慧科 只为赋能人才 | 开课吧 kaikeba.com

文本分类

jwzheng

课程大纲

- 文本分类
- 基本知识
  - 分词
  - 语言模型
  - tf-idf
- 基于传统机器学习算法的文本分类 (sklearn)
  - 逻辑回归,决策树,支持向量机
- 基于深度学习算法的文本分类（keras，tensorflow）
  - 词向量
  - CNN（TextCNN）
  - LSTM（TextRNN）
  - fastText

第一节课课程计划

- 什么是文本分类
- 文本分类和图像分类
- 文本分词
- 停用词
- 统计语言模型
- tf-idf
- 实践：文本分词
- 实践：计算tf-idf
- 文本的表示方法
- 实践：基于传统机器学习算法的文本分类

文本分类

- 什么是文本分类
  - 二分类
  - 多分类
  - 多标签
- 文本分类的应用
  - 垃圾邮件识别（是否是垃圾邮件）
  - 情感分析（好评 差评）
  - 电影分类（喜剧 战争 动作 犯罪）
- 文本分类方法
  - 传统机器学习
  - 深度学习

文本分类VS图像分类

- 图像分类方法回顾
  - 传统机器学习算法
    - 图片表示
  - 深度学习网络
    - 端到端
    - 卷积神经网络
    - 全连接神经网络（mnist），循环神经网络
- 文本分类方法
  - 传统机器学习算法
    - 文本表示（bert模型）
  - 深度学习网络
    - 文本表示
    - 卷积神经网络
    - 全连接神经网络，循环神经网络

分词

- 什么是分词
  - 中国航天官员应邀到美国与太空总署官员开会
  - 中国/航天/官员/应邀/到/美国/与/太空/总署/官员/开会
- 为什么要分词
  - 很多自然语言处理的任务是基于词来进行的——文本表示
  - 中文需要分词，英文不需要
  - 例如：
    - 明天我们一起去爬山吧
    - Let's go climbing together tomorrow.
- 如何实现
  - 基于词典：最长匹配（下一节讲）
  - 基于统计模型：马尔科夫模型（后面会讲）

如何实现分词

- 基于词典：最长匹配
  - 词典：北京 北京大学 大学生 学生 大学
  - 查字典方法的局限性
    - 发展中国家=> 发展/中国/家 发展/中/国家
    - 北京大学生=>北京大学/生 北京/大学生 北京/大/学生
    - 研究生命的起源=>研究生/命/的/起源 研究/生命/的/起源
- 基于统计模型：马尔科夫模型
  - 该方法理论性较强
- 分词代码实现
  - jieba

如何实现分词

- 停用词（Stop Words）
  - 北京的大学生=> 北京/的/大学生
  - 研究生命的起源=> 研究/生命/的/起源
  - 他创造了这个短语=> 他/创造/了/这个/短语
  - 注意：一些特殊符号，标点符号也可以被认为是停用词?”” 吗
- 停用词表
  - 人工维护
- 在分词结果中去掉停用词
  - 较少特征数量
  - 降低训练时间

分词code实例

- 这里展示一个分词的代码，使用jieba包
  - 分词
  - 加载用户自定义词典
  - 去停用词

语言模型

- 什么是语言模型
  - 给句子建立模型，给出每个句子出现的概率。本质：计算一个句子的概率的模型。
- 一个例子
  - 美联储主席本伯南克昨天告诉媒体7000亿美元的救助金将借给上百家银行 保险公司和汽车公司。
  - 本伯南克美联储主席昨天7000亿美元的救助金告诉媒体将借给银行保险公司和汽车公司上百家。
  - 美联储汽车主席本伯南媒体克昨天告诉70亿美00元的将上百家银行保险救助金公司和公司借给。
- 基于规则
  - 看这个句子是否合乎文法，含义是否正确。
- 基于统计
  - 一个句子是否合理，就看生成这个句子的可能性大小如何。可能性大小使用概率来衡量。
- 统计语言模型
  - 本质：计算句子的概率

统计语言模型

例如 S = 猴子/吃/桃

- 问题描述：
  - 假设S表示一个句子，由一连串的词 $w_1 w_2 w_3 .. w_n$ 组成，n是句子的长度。现在，想知道句子S出现的概率 $P(S)$，可以利用下面的思路。
- 思路一：暴力统计。将人类有史以来讲过的话统计一下，就可计算得到 $P(S)$
- 思路二：统计语言模型。

$$P(S) = P(w_1, w_2, w_3, ..., w_n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2)...P(w_n \mid w_1, w_2, ..., w_{n-1})$$

- 马尔科夫假设 $P(w_n \mid w_1, w_2, ..., w_{n-1}) = p(w_n \mid w_{n-1})$ $P(w_n \mid w_1, w_2, ..., w_{n-1}) = p(w_n \mid w_{n-1}, w_{n-2})$

$$P(S) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_2)...P(w_n \mid w_{n-1})$$

- 上面的是二元模型
- 有前面的N-1个词决定的模型被称为N元模型（ N-Gram ）

统计语言模型

条件概率公式

$$P(A|B) = \frac{P(AB)}{P(B)}$$

- 如何估计条件概率

$$p(w_i \mid w_{i-1}) = \frac{p(w_{i-1}, w_i)}{p(w_{i-1})}$$

- $p(w_{i-1})$和 $p(w_{i-1}, w_i)$ 的计算基于语料库(非常大的中文样本)

$$p(w_{i-1}) = \frac{N(w_{i-1})}{N} \qquad p(w_{i-1}, w_i) = \frac{N(w_{i-1}, w_i)}{N}$$

$$P(S) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_2)...P(w_n \mid w_{n-1})$$

- 基于上面的假设和公式计算 $P(S)$

统计语言模型实例

条件概率公式

$$P(A|B) = \frac{P(AB)}{P(B)}$$

- 如何估计条件概率

$$p(w_i \mid w_{i-1}) = \frac{p(w_{i-1}, w_i)}{p(w_{i-1})}$$

- $p(w_{i-1})$和 $p(w_{i-1}, w_i)$ 的计算 基于语料库

$$p(w_{i-1}) = \frac{N(w_{i-1})}{N} \qquad p(w_{i-1}, w_i) = \frac{N(w_{i-1}, w_i)}{N}$$

$$P(S) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_2)...P(w_n \mid w_{n-1})$$

- 基于上面的假设和公式计算 $P(S)$

统计语言模型

- 语料库如下：
  - doc1=‘我们/一起/出去/玩’
  - doc2=‘我们/出去/爬山’
  - doc3=‘他们/一起/出去/玩’
- 例子：
  - S=‘他们/一起/出去/爬山’，求P(S)

$$P(S)=P(w_1)P(w_2|w_1)P(w_3|w_2)...P(w_n|w_{n-1})$$

$$P(S)=\mathrm{P}(他们)\mathrm{P}(一起|他们)\mathrm{P}(出去|一起)\mathrm{P}(爬山|出去)$$

$$p(他们)=\frac{N(他们)}{N}=\frac{1}{11}$$

$$p(一起|他们)=\frac{N(一起,他们)}{N（他们）}=\frac{1}{1}$$

$$p(出去|一起)=\frac{N(出去,一起)}{N（一起）}=\frac{2}{2}$$

$$p(爬山|出去)=\frac{N(爬山,出去)}{N（出去）}=\frac{1}{3}$$

- S=‘一起/他们/出去/爬山’，求P(S)

$$p(出去|爬山)$$

再谈分词

- 基于统计模型的分词
- 假设一个句子S可以有以下几种分词方法，

$A_1, A_2, A_3, ..., A_k$

$B_1, B_2, B_3, ..., B_m$

$C_1, C_2, C_3, ..., C_n$

发展中国家=> 发展/中国/家 发展/中/国家
发/展中/国家

- 如果 $A_1, A_2, A_3, ..., A_k$ 是最好的分词方法，那么必须满足：

$$P(A_1, A_2, A_3, ..., A_k) > P(B_1, B_2, B_3, ..., B_m)$$

$$P(A_1, A_2, A_3, ..., A_k) > P(C_1, C_2, C_3, ..., C_n)$$

再谈分词

- 基于统计模型的分词
- 计算 $P(A_1, A_2, A_3, ..., A_k)$ $P(B_1, B_2, B_3, ..., B_m)$
- 利用统计语言模型即可解决

$$P(A_1, A_2, A_3, ..., A_k) =$$

$$P(B_1, B_2, B_3, ..., B_m) =$$

$$P(C_1, C_2, C_3, ..., C_n) =$$

tf-idf

- 什么是tf-idf
  - tf-idf评估一个词对于一个文件的重要程度。
  - 词的重要程度随着它在该文件中出现的次数增加而增加，随着它在语料库中出现的频率成反比下降。　的 了 我
  - 计算公式：tf-idf=tf*idf
- 词频（Term Frequency，TF）
  - 如果某个词很重要，它应该在这篇文章中多次出现。
    - TF = 某个词在文章中的出现次数
  - 规范化
    - TF = 某个词在文章中的出现次数 / 文章总词数
  - 词频可以作为文档的特征，用于表示文档，直接用于分类

文档的词频表示方法

- 例如
  - document1：I come to China to travel
  - document2：I like to travel in china
  - document3：I like tea
- 第一步：统计所有文档中出现的所有单词，得到词典
  - I,come,to,china,travel,like,in,tea 词典里共8个单词（是否需要去停用词？）
- 第二步：计算词典中的每个词在每个文档中出现的次数（词频）
  - document1: [1,1,2,1,1,0,0,0]
  - document2: [1,0,1,1,1,1,1,0]
  - document3: [1,0,0,0,0,1,0,1]
  - 每个文档向量维度和词典维度一致
- **可使用上面向量直接用于文本分类**

用词频表示文本的特点：
词典中每个单词权重一样，需要考虑到每个单词的权重

idf

- 什么是idf
  - IDF是一个词语重要性的度量，即评价一个词语对于整个语料库的重要性。
- 例如
  - document1：I come to China to travel
  - document2：I like to travel in china
  - document3：I like tea
- 对于document1，to和travel哪个重要？
  - to词频为2，travel词频为1
  - 在很多文档中出现的词，权重较低
- 逆文档频率（Inverse Document Frequency，IDF）
  - 逆文档频率（IDF）= log（语料库的文档总数/包含该词的文档总数+1）
  - 作用：表示词典中每个词的权重

tf-idf code实例

- document1：I come to China to travel
- document2：I like to travel in china
- document3：I like tea
- 代码计算的词频如右上表
- 代码计算的tf-idf如右下表
- 词频和tf-idf都可以用来表示文本

	china	come	in	like	tea	to	travel
document1	1	1	0	0	0	2	1
document2	1	0	1	1	0	1	1
document3	0	0	0	1	1	0	0

	china	come	in	like	tea	to	travel
document1	0.359	0.472	0	0	0	0.719	0.359
document2	0.417	0	0.549	0.417	0	0.417	0.417
document3	0	0	0	0.605	0.795	0	0

文档表示

- 词频向量表示
- tf-idf向量表示
- 机器学习算法进行文本分类

	china	come	in	like	tea	to	travel
document1	1	1	0	0	0	2	1
document2	1	0	1	1	0	1	1
document3	0	0	0	1	1	0	0

	china	come	in	like	tea	to	travel
document1	0.359	0.472	0	0	0	0.719	0.359
document2	0.417	0	0.549	0.417	0	0.417	0.417
document3	0	0	0	0.605	0.795	0	0


开课吧
kaikeba.com
慧 科 旗 下 企 业