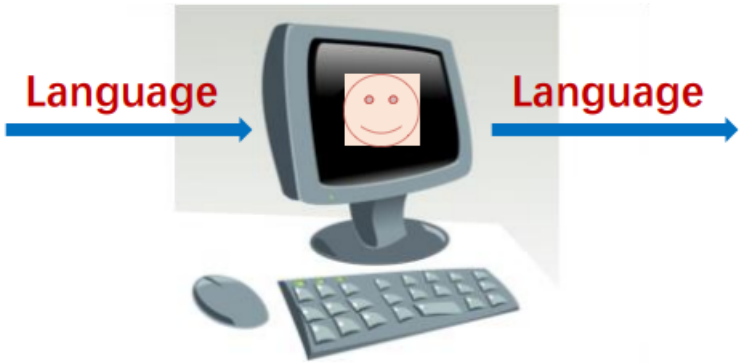
Lecture13 自然语言处理

1. NLP 介绍

什么是 NLP

自然语言处理（NLP）是计算机科学、人工智能和计算语言学的一个领域，主要研究计算机和人类语言之间的相互作用

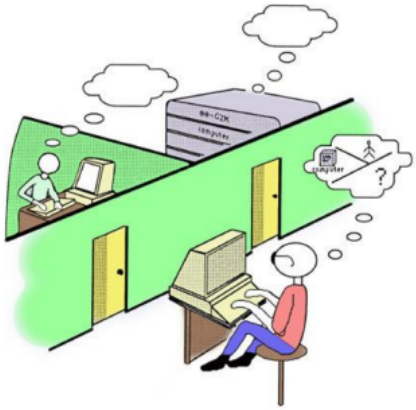
- 自然语言处理（NLP）是一个非常活跃和有吸引力的领域
- 我们的大多数在线活动都是基于文本的
- 电子邮件、博客、新闻搜索结果、评论、社交媒体、医疗报告、课程内容等
- 使用自然语言与计算机交流一直是一个梦想



- 理解语言
- 生成语言

Acting Humanly

图灵测试（Alan Turing 1950）：如果计算机能愚弄人类审讯者，那么它就能通过智力测试



NLP 的应用

- **语音识别 Speech recognition**
  - 虚拟助手：Siri(Apple), Echo(Amazon), Cortana(Microsoft)
  - 利用深度神经网络处理语音识别和自然语言理解
- **机器翻译 Machine Translation**
  - 经历了起起落落
  - 今天，统计机器翻译利用了大量可用的翻译语料库
  - 虽然还有改进的空间，但机器翻译已经取得了显著的进展
- **信息提取 Information Extraction**
  - 自动从非结构化或半结构化文本中提取结构化信息
- **文本总结 Text Summarization**
- **对话系统 Dialog Systems**
- **情感分析 Sentiment Analysis**

NLP 的困难

NLP 是 AI 领域中最难的问题之一 –– 人类的语言太复杂了

- 歧义 Ambiguity
- 指代 Anaphora
- 换喻 Metonymy
- 比喻 Metaphor
- 模糊，话语结构，自动纠错...

2. 文本分类问题

介绍

- 了解感兴趣的新闻文章
- 学会根据主题对网页进行分类
- **朴素贝叶斯**是最有效的算法之一
- 我们应该使用哪些属性来表示文本文档?

使用朴素贝叶斯进行分类

给定一个文档（语料库），为文档中的每个单词位置定义一个属性，属性的值是该位置上的英文单词

为了减少需要估计的概率数，除了朴素贝叶斯的独立假设外，我们假设：给定单词 w_k 出现的概率与单词在文本中的位置无关，即

$$p(x_1 = w_k|c_j), p(x_2 = w_k|c_j)$$

被化简为

$$p(w_k|c_j)$$

当然，我们需要对概率进行修正

$$p(w_k|c_j) = \frac{n_k + 1}{n_j + |Vocabulary|}$$

- n_j：对于样本 c_j 来说，单词出现的数量和
- n_k：在这 n_j 个单词位置上，单词 w_k 出现的次数

示例

对于表示 Radio 和表示 TV 的句子进行分类

TV

- **TV** programs are not interesting **TV** is annoying.
- Kids like **TV**.
- We receive **TV** by **radio** waves.

Radio

- It is interesting to listen to the **radio**.
- On the waves, kids programs are rare.
- The kids listen to the **radio**; it is rare!

Vocabulary

- V = {TV, program, interesting, kids, radio, wave, listen, rare}

我们计算概率

$$p(c_{TV}) = \frac{3}{6} = 0.5 \;\; p(c_{Radio}) = \frac{3}{6} = 0.5$$

$$n_{TV} = 9 \;\; n_{Radio} = 11$$

- 在 TV 分类中，Vocabulary 中的单词一共出现了 9 次
- 在 Radio 分类中，Vocabulary 中的单词一共出现了 11 次

$w \in \mathcal{V}$	Class "TV"			Class "Radio"		
	n_{TV}	n_w	$p(w\|C_{TV})$	n_{Radio}	n_w	$p(w\|C_{radio})$
TV	9	4	(4+1)/(9+8)	11	0	1/(11+8)
program	9	1	(1+1)/(9+8)	11	1	2/(11+8)
interesting	9	1	(1+1)/(9+8)	11	1	2/(11+8)
kids	9	1	(1+1)/(9+8)	11	2	3/(11+8)
radio	9	1	(1+1)/(9+8)	11	2	3/(11+8)
wave	9	1	(1+1)/(9+8)	11	1	2/(11+8)
listen	9	0	(0+1)/(9+8)	11	2	3/(11+8)
rare	9	0	(0+1)/(9+8)	11	2	3/(11+8)

如果有一个新的句子

- Some **kids** think watching **TV** is **interesting**

预测它属于 TV 类的概率为

$$p(c_{TV}) \cdot p(w_{TV}|c_{TV}) \cdot p(w_{interesting}|c_{TV}) \cdot p(w_{kids}|c_{TV}) = 0.5 \times \frac{5}{17} \times \frac{2}{17} \times \frac{2}{17}$$

预测它属于 Radio 类的概率为

$$p(c_{Radio}) \cdot p(w_{TV}|c_{Radio}) \cdot p(w_{interesting}|c_{Radio}) \cdot p(w_{kids}|c_{Radio}) = 0.5 \times \frac{1}{19} \times \frac{2}{19} \times \frac{3}{19}$$

可以看出，预测结果应该将该句子划分为 TV 类

3. 语言模型

介绍

- 我们看到语言是复杂的，没有单一的意思，我们在语法上有分歧，也没有一组明确的句子
- 我们讨论的不是一个句子的单一意义，而是意义上的**概率分布**
- 语言模型是语言的近似值
- 目的：建模自然的语言

模型构建

例如，我们有一个前面的文本 `Did you call your..`

- 如何推测下一个单词是什么
  - 可能的接在后面的单词有：mother, doctor, child...
  - 不太可能接在后面的单词有：dinosaur, oven...
- 对于任何单词 w 估计 $P(w|Did\ \ you\ \ call\ \ your...)$

建立一个概率语言模型

- 下一个可能单词的概率

    - P(mother| Did you call your…)
    - P(dinosaur| Did you call your…)
    - P(doctor| Did you call your…)
- 一个完整句子（单词序列）的概率
    - P(Open your book on page six)
    - P(Open your book on page six)
- 在一个大的语料库中估计 P(page| open your book on)
    - P(page| open your book on) = count(open your book on page)/ count(open your book on)
- 在一个大的语料库中估计 P(open your book on page)
    - P(open your book on page) = count(open your book on page)/ count(sentences of 5 words)
- 语料库必须非常非常大

N-gram 模型

- 问题：如何计算联合概率 $P(w_1, w_2, \ldots, w_n)$
- 解：利用概率的链式法则分解联合概率
  $P\left(w_1, \cdots, w_n\right)=p\left(w_1\right) P\left(w_2 \mid w_1\right) P\left(w_3 \mid w_1, w_2\right) \cdots P\left(w_n \mid w_1 \cdots w_{n-1}\right)$

$$P\left(w_1, \cdots, w_n\right)=\prod_{k=1}^{n} P\left(w_k \mid w_1 \cdots w_{k-1}\right)$$

- 问题是，使用整个链的计算数据太少了，需要语料库非常大
- 不用整个链，用最后一个词近似
- 使用马尔可夫假设 Markov assumption，用 $P(w_n|w_{n-1})$ 来近似 $P(w_n|w_1, \ldots, w_{n-1})$
    - 例如 P(page|on)
- Trigram 模型：看最近的前两个词
- N-gram 模型：看最近的 n-1 个词

N-gram 模型

$$P\left(w_n \mid w_1 \cdots w_{n-1}\right) \approx P\left(w_n \mid w_{n-N+1} \cdots w_{n-1}\right)$$

$$P\left(w_n \mid w_{n-N+1} \cdots w_{n-1}\right)=\frac{\operatorname{count}\left(w_{n-N+1} \cdots w_{n-1} w_n\right)}{\operatorname{count}\left(w_{n-N+1} \cdots w_{n-1}\right)}$$

Bigram 模型

$$P\left(w_1, \cdots, w_n\right) \approx \prod_{k=1}^{n} P\left(w_k \mid w_{k-1}\right)$$

$$P\left(w_n \mid w_{n-1}\right)=\frac{\operatorname{count}\left(w_{n-1} w_n\right)}{\operatorname{count}\left(w_{n-1}\right)}$$

示例

使用 Bigram 模型，假设有下面三个样本

1. * I love cheese STOP
2. * Cheese and crackers are delicious STOP
3. * I prefer swiss cheese STOP

$$P(I|*)=\frac{2+1}{3+4} \quad P(eat|I)=\frac{0+1}{2+4} \quad P(cheese|eat)=\frac{0+1}{0+4} \quad P(STOP|cheese)=\frac{2+1}{3+4}$$

$$P(*I\ eat\ cheese\ STOP)=P(I|*)P(eat|I)P(cheese|eat)P(STOP|cheese)$$

应用

语言模型可以运用到很多 NLP 应用中

- 拼写矫正
- 统计模型翻译
- 收集信息
- 语音识别
- 语言识别