sum_w c(w_1,w_2,\cdots,w_i,w)} = \frac{c(w_1,w_2,\cdots,w_i)}{c(w_1,w_2,\cdots,w_{i-1})}$$
>
> 为了使句首的条件概率有意义，需要给原始序列加上一个或多个起始符（$\langle BOS\rangle$, Begin Of Sentence），同理也需要加上一个或多个结束符（$\langle EOS\rangle$, End Of Sentence）。

5. **向量空间模型（VSM）**：将文档表达为一个矢量，视作向量空间中的一个点。

文档相似度计算

1. 文本相似度：表示两个文档、两个查询或一个文档与一个查询之间的相似度。
   - 方法：（1）基于概率模型的相关度；（2）基于VSM的相关度。
2. 基于概率模型的相关度：

> 查询与文档之间的相关度：
>
> $$\mathrm{sim}(d_j, q) \sim \sum_{i=1}^{t} w_{iq} \times w_{ij} \times \left(\log\frac{P(k_i|R)}{1-p(k_i|R)} + \log\frac{1-P(k_i|\overline{R})}{P(k_i|\overline{R})}\right)$$

3. 基于VSM的相关度计算方法：（1）欧氏距离；（2）向量内积；（3）向量夹角余弦。
   - 欧氏距离：$\mathrm{Dis}(x,y) = |x-y| = \sqrt{\sum_{k=1}^{t}(x_k-y_k)^2}$。
   - 向量内积：$\mathrm{Sim}(x,y) = x\cdot y = \sum_{k=1}^{t} x_k y_k$。
   - 向量夹角余弦：$\mathrm{Sim} = \dfrac{x\cdot y}{|x||y|} = \dfrac{\sum_{k=1}^{t} x_k y_k}{\sqrt{\sum_{k=1}^{t} x_k^2}\sqrt{\sum_{k=1}^{t} y_k^2}}$。
     - 相当于先对向量进行单位化（$x' = \dfrac{x}{|x|} = x/\sqrt{\sum_{k=1}^{t}(x_k^2)}$），再计算向量内积。
4. Jaccard相似度：$\mathrm{Sim}(x,y) = \dfrac{x\cdot y}{|x|+|y|-x\cdot y} = \dfrac{\sum_{k=1}^{t} x_k y_k}{\sum_{k=1}^{t} x_k^2 + \sum_{k=1}^{t} y_k^2 - \sum_{k=1}^{t} x_k y_k}$。
5. 文本序列的相似度：额外考虑文本的顺序。

- 四种情况：
  - 两条长度相近的序列相似，找出序列的差别；
  - 一条序列是否包含另一条序列（子序列）；
  - 两条序列中是否有非常相同的子序列；
  - 一条序列与另一条序列的逆序列相似。
- 距离计算方法：（1）海明距离；（2）编辑距离。