Center of Intelligence Science and Technology

词形态分析
Lexical Morphological Analysis - 2

王小捷
智能科学与技术中心
北京邮电大学

大纲

- 汉语切分
- FMM算法
- n元语言模型
- n元语言模型的平滑技术
- n元语言模型的评估

- 英语词法分析
  - Tokenization：词边界分析
  - Lemmatization：词结构分析
    - FSA+FST
- 汉语词法分析
  - Tokenization：词边界分析
  - Lemmatization：词结构分析

汉语词法分析

- Tokenization: Segmentation(切分)
  - 我们/在/学习

- Lemmatization
  - 生物学, 机械化， 初一， …
  - Other example?

- 汉语切分
  - 我们/在/学习
  - 工人/动用/了/两辆/吊车/和/一辆/货车
  - 最终/确定/的/画像/版本/
  - 答案/并/不/神秘
  - 这个/必然性/的/实现/却/不是/自然/的
  - …

- 汉语切分的两个主要问题
  - 边界歧义(Boundary Ambiguity)
    - 总是先吃苹果然后吃饭
    - 我家门前有条水沟很难过
  - 未登陆词

    (Out of Vocabulary:OOV)(Unknow words)
    - 深港通之后还有沪港通
    - 奥巴马访问乌克兰

- 边界歧义(Boundary Ambiguity)
  - 交集歧义(Overlap ambiguity: OA)
  - 组合歧义(Combination ambiguity: CA)

- 交集歧义(Overlap ambiguity: OA)
  - 定义: 串“XJY”中, 如果XJ 和 JY都是词典中的词, 那么对该串就有两种可能的切分: “XJ/Y/ ”and “X/JY/ ”，则XJY 称为OA串。
  - 交集长度：连续出现的OA串数量(处于交叉位置的字个数)
  - L=1：的确定
    - 毛领导地位的/确定/是在遵义会议上
    - 这件事一时/的确/定不下来 other example?
  - L=2：结合成分==结合成 合成分
  - L=3：为人民工作==为人民 人民工 民工作
  - L=4：中国产品质量==中国产 国产品 产品质 品质量
  - L=6：努力学习语法规则

- 组合歧义(Combination ambiguity: CA)
  - 定义：对于串XY, 如果X、Y和XY都是词典中的词, 那么该串就有两种切分: “AB/ ” 和“A/B/ ”，则AB称为CA串
  - For example：一起
    - 和孩子们/一起/活动
    - 一/起/特大火灾

- 混合歧义
  - 一个串中同时包含OA和CA
    - 在这里，有些/人/才能/发挥的很好，有的人则没有发挥好
    - 这些/人才/能/对这里的发展起到不小的作用
    - 这样的/人/才/能/经受住考验

- 真实歧义(Realized ambiguity)
  - 在真实语料中出现不同的切分
  - 例: “地面积”
    - 地面/积 /了 /水
    - 这块 /地/面积 / 很 /大
- 伪歧义(Pseudo ambiguity)
  - 仅在理论上可能会出现歧义，在真实语料中只会出现一种情况
  - 例: “挨批评”、“市政府”

未登录词 (OOV)

- 派生词
  - 晕晕乎乎 (AB→AABB)、自动化(N+词缀) …
  - 老马高兴高兴地玩遍了桂林的山山水水
- 命名实体(Named Entities)
  - 人名：李白、蒙博托·塞塞·塞科·库库·恩关杜·瓦·扎·邦加…
  - 地名：北大荒、塞瓦斯托波尔、…
  - 组织机构名：谷歌、中共中央书记处、…
  - 商标名、商品名、电影名、菜名、歌名……
- 新词(New word)
  - 博客、微博、给力、喜大普奔…

歧义和OOV的混合

- 敦煌市长孙玉龙 （命名实体+歧义）
  - 敦煌市长/孙玉龙/说
  - 敦煌市/长孙/玉龙/说
- 超女 （新词+歧义）
  - 5名女将在短短2个半小时内先后14次赶超/女举58公斤级世界纪录
  - 粉丝们欢呼着去赶/超女的见面会

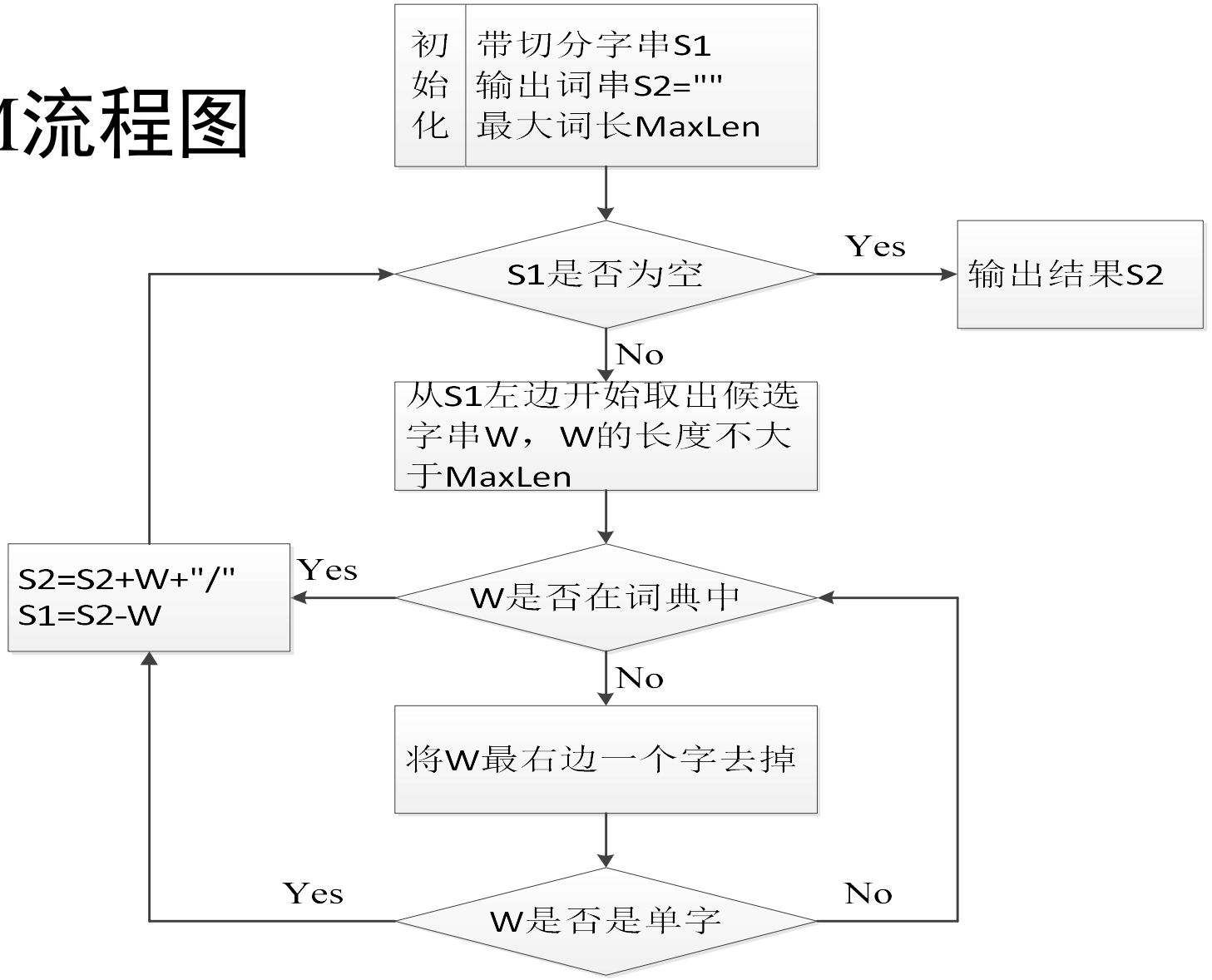
- 如何切分?
  - 切分知识有哪些？
  - 如何表示？
  - 如何使用？
- 先介绍一个最基础的，只使用词典知识

大纲

- 汉语切分
- FMM算法
- n元语言模型
- n元语言模型的平滑技术
- n元语言模型的评估

- 从一个Naïve的切分算法开始
- 前向最大匹配算法: Forward Maximum Match(FMM)
  - 切分的基本知识：词典及其使用

■FMM流程图


初始化
带切分字串S1
输出词串S2=""
最大词长MaxLen
S1是否为空
Yes
输出结果S2
No
从S1左边开始取出候选字串W，W的长度不大于MaxLen
W是否在词典中
Yes
S2=S2+W+"/"
S1=S2-W
No
将W最右边一个字去掉
W是否是单字
Yes
No

前向最大匹配切分示例

- 词典
  - 语言学
  - 重要
- 算法参数：最大匹配长度
  - MaxLen = 3
- 任务：切分句子“语言学很重要"
- 初始输入串：S1=“语言学很重要"
- 初始输出串：S2=“”

一步一步执行例

- S2=“”； S1不为空，从S1左边取出MaxLen=3个字W=“语言学”；
- 查词表，“语言学”在词表中，将W加入到S2中，S2=“语言学/ ”，并将W从S1中去掉，此时S1 =“很重要”；
- S1不为空，于是从S1左边取出3个字W=“很重要“；
- 查词表，W不在词表中，将W最右边一个字去掉，得到W=“很重”；
- 查词表，W不在词表中，将W最右边一个字去掉，得到W=“很”；
- 查词表，W不在词表中，但只有一个字，将W加入到S2中，S2=“语言学/很/”，并将W从S1中去掉，此时S1 =“重要”；
- S1不为空，于是从S1左边取出最后2个字W=“重要“；
- 查词表，W在词表中，将W加入到S2，S2=“语言学/很/重要/”，并将W从S1中去掉，此时S1 =“”；
- S1为空，则S2为切分结果，切分结束

- 影响算法性能的因素
  - MaxLen: 应该设多大？
  - MaxLen较小：
    - 中华人民共和国/成立/了/。
  - MaxLen较大：
    - 他/是/在/看/海/。

- 影响算法性能的因素
  - 词典：应该包含哪些词？
    - 普通词：

	语言学很重要	语言学习很重要
语言学	√	×
语言	×	√

    - 专名：李鹏

评估1-评测指标

- 给定评测集(标准答案gold)
  - 标准：本报 讯 春节 临近
  - 输出1：本 报 讯 春节 临近
  - 输出2：本报讯 春节 临近
  - 输出3：本 报讯 春节 临近
    - P： 正确切分词数/全部切分词数
    - R： 正确切分词数/全部词数(gold)
    - F1： 2*P*R/(P+R)

评估2–任务难度 （数据相关）

- Baseline(基线):
  - What is？
    - performance achieved by the simplest method
  - Why？
    - how difficult
    - how bad
- Ceiling
  - Inter-agreement

- 类似的：后向最大匹配算法
  - Backward Maximum Match(BMM)
  - 语言学很重要

FMM处理歧义和OOV的情况

- OA：XJY
  - 总是切分成 XJ/Y
  - 而在BMM: 总是切分为X/JY
- CA：XY
  - 总是切分为 XY 为一个词
- OOV
  - 不能处理

扩展FMM以处理歧义

- 指定一些规则的方法
  - 例如:
    - IF W = "个人", W_{Left} = 数词 THEN W = "个/ 人/" ENDIF
  - 问题:
  - 1 有多少这种规则要指定?
  - 2 规则的例外
    - 中国第一个人网站站长高春辉的今天
    - 美一个人网站被迫删除其反同性恋内容

总结

- 基于规则的切分方法
  - 知识源：词表
  - 知识表示：符号、规则
  - 知识应用：搜索匹配
- 性能
  - 没能提供高质量的切分
- 有其价值
  - 简单、快速
  - FMM+BMM可以用来发现OA

- 问题
  - 面对歧义选择固定，没有随上下文变化而进行不同切分选择的消歧能力
  - OA: XJ/Y
  - CA: /XY/
- 一起 ？ 一/起

- 问题分析
  - 在上述方法中，切分知识是语言学家依据对语言材料的思考和总结得到的
  - 1 专家人工获取：费时费力
  - 2 词表：无上下文信息，不能使用上下文
  - 3 规则：0-1规则，存在例外

- 因此，考虑自动方法、获取具有上下文信息、且具有柔性的知识？

- 自动获取切分知识
  - 知识源：大规模已切分文本语料(Corpus)
    - 真实文本中➔ 带有上下文
    - 大规模 ➔ 知识足够多、可靠
    - 已切分 ➔ 所以应当包含切分知识.
- 知识如何表示？
- 知识如何获取?

大纲

- 汉语切分
- FMM算法
- n元语言模型
- n元语言模型的平滑技术
- n元语言模型的评估

- n-gram语言模型提供了一种
  - 从语言数据中直接获取的、反映词间上下文搭配的概率知识模型

n-gram语言模型 (n-gram language model)

- 词之间的 n-1阶依赖关系：
- $p(w|w_1,w_2,\ldots,w_{n-1})$

- n-gram：连续出现的n个词(语言单元)
- n-gram语言模型：连续n-1个词后第n个词的概率

一元语言模型(Unigram model)

■P($w|w_1$ ….. w_{n-1}) 中n=1，即：P(w)，任意词w出现的概率

■基于语料的MLE估计 $$p(w) = \frac{C(w)}{\sum_{j=1}^{|V|} C(w_j)}$$

■Unigram LM：($p(w_1),\ldots, p(w_{|V|})$)

■w 称为unigram

二元语言模型(Bigram model)

■P($w|w_1$ ….. w_{n-1}) 中n=2，即：P($w|w_1$)，任意一个词在另一个词之后出现的概率

■基于语料的MLE估计 $$P(w_i \mid w_j) = \frac{C(w_j, w_i)}{\sum_{k=1}^{N} C(w_j, w_k)}$$

■Bigram LM：

$$\begin{pmatrix} p(w_1 \mid w_1) & \cdots & p(w_{|V|} \mid w_1) \\ \vdots & \ddots & \vdots \\ p(w_1 \mid w_{|V|}) & \cdots & p(w_{|V|} \mid w_{|V|}) \end{pmatrix}$$

■(w_j,w_i)称为bigram

三元语言模型(Trigram model)

■P($w|w_1$ ….. w_{n-1}) 中n=3，即：P($w|w_1,w_2$)

■基于语料的MLE估计

$$P(w_i \mid w_j, w_k) = \frac{C(w_j, w_k, w_i)}{\sum_{l=1}^{N} C(w_j, w_k, w_l)}$$

■Trigram LM：

■(w_i, w_j, w_k)称为trigram

■一般地n-gram 语言模型的参数估计：

$$P(w_n \mid w_1 \ldots w_{n-1}) = \frac{C(w_1 \ldots w_{n-1}, w_n)}{C(w_1 \ldots w_{n-1})}$$

■其中 $C(w_1 \ldots w_{n-1}) = \sum_{j \in V} C(w_1 \ldots w_{n-1}, w_j)$

- n-gram模型的简单应用:
  - 句子概率估计

$$P(W_1..W_m)$$

  - 先链式法则

$$P(w_1..w_m) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1, w_2)...P(w_m \mid w_1..w_{m-1})$$

  - 再利用n-gram模型(bigram为例)

$$P(w_1..w_m) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_2)...P(w_m \mid w_{m-1})$$

例:基于Berkeley Restaurant Project 语料来构建Bigram LM

- BERP语料例：
  - I’m looking for Cantonese food
  - I’d like to eat dinner someplace nearby
  - Tell me about Chez Panisse
  - I’m looking for a good place to eat breakfast
  - …

■ Bigram 计数

	I	want	to	eat	Chinese	food	lunch	…
I	8	1087	0	13	0	0	0	…
want	3	0	786	0	6	8	6	…
to	3	0	12	860	3	0	12	…
eat	0	0	0	0	19	2	52	…
Chinese	2	0	17	0	0	120	1	…
food	19	0	0	0	0	0	0	…
lunch	4	0	0	0	0	1	0	…
…	…	…	…	…	…	…	…	…

- 计算条件概率 $$P(w_i \mid w_j) = \frac{C(w_j, w_i)}{\sum_{k=1}^{N} C(w_j, w_k)}$$

- Example of P(I|I):
  - C(I,I): 8
  - $C(I)=Sum_i(C(I,w_i)= 8 + 1087 + 13 \ldots. = 3437$
  - P(I|I) = 8 / 3437 = .0023

$$P(I \mid I) = \frac{C(I, I)}{\sum_{i=1}^{N} C(I, w_i)}$$

■ Bigram probabilities

	I	want	to	eat	Chinese	food	lunch
I	.0023	.32	0	.0038	0	0	0
want	.0025	0	.65	0	.0049	.0066	.0049
to	.00092	0	.0031	.26	.00092	0	.0037
eat	0	0	.0021	0	.020	.0021	.055
Chinese	.0094	0	0	0	0	.56	.0047
food	.013	0	.011	0	0	0	0
lunch	.0087	0	0	0	0	.0022	0

利用这些概率计算句子概率

■P(I want to eat Chinese food)

■≈ P(I|”sentence start”) *
P(want|I) *
P(to|want) *
P(eat|to) *
P(Chinese|eat) *
P(food|Chinese)

=.25 * .32 * .65 * .26 * .002 * .60

= .000016

大纲

- 汉语切分
- FMM算法
- n元语言模型
- n元语言模型的平滑技术
- n元语言模型的评估

n-gram 模型中的参数分析

■n-gram: n多少为好？

■Probabilities

$$P(w_m \mid w_1..w_{m-1}) = \frac{C(w_1..w_{m-1}, w_m)}{C(w_1..w_{m-1})}$$

利用N-gram生成句子：语料源于莎士比亚著作

Unigram	• To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have • Every enter now severally so, let • Hill he late speaks; or! a more to leg less first you enter • Are where exeunt and sighs have rise excellency took of.. Sleep knave we. near; vile like
Bigram	• What means, sir. I confess she? then all sorts, he is trim, captain. •Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. •What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman? •Enter Menenius, if it so many good direction found'st thou art a strong upon command of fear not a liberal largess given away, Falstaff! Exeunt
Trigram	• Sweet prince, Falstaff shall die. Harry of Monmouth's grave. • This shall forbid it should be branded, if renown made it empty. • Indeed the duke; and had a very good friend. • Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.
Quadrigram	• King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in; • Will you not tell me who I am? • It cannot be but so. • Indeed the short and the long. Marry, 'tis a noble Lepidus.

- 结论:
  - 较大的N带来较好的句子
- 但是
  - 更大的N导致更大的参数规模

■以词表大小=10^4为例

N	N-gram模型参数	N-gram模型参数规模
1	p(.)	10^4
2	p(.|.)	10^8
3	p(.|..)	10^{12}
4	p(.|…)	10^{16}
…	…	…
n	p(.|……)	10^{4n}
…		

- 用于估计N-gram的语料
- 以语料含10^8词为例，词表大小=10^4

N	语料中N-gram的总数目	N-gram参数数目	平均每个参数所用N-gram的数目
1	10^8	10^4	10^4
2	10^8-1	10^8	<1
3	10^8-2	10^{12}	<10^{-4}
4	10^8-3	10^{16}	<10^{-8}
n	10^8-n+1	10^{4n}	<10^{-4n-8}

- 随N增大，越来越多的N-gram没在语料中出现(0出现次数) ➔ 越来越多的N-gram概率=0

- n-gram LM中的0概率越来越多

$$P(w_m \mid w_1..w_{m-1}) = \frac{C(w_1..w_{m-1}, w_m)}{C(w_1..w_{m-1})}$$

- 称之为数据稀疏！
- 导致LM应用中的问题
- 例如：

■计算一个句子的概率BERP语料

■P (Does a Chinese eat this food) ≈ P(Does|S) *

P(a|does) * P(chinese|a) * P(eat|chinese) * P(this|eat)

* P(food|this) =.25 * .32 * .65 * .0 * .002 * .60 = 0

- 连乘：只要有一个为零，整句概率为零

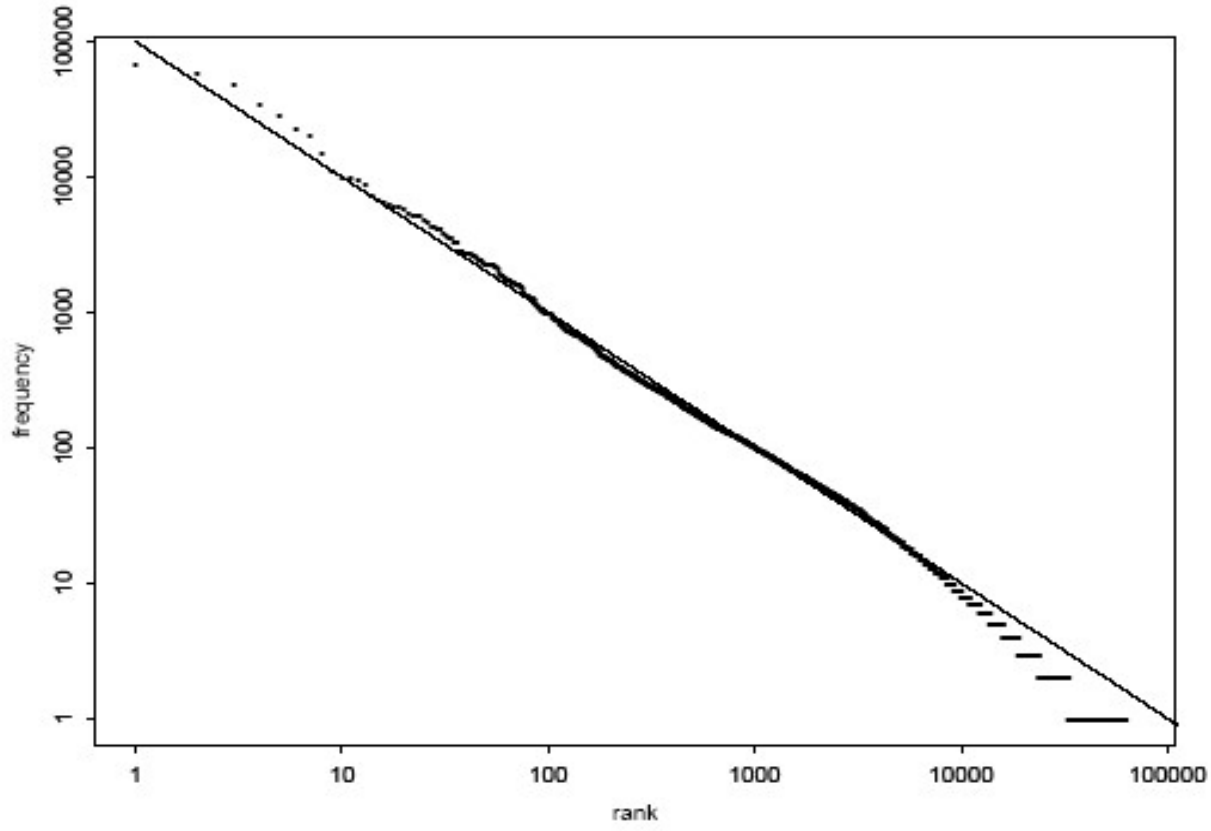
- 0出现次数的其他问题:
  - $C(w_j)/C(.)$; when $C(.)=0$
    - 不能计算!
- 还有：低出现次数
  - $C(w_j)/C(.)$; 当 $C(.)$ 很小 (1,2,etc.)
    - 估计不可靠
    - MLE: 大样本

数据稀疏的程度

- 一些数据:
  - Bahal(1983) : 150M IBM patent corpus, train 3-gram, then use to analyze anther corpus with same source. 23% of 3-gram do not occur in training corpus.
  - Essen and Steinbiss(1992): LOB corpus (M) 75% as training, 25% as test, 12% of 2-gram do not occur in training corpus.
  - Brown and Dellapietra(1992): 366M as training, new test corpus, 14.7% new 3-gram

- 解决方案：
  - 更大规模的语料？

Zipf 律 (Zipf's law): frequency*rank = c



- 更大的语料有帮助，但是不能根本解决问题
- 另外的方法：
- →→

大纲

- 汉语切分
- FMM算法
- n元语言模型
- n元语言模型的平滑技术
- n元语言模型的评估

解决0概率问题

- 平滑 (Smoothing): 重估0出现以及低出现的n-gram的概率
- 回退(Back-off): 用低阶概率来替代高阶
- 插值(Interpolation):综合采用多个n-gram

Laplace 平滑(加1平滑)

- Unigram:
  - 原始MLE $p(w_i) = \frac{c_i}{N}$
    - c_i为词w_i在语料中出现的次数，N为语料token总数
  - 加1： $P_{Laplace} = \frac{c_i + 1}{N + |V|}$
    - $|V|$为词表大小，分母+$|V|$的原因：归一化

Laplace 平滑(加1平滑)

- Unigram:
  - 从折扣的角度看加1平滑:
    - 令 c_i*是c_i的折扣数
    - $$c_i^* = \frac{(c_i + 1)N}{N + |V|}$$
    - 则 $$P_{Laplace} = \frac{c_i^*}{N}$$

分母+|V|:概率空间完备性

Laplace 平滑(加1平滑)

- bigram:
  - 原始MLE $$p(w_i \mid w_{i-1}) = \frac{c(w_{i-1}, w_i)}{c(w_{i-1})}$$
  - 加1： $$P_{Laplace} = \frac{c(w_{i-1}, w_i)+1}{c(w_{i-1})+|V|}$$
  - 从折扣的角度看加1平滑:
    - $c^*()$是$c()$的折扣： $$c_i^*(w_{i-1}, w_i) = \frac{(c(w_{i-1}, w_i)+1)c(w_{i-1})}{c(w_{i-1})+|V|}$$
    - 则 $$P_{Laplace} = \frac{c^*(w_{i-1}, w_i)}{c(w_{i-1})}$$

例:

- Berkeley Restaurant Project语料
  - 9332 个句子
  - 1446 个词
- 关注其中如下8个词之间的Bigram
  - i want to eat chinese food lunch spend

原始计数

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

加1计数

	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	1	2	1	1
spend	2	1	2	1	1	1	1	1

原始计数

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

加1折扣计数

	i	want	to	eat	chinese	food	lunch	spend
i	3.8	527	0.64	6.4	0.64	0.64	0.64	1.9
want	1.2	0.39	238	0.78	2.7	2.7	2.3	0.78
to	1.9	0.63	3.1	430	1.9	0.63	4.4	133
eat	0.34	0.34	1	0.34	5.8	1	15	0.34
chinese	0.2	0.098	0.098	0.098	0.098	8.2	0.2	0.098
food	6.9	0.43	6.9	0.43	0.86	2.2	0.43	0.43
lunch	0.57	0.19	0.19	0.19	0.19	0.38	0.19	0.19
spend	0.32	0.16	0.32	0.16	0.16	0.16	0.16	0.16

原始概率

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

加1概率

	i	want	to	eat	chinese	food	lunch	spend
i	0.0015	0.21	0.00025	0.0025	0.00025	0.00025	0.00025	0.00075
want	0.0013	0.00042	0.26	0.00084	0.0029	0.0029	0.0025	0.00084
to	0.00078	0.00026	0.0013	0.18	0.00078	0.00026	0.0018	0.055
eat	0.00046	0.00046	0.0014	0.00046	0.0078	0.0014	0.02	0.00046
chinese	0.0012	0.00062	0.00062	0.00062	0.00062	0.052	0.0012	0.00062
food	0.0063	0.00039	0.0063	0.00039	0.00079	0.002	0.00039	0.00039
lunch	0.0017	0.00056	0.00056	0.00056	0.00056	0.0011	0.00056	0.00056
spend	0.0012	0.00058	0.0012	0.00058	0.00058	0.00058	0.00058	0.00058

问题

- 加1平滑对原来的概率分配影响很大
  - E.g.: P(to|want) 从 .66 变为 .26!
- 太多概率从数据计得的n-gram上转移走了
  - Church and Gale(1991):46.5% 被转给了0出现n-gram
  - 打折打的太厉害了!

Lidstone(add-δ) 平滑

$$P(w_i \mid w_1 \ldots w_{i-1}) = \frac{N(w_1 \ldots w_i) + \delta}{N(w_1 \ldots w_{i-1}) + \mid V \mid \delta}$$

$$0 \leq \delta \leq 1$$

δ=1: **Laplace 平滑**

Good-Turing 平滑

- 基本思想: (以unigram为例)
  - 为出现n次的unigram分配概率的时候看出现n+1次的unigram的出现次数。
- 出现1次的unigram的数量:

$$N_1 = \sum_{w:count(w)=1} 1$$

- 出现c次的unigram的数量:

$$N_c = \sum_{w:count(w)=c} 1$$

- 折扣:
  - 出现c次的折扣为出现c*次 $c^* = (c+1)\frac{N_{c+1}}{N_c}$
- 则出现c次的词的概率为 $p_c = \frac{c^*}{N} = \frac{(c+1)N_{c+1}}{N_c N}$
- 出现0次的词在折扣后的出现次数：

$$c_0^* = (0+1)\frac{N_{0+1}}{N_0} = \frac{N_1}{N_0}$$

- 则出现c次的词的概率为 $p_0 = \frac{c_0^*}{N} = \frac{N_1}{N_0 N}$

■GT的归一化特性 (完备)

$$\sum_{c=0}^{\infty} n_c p_c = \sum_{c=0}^{\infty} n_c \frac{c^*}{N} = \sum_{c=0}^{\infty} n_c \frac{(c+1)\frac{n_{c+1}}{n_c}}{N}$$

$$= \sum_{c=0}^{\infty} \frac{(c+1)n_{c+1}}{N} = \frac{\sum_{c=0}^{\infty}(c+1)n_{c+1}}{N}$$

$$= \frac{\sum_{c=1}^{\infty} cn_c}{N} = 1$$

■问题1：

$$p_0 = \frac{c_0^*}{N} = \frac{N_1}{N_0 N}$$

N_0？

■问题2：

$$p_c = \frac{c^*}{N} = \frac{(c+1)N_{c+1}}{N_c N}$$

$N_{c+1} = 0$怎么办？

AP Newswire			Berkeley Restaurant		
c (MLE)	N_c	c^* (GT)	c (MLE)	N_c	c^* (GT)
0	74,671,100,000	0.0000270	0	2,081,496	0.002553
1	2,018,046	0.446	1	5315	0.533960
2	449,721	1.26	2	1419	1.357294
3	188,933	2.24	3	642	2.373832
4	105,668	3.24	4	381	4.081365
5	68,379	4.22	5	311	3.781350
6	48,190	5.19	6	196	4.500000

Good-Turing 平滑

- 对于 n-gram 串 W，
  - if its frequency $r > 0$：
$$P(W) = \frac{r^*}{T}$$
  - 否者, 即$r = 0$:
$$P(W) = \frac{1 - \sum_{r=1}^{\infty} n_r \frac{r^*}{T}}{n_0} \approx \frac{n_1}{n_0 T}$$
  - 其中:
$$r^* = \frac{(r+1)n_{r+1}}{n_r}$$

- 其他平滑
  - Jelinek*-Mercer平滑
  - Kneser-Ney平滑
    - 打折中的问题
      - 频次相同的词打折一样，但是Francisco现象：Francisco几乎都出现在San后，与其相同词频的词一样平滑直观上不合适
        - P(Francisco|bite) = P(table|bite)
      - 直观：出现在较多上下文中的词更可能出现在新的上下文中

回退 (Backoff): 基本思想

$$P(w_i \mid w_{i-2}, w_{i-1}) = \begin{cases} P(w_i \mid w_{i-2}, w_{i-1}) & if \quad C(w_{i-2}, w_{i-1}, w_i) > 0 \\ \alpha_1 P(w_i \mid w_{i-1}) & if \quad C(w_{i-2}, w_{i-1}, w_i) = 0 and C(w_{i-1}, w_i) > 0 \\ \alpha_2 P(w_i) & otherwise \end{cases}$$

插值 (Interpolation)：基本思想

$$P(w_n \mid w_{n-2}, w_{n-1}) = \lambda_1 P(w_n) + \lambda_2 P(w_n \mid w_{n-1}) + \lambda_3 P(w_n \mid w_{n-2}, w_{n-1})$$

$0 \le \lambda_i \le 1 \qquad \neq 1,2,3 \qquad \lambda_1 + \lambda_2 + \lambda_3 = 1$

大纲

- 汉语切分
- FMM算法
- n元语言模型
- n元语言模型的平滑技术
- n元语言模型的评估

N-gram Model 评估

- 评估方法
  - 外部评估(Extrinsic evaluation)
    - 帮助应用提高性能
  - 内部评估(Intrinsic evaluation)
    - Perplexity(PP，复杂度、困惑度…)
      - 以bigram为例，测试集为 $W=(w_1,\ldots w_N)$:

$$PP(W) = P(W)^{-\frac{1}{N}} = \left[\prod_{i=1}^{N} P(w_i \mid w_{i-1})\right]^{-\frac{1}{N}}$$

      - P(W) 的几何平均值 的倒数
    - PP值大好？还是小好？

$$PP(W) = P(W)^{-\frac{1}{N}} = \left[\prod_{i=1}^{N} P(w_i \mid w_{i-1})\right]^{-\frac{1}{N}}$$

- 对于给定的测试语料W
  - PP值越小，所用的LM越好

- 为什么？
  - 解释方式1
    - 因为测试语料是一个正确的句子(集), 因此它的概率越大(也即PP值越小)越好。

- 解释方式2:
  - PP 看成是 **branching factor of a language**:
  - 在一个词后可能接的词的数目。较大的PP意味着更多的不确定性。
  - 例子:
    - Trained on 38 million WSJ words
    - Tested on 1.5 million WSJ words

N-gram order	Uni-	Bi-	Tri-
PP	962	170	109

PP: 与熵的关系

■串$W=w_1,w_2,\ldots w_N$的熵为：

$$H(W) = -\sum q(w_1 \ldots w_N) \log_2 q(w_1 \ldots w_N)$$

■每个词的平均熵为：

$$H(per-word) = -\frac{1}{N} \sum q(w_1 \ldots w_N) \log_2 q(w_1 \ldots w_N)$$

■这种语言的熵为

$$H(L) = -\lim_{N\to\infty} \frac{1}{N} \sum q(w_1 \ldots w_N) \log_2 q(w_1 \ldots w_N)$$

$$= -\lim_{N\to\infty} \frac{1}{N} \log_2 q(w_1 \ldots w_N) \quad \text{(stationary and ergodic L)}$$

■但是 q(.) 未知, 在n-gram模型中，用p来估计q，

■因此，对于充分大的 N，有:

$$H(W) = -\frac{1}{N}\log_2 p(w_1 \ldots w_N)$$

■而：
$$PP(W) = [p(w_1, \ldots w_N)]^{-\frac{1}{N}}$$

■因此：
$$PP(W) = 2^{H(W)}$$

■H越小，则PP越小，反之亦然。

- 基于PP评估注意：
  - 足够大的语料
  - 比较不同的LM要用相同的语料


Center of Intelligence Science and Technology


BEIJING UNIVERSITY OF POSTS AND TELECOMMUNICATIONS
北京邮电大学

谢谢！