

NLP复习

NLP复习

绪论

基本概念

- 定义1-1：语言学(linguistics)
- 定义1-2：语音学(photonetics)
- 定义1-3：计算语言学(Computational Linguistics)
- 定义1-4：自然语言理解(Natural Language Understanding, NLU)
- 定义1-5：自然语言处理(Natural Language Processing, NLP)
- 定义1-6：中文信息处理(Chinese Information Processing)

三个不同的语系

HLT的产生与发展

基本问题和主要困难

- 基本问题之一：形态学(Morphology) 问题
 - 基本问题之二：句法(Syntax) 问题
 - 基本问题之三：语义(Semantics) 问题
 - 基本问题之四：语用学(Pragmatics) 问题
 - 基本问题之五：语音学(Phonetics) 问题
- 困难之一：大量歧义(ambiguity)现象
 - 困难之二：大量未知语言现象

基本研究方法

数学基础

概率论基础

基本概念

信息论基础

语言模型

n 元文法(n-gram)模型

- 应用-1：音字转换问题
- 应用-2：汉语分词问题

数据平滑

基本思想：

- 加1法(Additive smoothing)
- 减值法/折扣法(Discounting)
 - ①Good-Turing 估计
 - ②Back-off (后备/后退)方法
 - ③绝对减值法 (Absolute discounting)
 - ④线性减值法 (Linear discounting)

语言模型的自适应方法：

- (1)基于缓存的语言模型 (cache-based LM)
- (2)基于混合方法的语言模型
- (3)基于最大熵的语言模型

应用到汉语分词

隐马尔可夫模型与条件随机场

隐马尔可夫模型

一般来说，隐马尔可夫模型中包含下面三个问题：

条件随机场

词法分析与词性标注

概述

不同语言的词法分析

英语的形态分析

汉语自动分词

分词与词性标注结果评价方法

评价指标

汉语自动分词基本算法

词性标注方法

- 基于规则的词性标注方法
- 基于统计模型的词性标注方法
- 基于 HMM 的词性标注方法
- 规则和统计方法相结合的词性标注方法
- 基于有限状态变换机的词性标注方法
- 基于神经网络的词性标注方法

语义分析

语义理论

格语法

- 基本观点
- 格的定义
- 格语法的三条基本规则
- 格表
- 格语法描写汉语的局限性

语义网络

- 语义网络的概念

概念依存理论

词义消歧

- 基本方法
- 有监督的词义消歧方法
- 基于词典的词义消歧方法
- 无监督的词义消歧方法

语义角色标注

机器翻译

机器翻译的困难

直接转换法

基于规则的翻译方法

基于中间语言的翻译方法

基于语料库的翻译方法

基于事例的翻译方法

统计翻译方法

噪声信道模型

统计翻译中的三个关键问题：

基于词的机器翻译建模

基于短语的翻译模型

基于最大熵的方法(判别式)

基于短语的翻译模型[Koehn, 2003]

短语划分模型

基于短语的翻译模型的解码算法

柱搜索(**beam search**) [Ney, 1992; Tillmann, 2003]

基于层次化短语的翻译模型

树翻译模型

树到串的翻译模型

树到树的翻译模型

串到树的翻译模型

系统融合

译文评估方法

神经网络机器翻译

单词表示模型

One-hot 编码

分布式表示

word2vec

CBOW

Skip-gram

transfer learning

Model Fine-tuning (labelled source, labelled target)

Multitask Learning (labelled source, labelled target)
Domain-adversarial training (labelled source, unlabelled target)
Zero-shot Learning (labelled source, unlabelled target)
Self-taught learning 和 Self-taught Clustering
前馈神经网络语言模型
编码器-解码器模型
循环神经网络模型
LSTM
GRU
双向模型
注意力机制
GNMT
自注意力机制
transformer
架构
编码器
解码器
神经机器翻译结构优化
感想
情感分析
相关定义
情感分析发展七项关键技术
文本自动摘要
文本摘要的定义
文本摘要分类
文本摘要方法
抽取式摘要
压缩式摘要
理解式摘要
文本摘要评价

绪论

基本概念

定义1-1：语言学(linguistics)

语言学是指对语言的科学的研究。
研究语言的本质、结构和发展规律的科学。
语音和文字是语言的两个基本属性。

定义1-2：语音学(photonetics)

研究人类发音特点，特别是语音发音特点，并提出各种语音描述、分类和转写方法的科学。

定义1-3：计算语言学(Computational Linguistics)

通过建立形式化的计算模型来分析、理解和生成自然语言的学科，是人工智能和语言学的分支学科。

定义1-4：自然语言理解(Natural Language Understanding, NLU)

自然语言理解是探索人类自身语言能力和语言思维活动的本质，研究模仿人类语言认知过程的自然语言处理方法和实现技术的一门学科。

定义1-5: 自然语言处理(Natural Language Processing, NLP)

自然语言处理是研究如何利用计算机技术对语言文本(句子、篇章或话语等)进行处理和加工的一门学科，研究内容包括对词法、句法、语义和语用等信息的识别、分类、提取、转换和生成等各种处理方法和实现技术。

定义1-6: 中文信息处理(Chinese Information Processing)

针对中文的自然语言处理技术。

三个不同的语系

- 屈折语(fusional language/ inflectional language): 用词的形态变化表示语法关系，如英语、法语等。
- 黏着语(agglutinative language): 词内有专门表示语法意义的附加成分，词根或词干与附加成分的结合不紧密，如日语、韩语、土耳其语等。
- 孤立语(analytic language)(分析语, isolating language): 形态变化少，语法关系靠词序和虚词表示，如汉语。
-

HLT的产生与发展

曲折的发展历程：

- 1960S 中期之前：萌芽期
- 1960S 中期到1970S 中后期：步履维艰
- 1970S 中后期到1980S 后期：复苏
- 1980S至2010左右：快速发展
- 2010至今：繁荣时期

信息检索(Information retrieval)

信息检索也称情报检索，就是利用计算机系统从大量文档中找到符合用户需要的相关信息。

自动文摘 (Automatic summarization / Automatic abstracting)

将原文档的主要内容或某方面的信息自动提取出来，并形成原文档的摘要或缩写。

问答系统 (Question-answering system)

通过计算机系统对人提出的问题的理解，利用自动推理等手段，在有关知识资源中自动求解答案并做出相应的回答。

信息过滤(Information filtering)

通过计算机系统自动识别和过滤那些满足特定条件的文档信息。

信息抽取(Information extraction)

从指定文档中或者海量文本中抽取出用户感兴趣的信息。

文档分类(Document categorization)

文档分类也叫文本自动分类(Text categorization / classification) 或信息分类(Information categorization / classification)，其目的就是利用计算机系统对大量的文档按照一定的分类标准(例如，根据主题或内容划分等)实现自动归类。如情感分类(Sentimental classification)

文字编辑和自动校对(Automatic proofreading)

对文字拼写、用词、甚至语法、文档格式等进行自动检查、校对和编排。

应用：排版、印刷和书籍编撰等。

语言教学(Language teaching)

文字识别(Character recognition)

语音识别 (automatic speech recognition, ASR)

将输入语音信号自动转换成书面文字。

基本问题和主要困难

基本问题之一：形态学(Morphology) 问题

研究词(word) 由有意义的基本单位 - 词素(morphemes)的构成问题。

基本问题之二：句法(Syntax) 问题

研究句子结构成分之间的相互关系和组成句子序列的规则。

基本问题之三：语义(Semantics) 问题

研究如何从一个语句中词的意义，以及这些词在该语句中句法结构中的作用来推导出该语句的意义。

基本问题之四：语用学(Pragmatics) 问题

研究在不同上下文中语句的应用，以及上下文对语句理解所产生的影响。语用学最宽泛的定义是研究语义学未

能涵盖的那些意义。

基本问题之五：语音学(Phonetics) 问题

研究语音特性、语音描述、分类及转写方法等

困难之一：大量歧义(ambiguity)现象

- 词法歧义
- 词性歧义
- 结构歧义
- 语义歧义
- 语音歧义
- 多音字及韵律等歧义

困难之二：大量未知语言现象

归纳起来，NLU 所面临的挑战：

- 普遍存在的不确定性：词法、句法、语义、语用和语音各个层面
- 未知语言现象的不可预测性：新的词汇、新的术语、新的语义和语法无处不在
- 始终面临的数据不充分性：有限的语言集合永远无法涵盖开放的语言现象
- 语言知识表达的复杂性：语义知识的模糊性和错综复杂的关联性难以用常规方法有效地描述，为语义计算带来了极大的困难
- 机器翻译中映射单元的不对等性：词法表达不相同、句法结构不一致、语义概念不对等

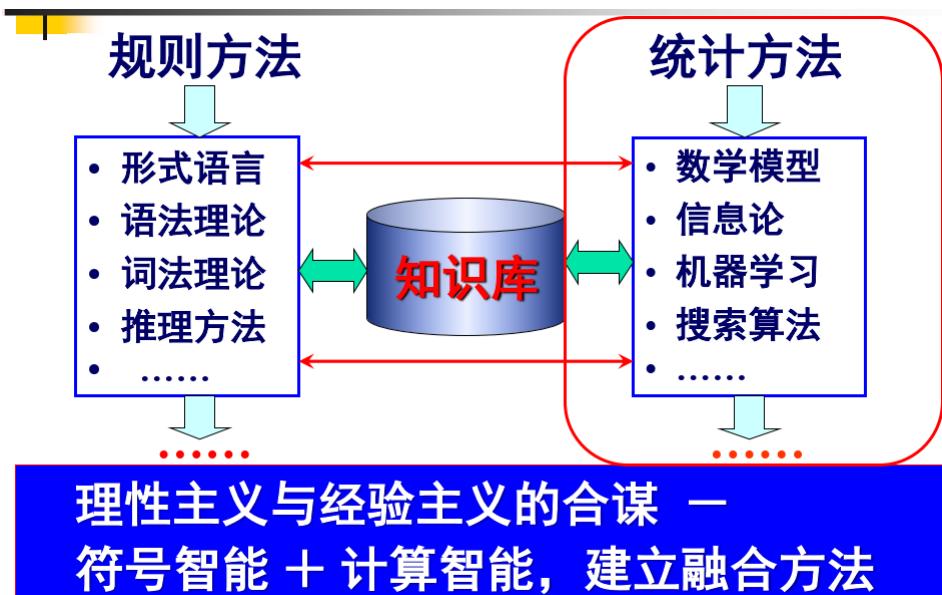
基本研究方法

基于规则的分析方法建立符号处理系统

- 知识库 + 推理系统 → NLP 系统
- 理论基础: Chomsky 的文法理论

基于大规模真实语料(语言数据)建立计算方法

- 语料库 + 统计模型 → NLP 系统
- 理论基础: 统计学、信息论、机器学习



数学基础

概率论基础

基本概念

- 概率(probability)
- 最大似然估计(maximum likelihood estimation)
- 条件概率(conditional probability)
- 全概率公式(full probability)
- 贝叶斯决策理论(Bayesian decision theory)
- 贝叶斯法则(Bayes' theorem)
- 二项式分布(binomial distribution)
- 期望(expectation)
- 方差(variance)

信息论基础

熵(entropy)

如果 X 是一个离散型随机变量, 其概率分布为: $p(x) = p(X = x)$, 事件 $x \sqsubset X$ 。那么事件 x 的信息量 $I(x)$ 定义为:

$$I(x) = -\log_2 p(x)$$

X 的熵表示所有事件信息量的期望： $H(X) = \sum_{x \in X} p(x)I(x) = -\sum_{x \in X} p(x)\log_2 p(x)$, 其中
 $\log 0 = 0$

熵又称为**自信息(self-information)**, 表示信源 X 每发一个符号(不论发什么符号)所提供的平均信息量。

熵也可以被视为描述一个随机变量的不确定性的量。一个随机变量的熵越大，它的不确定性越大。那么，正确估计其值的可能性就越小。

联合熵(joint entropy)

如果 X, Y 是一对离散型随机变量 $X, Y \sim p(x, y)$, X, Y 的联合熵 $H(X, Y)$ 为：

$$H(X, Y) = -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(x, y)$$

联合熵实际上就是描述一对随机变量平均所需要的信息量。

条件熵(conditional entropy)

给定随机变量 X 的情况下，随机变量 Y 的条件熵定义为：

$$\begin{aligned} H(Y | X) &= \sum_{x \in X} p(x)H(Y | X = x) \\ &= \sum_{x \in X} p(x) \left[-\sum_{y \in Y} p(y | x) \log_2 p(y | x) \right] \\ &= -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(y | x) \end{aligned}$$

例题：

例2-4：假设 (X, Y) 服从如下联合概率分布：

$Y \backslash X$	1	2	3	4
1	1/8	1/16	1/32	1/32
2	1/16	1/8	1/32	1/32
3	1/16	1/16	1/16	1/16
4	1/4	0	0	0

请计算 $H(X)$ 、 $H(Y)$ 、 $H(X|Y)$ 、 $H(Y|X)$ 和 $H(X, Y)$ 各是多少？

$Y \backslash X$	1	2	3	4
1	1/8	1/16	1/32	1/32
2	1/16	1/8	1/32	1/32
3	1/16	1/16	1/16	1/16
4	1/4	0	0	0
$p(X)$	1/2	1/4	1/8	1/8

$$\begin{aligned}
 H(X) &= -\sum_{x \in X} p(x) \log_2 p(x) \\
 &= -\left(\frac{1}{2} \times \log_2\left(\frac{1}{2}\right) + \frac{1}{4} \times \log_2\left(\frac{1}{4}\right) + \frac{1}{8} \times \log_2\left(\frac{1}{8}\right) + \frac{1}{8} \times \log_2\left(\frac{1}{8}\right)\right) \\
 &= \frac{7}{4}
 \end{aligned}$$

类似地，可以计算 $H(Y)$ 。

$Y \backslash X$	1	2	3	4	$p(Y)$
1	1/8	1/16	1/32	1/32	1/4
2	1/16	1/8	1/32	1/32	1/4
3	1/16	1/16	1/16	1/16	1/4
4	1/4	0	0	0	1/4

$$H(Y) = -\sum_{y \in Y} p(y) \log_2 p(y) = 2 \text{ (bits)}$$

$Y \backslash X$	1	2	3	4	$p(Y)$
1	1/8	1/16	1/32	1/32	1/4
2	1/16	1/8	1/32	1/32	1/4
3	1/16	1/16	1/16	1/16	1/4
4	1/4	0	0	0	1/4
$p(X)$	1/2	1/4	1/8	1/8	

$$p(x_1 | y_1) = \frac{p(x_1, y_1)}{p(y_1)} = \frac{1}{8} \times \frac{4}{1} = \frac{1}{2} \quad p(x_2 | y_1) = \frac{p(x_2, y_1)}{p(y_1)} = \frac{1}{16} \times \frac{4}{1} = \frac{1}{4}$$

$$p(x_3 | y_1) = \frac{p(x_3, y_1)}{p(y_1)} = \frac{1}{32} \times \frac{4}{1} = \frac{1}{8} \quad p(x_4 | y_1) = \frac{p(x_4, y_1)}{p(y_1)} = \frac{1}{32} \times \frac{4}{1} = \frac{1}{8}$$

.....

$$\begin{aligned}
H(X|Y) &= \sum_{i=1}^4 p(y=i) H(X|Y=i) \\
&= \frac{1}{4} H\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right) + \frac{1}{4} H\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{8}, \frac{1}{8}\right) \\
&\quad + \frac{1}{4} H\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) + \frac{1}{4} H(1,0,0,0) \\
&= \frac{1}{4} \times \frac{7}{4} + \frac{1}{4} \times \frac{7}{4} + \frac{1}{4} \times 2 + \frac{1}{4} \times 0 = \frac{11}{8} \quad (\text{bits})
\end{aligned}$$

$$\begin{aligned}
H(X|Y) &= \sum_{i=1}^4 p(y=i) H(X|Y=i) \\
&= \frac{1}{4} H\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}\right) + \frac{1}{4} H\left(\frac{1}{4}, \frac{1}{2}, \frac{1}{8}, \frac{1}{8}\right) \\
&\quad + \frac{1}{4} H\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right) + \frac{1}{4} H(1,0,0,0) \\
&= \frac{1}{4} \times \frac{7}{4} + \frac{1}{4} \times \frac{7}{4} + \frac{1}{4} \times 2 + \frac{1}{4} \times 0 = \frac{11}{8} \quad (\text{bits})
\end{aligned}$$

类似地, $H(Y|X)=13/8$ (bits), $H(X, Y)=27/8$ (bits)。

可见, $H(Y|X) \neq H(X|Y)$ 。

一般地, 对于一条长度为 n 的信息, 每一个字符或字的熵为:

$$H_{\text{rate}} = \frac{1}{n} H(X_{1:n}) = -\frac{1}{n} \sum_{x_{1:n}} p(x_{1:n}) \log p(x_{1:n})$$

这个数值我们也称为 熵率(entropy rate)。其中, 变量 $X_{1:n}$ 表示随机变量序列 (X_1, \dots, X_n) , $x_{1:n} = (x_1, \dots, x_n)$ 表示随机变量的具体取值。有时将 $x_{1:n}$ 写成 x_1^n 。

相对熵(relative entropy, 或称Kullback-Leibler divergence, KL 距离)

两个概率分布 $p(x)$ 和 $q(x)$ 的相对熵定义为:

$$D(p \| q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

该定义中约定 $0 \log (0/q) = 0$, $p \log (p/0) = \infty$ 。

相对熵常被用以衡量两个随机分布的差距。当两个随机分布相同时, 其相对熵为0。当两个随机分布的差别增加时, 其相对熵也增加。

交叉熵(cross entropy)

$$\begin{aligned}
D(p \parallel q) &= \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} \\
&= \sum_{x \in X} p(x) \log p(x) - \sum_{x \in X} p(x) \log q(x) \\
&= -H(X) + [-\sum_{x \in X} p(x) \log q(x)] \\
&= -H(X) + H(X, q)
\end{aligned}$$

$$\text{令交叉熵 } H(X, q) = -\sum_x p(x) \log q(x)$$

$$H(X, q) = H(X) + D(p \parallel q)$$

交叉熵(cross entropy)

如果一个随机变量 $X \sim p(x)$, $q(x)$ 为用于近似 $p(x)$ 的概率分布, 那么, 随机变量 X 和模型 q 之间的交叉熵 $H(X, q)$ (或写为 $H(p, q)$) 定义为:

$$H(X, q) = -\sum_x p(x) \log q(x)$$

由此, 我们可以根据模型 q 和一个含有大量数据的 L 的样本来计算交叉熵。在设计模型 q 时, 我们的目的是使交叉熵最小, 从而使模型最接近真实的概率分布 $p(x)$ 。

困惑度(perplexity)

在设计语言模型时, 我们通常用困惑度来代替交叉熵衡量语言模型的好坏。给定语言 L 的样本

$l_1^n = l_1 \dots l_n$, L 的困惑度 PP_q 定义为:

$$PP_q = 2^{H(L, q)} \approx 2^{-\frac{1}{n} \log q(l_1^n)} = [q(l_1^n)]^{-\frac{1}{n}}$$

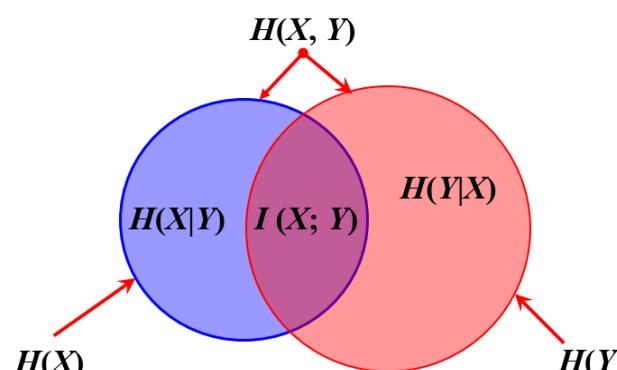
语言模型设计的任务就是寻找困惑度最小的模型, 使其最接近真实的语言。

互信息(mutual information)

如果 $(X, Y) \sim p(x, y)$, X, Y 之间的互信息 $I(X; Y)$ 定义为: $I(X; Y) = H(X) - H(X|Y)$

根据 $H(X)$ 和 $H(X|Y)$ 的定义:

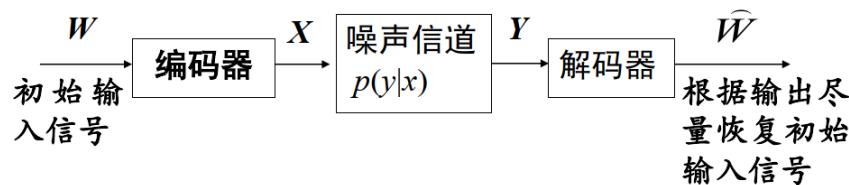
$$\begin{aligned}
H(X) &= -\sum_{x \in X} p(x) \log_2 p(x) \\
H(X|Y) &= -\sum_{x \in X} \sum_{y \in Y} p(x, y) \log_2 p(x|y)
\end{aligned}$$



两个单个离散事件(x_i, y_j)之间的互信息 $I(x_i, y_j)$ 可能为负值，但两个随机变量(X, Y)之间的互信息 $I(X, Y)$ 不可能为负值。后者通常称为平均互信息。

噪声信道模型(noisy channel model)

噪声信道模型的目标就是优化噪声信道中信号传输的吞吐量和准确率，其基本假设是一个信道的输出以一定的概率依赖于输入。



信息论中很重要的一个概念就是信道容量(capacity)，其基本思想是用降低传输速率来换取高保真通讯的可能性。其定义可以根据互信息给出：

$$C = \max_{p(X)} I(X; Y)$$

据此定义，如果我们能够设计一个输入编码 X ，其概率分布为 $p(X)$ ，使其输入与输出之间的互信息达到最大值，那么，我们的设计就达到了信道的最大传输容量。在语言处理中，我们不需要进行编码，只需要进行解码，使系统的输出更接近于输入，如机器翻译。

词汇歧义消解

● 基于上下文分类的消歧方法

(1) 基于贝叶斯分类器 (Gale et al., 1992)

➤ 数学描述：

假设某个多义词 w 所处的上下文语境为 C ，如果 w 的多个语义记作 $s_i (i \geq 2)$ ，那么，可通过计算 $\arg \max p(s_i | C)$ 确定 w 的词义。

(2) 基于最大熵的消歧方法

基本思想：

➤ 在只掌握关于未知分布的部分知识的情况下，符合已知知识的概率分布可能有多个，但使熵值最大的概率分布最真实地反映了事件的分布情况，因为熵定义了随机变量的不确定性，当熵最大时，随机变量最不确定。

➤ 也就是说，在已知部分知识的前提下，关于未知分布最合理的推断应该是符合已知知识最不确定或最大随机的推断。

➤ 均匀分布的熵最大。

语言模型

n 元文法(n-gram)模型

通常地，

◆当 n=1 时，即出现在第 i 位上的基元 w_i 独立于历史。

一元文法也被写为 uni-gram 或 monogram；

◆当 n=2 时, 2-gram (bi-gram) 被称为1阶马尔可夫链；

◆当 n=3 时, 3-gram(tri-gram)被称为2阶马尔可夫链，
依次类推。

为了保证条件概率在 $i=1$ 时有意义，同时为了保证句子内所有字符串的概率和为 1，即 $\sum_s p(s) = 1$ ，可以在句子首尾两端增加两个标志: $\textcolor{red}{<BOSS>} w_1 w_2 \dots w_m \textcolor{red}{<EOS>}$ 。不失一般性，对于 $n > 2$ 的 n -gram, $p(s)$ 可以分解为：

$$p(s) = \prod_{i=1}^{m+1} p(w_i | w_{i-n+1}^{i-1}) \quad \dots (5-4)$$

其中， w_i^j 表示词序列 $w_i \dots w_j$, w_{i-n+1}^{i-1} 从 w_0 开始，
 w_0 为 $\textcolor{red}{<BOSS>}$, w_{m+1} 为 $\textcolor{red}{<EOS>}$ 。

例题:

例如，给定训练语料：

“John read Moby Dick”，

“Mary read a different book”，

“She read a book by Cher”

根据 2 元文法求句子的概率？

$\textcolor{blue}{<BOSS>} John \textcolor{blue}{read} Moby \textcolor{blue}{Dick} \textcolor{blue}{<EOS>}$
 $\textcolor{blue}{<BOSS>} Mary \textcolor{blue}{read} a \textcolor{blue}{different} \textcolor{blue}{book} \textcolor{blue}{<EOS>}$
 $\textcolor{blue}{<BOSS>} She \textcolor{blue}{read} a \textcolor{blue}{book} \textcolor{blue}{by} \textcolor{blue}{Cher} \textcolor{blue}{<EOS>}$

$$p(John | \textcolor{blue}{<BOSS>}) = \frac{c(\textcolor{blue}{<BOSS>} John)}{\sum_w c(\textcolor{blue}{<BOSS>} w)} = \frac{1}{3}$$

$$p(read | John) = \frac{c(John \textcolor{blue}{read})}{\sum_w c(John \textcolor{blue}{w})} = \frac{1}{1}$$

$$p(a | read) = \frac{c(read \textcolor{blue}{a})}{\sum_w c(read \textcolor{blue}{w})} = \frac{2}{3} \quad p(book | a) = \frac{c(a \textcolor{blue}{book})}{\sum_w c(a \textcolor{blue}{w})} = \frac{1}{2}$$

$$p(\textcolor{blue}{<EOS>} | book) = \frac{c(book \textcolor{blue}{<EOS>})}{\sum_w c(book \textcolor{blue}{w})} = \frac{1}{2}$$

$$p(John \textcolor{blue}{read} a \textcolor{blue}{book}) = \frac{1}{3} \times 1 \times \frac{2}{3} \times \frac{1}{2} \times \frac{1}{2} \approx 0.06$$

应用-1：音字转换问题

$$\begin{aligned}\hat{CString} &= \arg \max_{CString} p(CString | Pinyin) \\ &= \arg \max_{CString} \frac{p(Pinyin | CString) \times p(CString)}{p(Pinyin)} \\ &= \arg \max_{CString} p(Pinyin | CString) \times p(CString) \\ &= \arg \max_{CString} p(CString)\end{aligned}$$

如果汉字的总数为：N

- >一元语法：
 - 1)样本空间为 N
 - 2)只选择使用频率最高的汉字
- >2元语法：
 - 1)样本空间为 N²
 - 2)效果比一元语法明显提高
- >估计对汉字而言四元语法效果会好一些
- >智能狂拼、微软拼音输入法基于 n-gram.

应用-2：汉语分词问题

$$\begin{aligned}\hat{Seg} &= \arg \max_{Seg} p(Seg | Text) \\ &= \arg \max_{Seg} \frac{p(Text | Seg) \times p(Seg)}{p(Text)} \\ &= \arg \max_{Seg} p(Text | Seg) \times p(Seg) \\ &= \arg \max_{Seg} p(Seg)\end{aligned}$$

训练集是标注好的，所以必须后验转先验，即利用 Seg 来计算生成 Text 的概率

- >训练语料(training data): 用于建立模型，确定模型参数的已知语料。
- >最大似然估计(maximum likelihood Evaluation, MLE): 用相对频率计算概率的方法。

数据平滑

基本思想：

- 调整最大似然估计的概率值，使零概率增值，使非零概率下调，“劫富济贫”，消除零概率，改进模型的整体正确率。
- 基本目标：测试样本的语言模型困惑度越小越好。

$$\text{•基本约束: } \sum_{w_i} p(w_i | w_1, w_2, \dots, w_{i-1}) = 1$$

加1法(Additive smoothing)

基本思想: 每一种情况出现的次数加1。

对于2-gram有:

$$p(w_i | w_{i-1}) = \frac{1 + c(w_{i-1} w_i)}{\sum_{w_i} [1 + c(w_{i-1} w_i)]}$$
$$= \frac{1 + c(w_{i-1} w_i)}{|V| + \sum_{w_i} c(w_{i-1} w_i)}$$

其中, V 为被考虑语料的词汇量 (全部可能的基元数)。

例如, 对于 *uni-gram*, 设 w_1, w_2, w_3 三个词, 概率分别为: $1/3, 0, 2/3$, 加1后情况?

2/6, 1/6, 3/6

词汇量: $|V|=11$

<BOS>John read Moby Dick<EOS>
<BOS>Mary read a different book<EOS>
<BOS>She read a book by Cher<EOS>

平滑以后:

$$p(\text{Cher}|\text{BOS}) = (0+1)/(11+3) = 1/14$$
$$p(\text{read}|\text{Cher}) = (0+1)/(11+1) = 1/12$$
$$p(a|\text{read}) = (1+2)/(11+3) = 3/14$$
$$p(\text{book}|a) = (1+1)/(11+2) = 2/13$$
$$p(\text{EOS}|book) = (1+1)/(11+2) = 2/13$$

$$p(\text{Cher read a book}) = \frac{1}{14} \times \frac{1}{12} \times \frac{3}{14} \times \frac{2}{13} \times \frac{2}{13} \approx 0.00003$$

减值法/折扣法(Discounting)

基本思想: 修改训练样本中事件的实际计数, 使样本中(实际出现的)不同事件的概率之和小于1, 剩余的概率量分配给未见概率。

①Good-Turing 估计

假设 N 是原来训练样本数据的大小, n_r 是在样本中正好出现 r 次的事件的数目(此处事件为 n -gram), 即出现 1 次的 n -gram 有 n_1 个, 出现 2 次的 n -gram 有 n_2 个, ……, 出现 r 次的有 n_r 个。

$$\text{那么, } N = \sum_{r=1}^{\infty} n_r r = \sum_{r=0}^{\infty} (r+1) n_{r+1} \dots (5-6)$$

设：原先出现 r 次的 n -gram在平滑后出现 r^* 次

$$\text{则 } N = \sum_{r=0}^{\infty} n_r r^* \quad \text{则 } \sum_{r=0}^{\infty} n_r r^* = \sum_{r=0}^{\infty} (r+1) n_{r+1}$$

$$\text{所以, } r^* = (r+1) \frac{n_{r+1}}{n_r}$$

那么，Good-Turing 估计在样本中出现 r 次的事件的平滑后的概率为：

$$p_r = \frac{r^*}{N} \dots (5-7)$$

举例说明：假设有如下英语文本，估计 2-gram 概率：

<BO>John read Moby Dick<EO>
 <BO>Mary read a different book<EO>
 <BO>She read a book by Cher<EO>

从文本中统计出不同 2-gram 出现的次数：

<BO>	John	15
<BO>	Mary	10
.....		
read	Moby	5
.....		

假设要估计以 read 开始的 2-gram 概率，列出以 read 开始的所有 2-gram，并转化为频率信息：

r	n_r	r^*
1	2053	0.446
2	458	1.25
3	191	2.24
4	107	3.22
5	69	4.17
6	48	5.25
7	36	保持原来的计数7

$$r^* = (r+1) \frac{n_{r+1}}{n_r}$$

因为 $n_{r+1} = 0$

得到 r^* 后，就可以应用公式(5-7) 计算概率：

$$p_r = \frac{r^*}{N} \quad \dots (5-7)$$

其中， N 为以 read 开始的 2-gram 的总数(样本空间)，即 read 出现的次数。

那么，以 read 开始，没有出现过的 2-gram 的概率总和为：

$$P_0 = \frac{n_1}{N}$$

以 read 作为开始，没有出现过的 2-gram 的个数等于：

$$n_0 = |V_T| - \sum_{r>0} n_r \quad \text{其中, } |V_T| \text{ 为语料的词汇量。}$$

那么，没有出现过的那些以 read 为开始的 2-gram 的概率平均为： $\frac{P_0}{n_0}$ 。

注意： $\sum_{r=0}^7 p_r \neq 1$

因此，需要归一化处理：

$$\hat{p}_r = \frac{p_r}{\sum_r p_r}$$

r	n_r	r^*
1	2053	0.446
2	458	1.25
3	191	2.24
4	107	3.22
5	69	4.17
6	48	5.25
7	36	—

②Back-off (后备/后退)方法

又称 Katz 后退法。

基本思想：当某一事件在样本中出现的频率大于阈值 K (通常取 K 为 0 或 1) 时，运用最大似然估计的减值法来估计其概率，否则，使用低阶的，即 $(n-1)$ -gram 的概率替代 n -gram 概率，而这种替代需受归一化因子 α 的作用。

以2-gram模型为例, 说明Katz平滑方法:

对于一个出现次数为 $r = c(w_{i-1}^i)$ 的 2-gram w_{i-1}^i , 修正其概率:

$$p_{katz}(w_i|w_{i-1}) = \begin{cases} d_r \frac{C(w_{i-1}w_i)}{C(w_{i-1})} & \text{if } C(w_{i-1}w_i) = r > 0 \\ \alpha(w_{i-1})p_{ML}(w_i) & \text{if } C(w_{i-1}w_i) = 0 \end{cases}$$

其中, $p_{ML}(w_i)$ 表示 w_i 的最大似然估计概率。

公式的意思是, 所有具有非零计数 r 的 2-gram 都根据折扣率 d_r ($0 < d_r < 1$) 被减值了, 折扣率 d_r 近似地等于 r^*/r , 减值由Good-Turing估计方法预测。

③绝对减值法 (Absolute discounting)

基本思想: 从每个计数 r 中减去同样的量, 剩余的概率量由未见事件均分。设 R 为所有可能事件的数目 (当事件为 n-gram 时, 如果统计基元为词, 且词汇集的大小为 L , 则 $R=L^n$)。

那么, 样本出现了 r 次的事件的概率可以由如下公式估计:

$$p_r = \begin{cases} \frac{r-b}{N} & \text{当 } r > 0 \\ \frac{b(R-n_0)}{Nn_0} & \text{当 } r = 0 \end{cases} \dots (5-10)$$

其中, n_0 为样本中未出现的事件的数目。 b 为减去的常量, $b \leq 1$ 。

④线性减值法 (Linear discounting)

基本思想: 从每个计数 r 中减去与该计数成正比的量(减值函数为线性的), 剩余概率量被 n_0 个未见事件均分。

$$p_r = \begin{cases} (1-\alpha) \frac{N_r}{N} & \text{当 } r > 0 \\ \frac{\alpha}{n_0} & \text{当 } r = 0 \end{cases}$$

自由参数 α 的优化值为: $\frac{n_1}{N}$

绝对减值法产生的n-gram 通常优于线性减值法。

◆ 四种减值法的比较

- **Good-Turing 法：**对非0事件按公式削减出现的次数，节留出来的概率**均分**给0概率事件。
- **Katz 后退法：**对非0事件按Good-Turing法计算减值，节留出来的概率**按低阶分布**分给0概率事件。
- **绝对减值法：**对非0事件无条件削减某一**固定的**出现次数值，节留出来的概率**均分**给0概率事件。
- **线性减值法：**对非0事件根据出现次数**按比例**削减次数值，节留出来的概率**均分**给0概率事件。

(3)删除插值法(Deleted interpolation)

基本思想：用低阶语法估计高阶语法，即当3-gram的值不能从训练数据中准确估计时，用2-gram来替代，同样，当2-gram的值不能从训练语料中准确估计时，可以用1-gram的值来代替。插值公式：

$$p(w_3 | w_1 w_2) = \lambda_3 p'(w_3 | w_1 w_2) + \lambda_2 p'(w_3 | w_2) + \lambda_1 p'(w_3) \quad \dots (5-13)$$

其中， $\lambda_1 + \lambda_2 + \lambda_3 = 1$

语言模型的自适应方法：

(1) **基于缓存的语言模型 (cache-based LM)**

该方法针对的问题是：在文本中刚刚出现过的一些词在后边的句子中再次出现的可能性往往较大，比标准的n-gram模型预测的概率要大。

针对这种现象，cache-based自适应方法的基本思路是：

语言模型通过 n-gram 的线性插值求得：

$$\hat{p}(w_i | w_1^{i-1}) = \lambda \hat{p}_{Cache}(w_i | w_1^{i-1}) + (1 - \lambda) \hat{p}_{n-gram}(w_i | w_{i-n+1}^{i-1}) \quad \dots (5-14)$$

插值系数 λ 可以通过EM算法求得。

通常的处理方法是：在缓存中**保留前面的 K 个单词**，每个词的概率（缓存概率）用其在缓存中出现的**相对频率**计算得出：

$$\hat{p}_{Cache}(w_i | w_1^{i-1}) = \frac{1}{K} \sum_{j=i-K}^{i-1} I_{\{w_j=w_i\}} \quad \dots (5-15)$$

其中， I_{ε} 为指示器函数(indicator function)，如果词重复出现($w_j=w_i$)，则 $I_{\varepsilon}=1$ ，否则 $I_{\varepsilon}=0$ 。

这种方法的缺陷是，缓存中一个词的重要性独立于该词与当前词的距离。

P. R. Clarkson等人(1997) 的研究表明，缓存中每个词对当前词的影响随着与该词距离的增大呈指数级衰减，因此，将(5-15)式写成：

$$\hat{p}_{Cache}(w_i | w_1^{i-1}) = \beta \sum_{j=1}^{i-1} I_{\{w_i=w_j\}} e^{-\alpha(i-j)} \quad \dots(5-16)$$

其中， α 为衰减率， β 为归一化常数，以使得：

$$\sum_{w_i \in V} \hat{p}_{Cache}(w_i | w_1^{i-1}) = 1, \quad V \text{ 为词汇表。}$$

(2) 基于混合方法的语言模型

该方法针对的问题是：由于大规模训练语料本身是异源的(heterogenous)，来自不同领域的语料无论在主题(topic)方面，还是在风格(style)方面，或者两者都有一定的差异，而测试语料一般是同源的(homogeneous)，因此，为了获得最佳性能，语言模型必须适应各种不同类型的语料对其性能的影响。

处理方法是：将语言模型划分成 n 个子模型 M_1, M_2, \dots, M_n ，整个语言模型的概率通过下面的线性插值公式计算得到：

$$\hat{p}(w_i | w_1^{i-1}) = \sum_{j=1}^n \lambda_j \hat{p}_{M_j}(w_i | w_1^{i-1}) \quad \dots(5-17)$$

其中， $0 \leq \lambda_j \leq 1, \sum_{j=1}^n \lambda_j = 1$

λ 值可以通过 EM 算法计算出来。

基本方法

- ① 对训练语料按来源、主题或类型等聚类(设为 n 类)；
- ② 在模型运行时识别测试语料的主题或主题的集合；
- ③ 确定适当的训练语料子集，并利用这些语料建立特定的语言模型；
- ④ 利用针对各个语料子集的特定语言模型和线性插值公式(5-17)，获得整个语言模型。

$$\hat{p}(w_i | w_1^{i-1}) = \sum_{j=1}^n \lambda_j \hat{p}_{M_j}(w_i | w_1^{i-1}) \quad \dots(5-17)$$

(3) 基于最大熵的语言模型

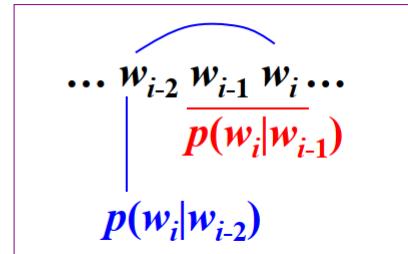
基本思想：通过结合不同信息源的信息构建一个语言模型。每个信息源提供一组关于模型参数的约束条件，在所有满足约束的模型中，选择熵最大的模型。

例如，考虑两个语言模型 M_1 和 M_2 ，假设 M_1 是标准的 2 元模型，表示为 f 函数：

$$\hat{p}_{M_1}(w_i | w_1^{i-1}) = f(w_i, w_{i-1}) \quad \dots (5-18)$$

M_2 是距离为 2 的 2 元模型 (distance-2 bigram)，定义为 g 函数：

$$\hat{p}_{M_2}(w_i | w_1^{i-1}) = g(w_i, w_{i-2}) \quad \dots (5-19)$$



用线性插值方法通过取这两个概率估计的平均值，并采用后备(backing-off) 平滑技术来解决这个问题。

最大熵原则将所有的信息源组合成一个模型，对于该模型的约束并不是让公式(5-18)和(5-19)对于所有可能的历史都成立，而是更宽松的限制，即它们在训练数据上平均成立即可，因此，公式(5-18)和(5-19)被分别改写成：

$$E(\hat{p}_{M_1}(w_i | w_1^{i-1}) | w_{i-1} = a) = f(w_i, a) \quad \dots (5-20)$$

$$E(\hat{p}_{M_2}(w_i | w_1^{i-1}) | w_{i-2} = b) = g(w_i, b) \quad \dots (5-21)$$

如果约束条件是一致的(相互之间不矛盾)，那么，总有模型满足这些条件，余下的问题就是利用通用迭代算法 (generalized iterative scaling, GIS) 选择使熵最大的模型。

应用到汉语分词

采用基于语言模型的分词方法

➤ 方法描述:

设对于待切分的句子 $S = z_1 z_2 \dots z_m$, $W = w_1 w_2 \dots w_k$ ($1 \leq k \leq n$) 是一种可能的切分。那么,

$$\begin{aligned}\hat{W} &= \arg \max_W p(W | S) \\ &= \arg \max_W p(W) \times p(S | W) \\ &\cong \arg \max_W p(W)\end{aligned}$$

最基本的做法是以词为独立的统计基元，但效果不佳。

那么, $\hat{C} = \arg \max_C p(C | S)$

$$= \arg \max_C p(C) \times p(S | C) \quad \dots (5-22)$$

语言模型 生成模型

$p(C)$ 可采用三元语法:

$$p(C) = p(c_1) \times p(c_2 | c_1) \prod_{i=3}^N p(c_i | c_{i-2} c_{i-1}) \quad \dots (5-23)$$

$$p(c_i | c_{i-2} c_{i-1}) = \frac{\text{count}(c_{i-2} c_{i-1} c_i)}{\text{count}(c_{i-2} c_{i-1})} \quad \dots (5-24)$$

生成模型在满足独立性假设的条件下，可近似为:

$$p(S | C) \approx \prod_{i=1}^N p(s_i | c_i) \quad \dots (5-25)$$

该公式的含意是，任意一个词类 c_i 生成汉字串 s_i 的概率只与自身有关，而与其上下文无关。

例如，如果“教授”是词表里的词，那么

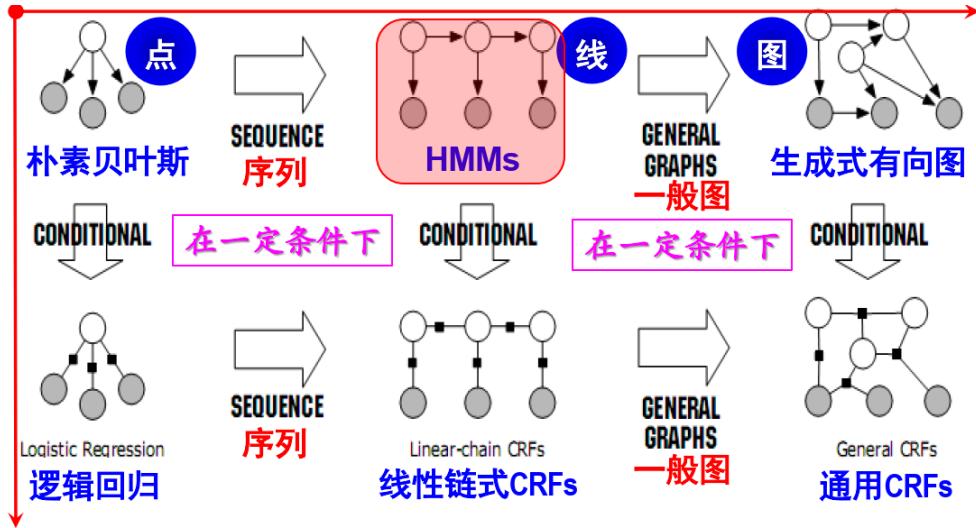
$$p(s_i = \text{教授} | c_i = \text{LW}) = 1, \text{ 否则, } p(s_i | c_i) = 0.$$

模型的训练分三步:

- (1) 在词表和派生词表的基础上，用一个基本的分词工具切分训练语料，专有名词通过一个专门模块标注，实体名词通过相应的规则和有限状态自动机标注，由此产生一个带词类别标记的初始语料；
- (2) 用带词类别标记的初始语料，采用最大似然估计方法估计语言模型的概率参数，公式(5-24)；
- (3) 用得到的模型（公式(5-22)、(5-23)、(5-25)）对训练语料重新切分和标注，得到新的训练语料；
- (4) 重复(2)(3)步，直到系统的性能不再有明显的变化为止。

隐马尔可夫模型与条件随机场

概率图模型(Probabilistic Graphical Model)是使用图表示变量及变量间概率依赖关系的方法。在概率图模型中，可以根据可观测变量推测出未知变量的条件概率分布等信息。如果把序列标注任务中的输入序列看作观测变量，而把输出序列看作需要预测的未知变量，那么就可以把概率图模型应用于命名实体识别等序列标注任务。



隐马尔可夫模型

隐马尔可夫模型是一种经典的序列模型[96, 102, 103]。它在语音识别、自然语言处理的很多领域得到了广泛的应用。隐马尔可夫模型的本质就是概率化的马尔可夫过程，这个过程隐含着状态间转移和可见状态生成的概率。该模型是一个双重随机过程，我们不知道具体的状态序列，只知道状态转移的概率，即模型的状态转换过程是不可观察的（隐蔽的），而可观察事件的随机过程是隐蔽状态转换过程的随机函数。

一方面，**隐马尔可夫模型中用发射概率(Emission Probability)**来描述**隐含状态**和**可见状态**之间存在的输出概率，同样的，**隐马尔可夫模型还会描述系统**隐含状态的转移概率(Transition Probability)****，它们都可以被看做是条件概率矩阵。

一般来说，隐马尔可夫模型中包含下面三个问题：

- 隐含状态序列的概率计算，即给定模型(转移概率和发射概率)，根据可见状态序列计算在该模型下得到这个结果的概率，这个问题的求解需要用到前后向算法。
- 参数学习，即给定硬币种类(隐含状态数量)，根据多个可见状态序列估计模型的参数(转移概率)，这个问题的求解需要用到 EM 算法。
- 解码，即给定模型(转移概率和发射概率)和可见状态序列，计算在可见状态序列的情况下，最可能出现的对应的状态序列，这个问题的求解需要用到基于动态规划的方法，通常也被称作维特比算法(Viterbi Algorithm)。

隐马尔可夫模型处理序列标注问题的基本思路是：

- 第一步：根据可见状态序列（输入序列）和其对应的隐含状态序列（标记序列）样本，估算模型的转移概率和发射概率；
- 第二步：对于给定的可见状态序列，预测概率最大的隐含状态序列，比如，根据输入的词序列预测最有可能的命名实体标记序列

◆问题1：快速计算观察序列概率 $p(O|\mu)$

给定模型 $\mu = (A, B, \pi)$ 和观察序列 $O = O_1 O_2 \dots O_T$ ，
快速计算 $p(O|\mu)$ ：

对于给定的状态序列 $Q = q_1 q_2 \dots q_T$, $p(O|\mu) = ?$

●解决办法：动态规划

前向算法(The forward procedure)

●基本思想：定义前向变量 $\alpha_t(i)$: 在时间 t , 输出序列 O_1, O_2, \dots, O_t 并且位于状态 s_i 的概率

●后向算法 (The backward procedure)

定义后向变量 $\beta_t(i)$ 是在给定了模型 $\mu = (A, B, \pi)$ 和假定在时间 t 状态为 s_i 的条件下，模型输出观察序列 $O_{t+1} O_{t+2} \dots O_T$ 的概率：

◆问题2—如何发现“最优”状态序列 能够“最好地解释”观察序列

解释不是唯一的，关键在于如何理解“最优”的状态序列？一种解释是：状态序列中的每个状态都单独地具有概率，对于每个时刻 t ($1 \leq t \leq T$)，寻找 q_t 使得 $\gamma_t(i) = p(q_t = s_i | O, \mu)$ 最大。

另一种解释：在给定模型 μ 和观察序列 O 的条件下求概率最大的状态序列：

$$Q = \arg \max_Q p(Q | O, \mu) \quad \dots (6.21)$$

Viterbi 算法: 动态搜索最优状态序列。

定义: Viterbi 变量 $\delta_t(i)$ 是在时间 t 时，模型沿着某一条路径到达 s_i ，并输出观察序列 $O = O_1 O_2 \dots O_t$ 的最大概率：

算法复杂性均为 $O(N^2 T)$

◆问题3—模型参数学习

给定一个观察序列 $O = O_1 O_2 \dots O_T$, 如何根据最大似然估计来求模型的参数值? 或者说如何调节模型 μ 的参数, 使得 $p(O|\mu)$ 最大? 即估计模型中的 $\pi_i, a_{ij}, b_j(k)$ 使得观察序列 O 的概率 $p(O|\mu)$ 最大。

如果产生观察序列 O 的状态 $Q = q_1 q_2 \dots q_T$ 已知(即存在大量标注的样本), 可以用最大似然估计来计算 μ 的参数:

如果不存在大量标注的样本 -

- **期望值最大化算法** (Expectation-Maximization, EM)

基本思想: 初始化时随机地给模型的参数赋值(遵循限制规则, 如: 从某一状态出发的转移概率总和为1, 得到模型 μ_0 ,

然后可以从 μ_0 得到从某一状态转移到另一状态的期望次数, 然后以期望次数代替公式中的次数, 得到模型参数的新估计, 由此得到新的模型 μ_1 ,

从 μ_1 又可得到模型中隐变量的期望值, 由此重新估计模型参数。循环这一过程, 参数收敛于最大似然估计值。

算法6.4: Baum-Welch 算法(前向后向算法)描述:

(1) 初始化：随机地给 π_i , a_{ij} , $b_j(k)$ 赋值,

使得

$$\left\{ \begin{array}{l} \sum_{i=1}^N \pi_i = 1 \\ \sum_{j=1}^N a_{ij} = 1 \quad 1 \leq i \leq N \\ \sum_{k=1}^M b_j(k) = 1 \quad 1 \leq j \leq M \end{array} \right. \dots (6.31)$$

由此得到模型 μ_0 , 令 $i = 0$ 。

(2) 执行 EM 算法:

$$\xi_t(i, j) = \frac{\alpha_i(i) \times a_{ij} b_j(O_{t+1}) \times \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^M \alpha_i(i) \times a_{ij} b_j(O_{t+1}) \times \beta_{t+1}(j)}$$

$$\gamma_t(i) = \sum_{j=1}^N \xi_t(i, j)$$

E-步: 由模型 μ_i 根据公式 (6.26) 和 (6.27) 计算期望值 $\xi_t(i, j)$ 和 $\gamma_t(i)$ 。

M-步: 用 E-步中所得到的期望值, 根据公式 (6.28-6.30) 重新估计 π_i , a_{ij} , $b_j(k)$ 得到模型 μ_{i+1} 。

循环: $i = i+1$, 重复执行 E-步和 M-步, 直至 π_i , a_{ij} , $b_j(k)$ 的值收敛: $|\log p(O | \mu_{i+1}) - \log p(O | \mu_i)| < \varepsilon$ 。

(3) 结束算法, 获得相应的参数。

● HMM 使用中注意的问题

● Viterbi 算法运算中的小数连乘, 出现溢出
—取对数

● Baum-Welch 算法的小数溢出
—放大系数

条件随机场

条件随机场模型在隐马尔可夫模型的基础上, 解决了这个问题 标注偏置 (Label Bias)。

基本思路: 给定观察序列 X, 输出标识序列 Y, 通过计算 $P(Y|X)$ 求解最优标注序列。

实现 CRFs 也需要解决如下三个问题:

- ① 特征选取
- ② 参数训练
- ③ 解码

定义和选取特征函数, 利用 GIS 迭代算法选取 λ 权重。

请参阅前面第 2 章的最大熵模型

条件随机场模型处理命名实体识别任务时, 可见状态序列对应着文本内容, 隐含状态序列对应着待预测的标签。对于命名实体识别任务, 需要单独设计若干适合命名实体识别任务的特征函数。例如在使用 BIOES 标准标注命名实体识别任务时, 标签“B-ORG”后面的标签必然是“I-ORG”或是“E-ORG”, 而不可能是“O”, 针对此规则可以设计相应特征函数。

条件随机场中一般有若干个特征函数，都是经过设计的、能够反映序列规律的一些二元函数⁴，并且每个特征函数都有其对应的权重 λ 。特征函数一般由两部分组成：能够反映隐含状态序列之间转移规则的转移特征 $t(y_{i-1}, y_i, x, i)$ 和状态特征 $s(y_i, x, i)$ 。其中 y_i 和 y_{i-1} 分别是位置 i 和前一个位置的隐含状态， x 则是可见状态序列。转移特征 $t(y_{i-1}, y_i, x, i)$ 反映了两个相邻的隐含状态之间的转换关系，而状态特征 $s(y_i, x, i)$ 则反映了第 i 个可见状态应该对应什么样的隐含状态，这两部分共同组成了一个特征函数 $F(y_{i-1}, y_i, x, i)$ ，即

$$F(y_{i-1}, y_i, x, i) = t(y_{i-1}, y_i, x, i) + s(y_i, x, i) \quad (3.10)$$

实际上，基于特征函数的方法更像是对隐含状态序列的一种打分：根据人为设计的模板（特征函数），测试隐含状态之间的转换以及隐含状态与可见状态之间的对应关系是否符合这种模板。在处理序列问题时，假设可见状态序列 x 的长度和待预测隐含状态序列 y 的长度均为 m ，且共设计了 k 个特征函数，则有：

$$P(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i=1}^m \sum_{j=1}^k \lambda_j F_j(y_{i-1}, y_i, x, i)\right) \quad (3.11)$$

◆HMM 的构成：

①状态数 ②输出符号数 ③初始状态的概率分布 ④状态转移的概率 ⑤输出概率

◆HMM 的三个基本问题：

- (1) 快速计算给定模型的观察序列概率：前/后向算法
- (2) 求最优状态序列：Viterbi 算法
- (3) 参数估计：Baum-Welch 算法

◆模型实现中需要注意的问题：小数溢出

◆条件随机场(CRFs)

词法分析与词性标注

概述

词性或称词类(Part-of-Speech, POS)是词汇最重要的特性，是连接词汇到句法的桥梁。

不同语言的词法分析

- 曲折语(如，英语、德语、俄语等)：用词的形态变化表示语法关系，一个形态成分可以表示若干种不同的语法意义，词根和词干与语词的附加成分结合紧密。

词法分析：词的形态分析(形态还原)。

- 分析语(孤立语)(如：汉语)：分词。
- 黏着语(如：日语等)：分词 + 形态还原。

英语的形态分析

基本任务

- 单词识别

- 形态还原

◆形态分析的一般方法

- 1) 查词典，如果词典中有该词，直接确定该词的原形；
- 2) 根据不同情况查找相应规则对单词进行还原处理，如果还原后在词典中找到该词，则得到该词的原形；如果找不到相应变换规则或者变换后词典中仍查不到该词，则作为未登录词处理；
- 3) 进入未登录词处理模块。

汉语自动分词

汉语自动分词的基本原则

- 1、语义上无法由组合成分直接相加而得到的字串应该合并为一个分词单位。(合并原则)
- 2、语类无法由组合成分直接得到的字串应该合并为一个分词单位。(合并原则)

汉语自动分词的辅助原则

1. 有明显分隔符标记的应该切分之(切分原则)
2. 附着性语(词)素和前后词合并为一个分词单位(合并原则)
3. 使用频率高或共现率高的字串尽量合并为一个分词单位(合并原则)
4. 双音节加单音节的偏正式名词尽量合并为一个分词单位(合并原则)
5. 双音节结构的偏正式动词应尽量合并为一个分词单位(合并原则)
6. 内部结构复杂、合并起来过于冗长的词尽量切分(切分原则)

分词与词性标注结果评价方法

◆两种测试

- 封闭测试/开放测试
- 专项测试/总体测试

评价指标

- 正确率(Correct ratio/Precision, P): 测试结果中正确切分或标注的个数占系统所有输出结果的比例。假设系统输出N个，其中，正确的结果为n个，那么， $P = \frac{n}{N} \times 100\%$
- 召回率(找回率) (Recall ratio, R): 测试结果中正确结果的个数占标准答案总数的比例。假设系统输出N个结果，其中，正确的结果为n个，而标准答案的个数为M个，那么，

$$R = \frac{n}{M} \times 100\%$$

- 两种标记：ROOV 指集外词的召回率；RIV 指集内词的召回率。

F-测度值(F-Measure): 正确率与找回率的综合值。

计算公式为：

$$F - measure = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R} \times 100 \%$$

一般地，取 $\beta=1$ ，即

$$F1 = \frac{2 \times P \times R}{P + R} \times 100 \%$$

假设某个汉语分词系统在一测试集上输出 5260 个分词结果，而标准答案是 4510 个词语，根据这个答案，系统切分出来的结果中有 4120 个是正确的。那么：

$$P = \frac{4120}{5260} \times 100\% = 78.33\%$$

$$R = \frac{4120}{4510} \times 100\% = 91.35\%$$

$$\begin{aligned} F1 &= \frac{2 \times P \times R}{P + R} \times 100\% \\ &= \frac{2 \times 78.33 \times 91.35}{78.33 + 91.35} \times 100\% \\ &= 84.34\% \end{aligned}$$

汉语自动分词基本算法

- ◆有词典切分/无词典切分
- ◆基于规则的方法/基于统计的方法

1. 最大匹配法 (Maximum Matching, MM)

—有词典切分，机械切分

- 正向最大匹配算法 (Forward MM, FMM)
- 逆向最大匹配算法 (Backward MM, BMM)
- 双向最大匹配算法 (Bi-directional MM)

假设句子: $S = c_1 c_2 \dots c_n$, 某一词:

$w_i = c_1 c_2 \dots c_m$, m 为词典中最长词的字数。

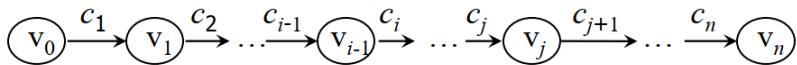
➤ FMM 算法描述

- (1) 令 $i=0$, 当前指针 p_i 指向输入字串的初始位置, 执行下面的操作:
- (2) 计算当前指针 p_i 到字串末端的字数 (即未被切分字串的长度) n , 如果 $n=1$, 转(4), 结束算法。否则, 令 m =词典中最长单词的字数, 如果 $n < m$, 令 $m=n$;
- (3) 从当前 p_i 起取 m 个汉字作为词 w_i , 判断:
 - (a) 如果 w_i 确实是词典中的词, 则在 w_i 后添加一个切分标志, 转(c);
 - (b) 如果 w_i 不是词典中的词且 w_i 的长度大于1, 将 w_i 从右端去掉一个字, 转(a)步; 否则 (w_i 的长度等于1), 则在 w_i 后添加一个切分标志, 将 w_i 作为单字词添加到词典中, 执行 (c)步;
 - (c) 根据 w_i 的长度修改指针 p_i 的位置, 如果 p_i 指向字串末端, 转(4), 否则, $i=i+1$, 返回 (2);
- (4) 输出切分结果, 结束分词程序。

2. 最少分词法 (最短路径法)

➤ 基本思想

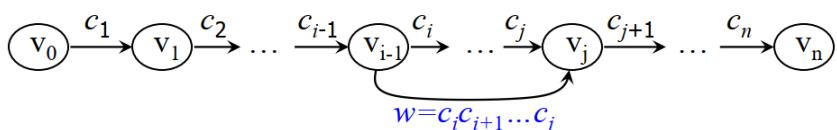
设待切分字串 $S=c_1 c_2 \dots c_n$, 其中 $c_i (i=1, 2, \dots, n)$ 为单个的字, n 为串的长度, $n \geq 1$ 。建立一个节点数为 $n+1$ 的切分有向无环图 G , 各节点编号依次为 $V_0, V_1, V_2, \dots, V_n$ 。



求最短路径: 贪心法或简单扩展法。

➤ 算法描述:

- (1) 相邻节点 v_{k-1}, v_k 之间建立有向边 $\langle v_{k-1}, v_k \rangle$, 边对应的词默认为 c_k ($k=1, 2, \dots, n$)。
- (2) 如果 $w=c_i c_{i+1} \dots c_j$ ($0 < i < j \leq n$) 是一个词, 则节点 v_{i-1}, v_j 之间建立有向边 $\langle v_{i-1}, v_j \rangle$, 边对应的词为 w 。



- (3) 重复步骤(2), 直到没有新路径(词序列)产生。

- (4) 从产生的所有路径中, 选择路径最短的(词数最少的)作为最终分词结果。

3. 基于语言模型的分词方法

➤ 方法描述:

设对于待切分的句子 S , $W = w_1 w_2 \dots \dots w_k$ ($1 \leq k \leq n$) 是一种可能的切分。

$$\begin{aligned} W^* &= \arg \max_W p(W | S) \\ &= \arg \max_W p(W) \times p(S | W) \end{aligned}$$

详见第5章举例。

语言模型

生成模型

➤ 优点:

- 减少了很多手工标注的工作;
- 在训练语料规模足够大和覆盖领域足够多时, 可以获得较高的切分正确率。

➤ 弱点:

- 训练语料的规模和覆盖领域不好把握;
- 计算量较大。

4. 基于HMM的分词方法

➤ 基本思想:

把输入字串(句子) S 作为HMM μ 的输入; 切分后的单词串 S_w 为状态的输出, 即观察序列 $S_w = w_1 w_2 \dots w_n$, $n \geq 1$ 。词性序列 S_c 为状态序列, 每个词性标记 c_i 对应 HMM 中的一个状态 q_i , $S_c = c_1 c_2 \dots c_n$ 。

➤ 优点:

- 可以减少很多手工标注的工作量;
- 在训练语料规模足够大和覆盖领域足够多时, 可以获得较高的切分正确率。

➤ 弱点:

- 训练语料的规模和覆盖领域不好把握;
- 模型实现复杂、计算量较大。

5. 由字构词 (基于字标注) 的分词方法

(Character-based tagging)

➤ **基本思想:** 将分词过程看作是字的分类问题。该方法认为，每个字在构造一个特定的词语时都占据着一个确定的构词位置(即词位)。假定每个字只有4个词位：词首(B)、词中(M)、词尾(E)和单独成词(S)，那么，每个字归属一特定的词位。

➤ **评价:**

该方法的重要优势在于，它能够平衡地看待词表词和未登录词的识别问题，文本中的词表词和未登录词都是用统一的字标注过程来实现的。在学习构架上，既可以不必专门强调词表词信息，也不用专门设计特定的未登录词识别模块，因此，大大地简化了分词系统的设计[黄昌宁，2006]

6. 生成式方法与区分式方法的结合

大部分基于词的分词方法采用的是生成式模型

(Generative model):

$$\begin{aligned} WSeq^* &= \arg \max_{WSeq} p(WSeq | c_1^n) \\ &= \arg \max_{WSeq} p(WSeq) \end{aligned}$$

使用 3-gram:

$$p(w_1^m) = \prod_{i=1}^m p(w_i | w_1^{i-1}) \approx \prod_{i=1}^m p(w_i | w_{i-2}^{i-1})$$

而基于字的分词方法采用区分式模型

(Discriminative model):

$$\begin{aligned} P(t_1^n | c_1^n) &= \prod_{k=1}^n P(t_k | t_1^{k-1}, c_1^n) \approx \prod_{k=1}^n P(t_k | c_{k-2}^{k+2}) \\ &\cdots \quad \cdots \quad c_{k-2} \quad c_{k-1} \quad c_k \quad c_{k+1} \quad c_{k+2} \quad \cdots \quad \cdots \\ &\qquad\qquad\qquad \uparrow \\ &\qquad\qquad\qquad \underbrace{\quad\quad\quad}_{B, M, S, E} \end{aligned}$$

生成式模型与判别式模型的比较

➤生成(产生)式模型 (Generative Model)

假设 o 是观察值, q 是模型。如果对 $p(o|q)$ 进行建模, 就是生成式模型。其基本思想是: 首先建立样本的概率密度模型, 再利用模型进行推理预测。要求已知样本无穷多或者尽可能地多。该方法一般建立在统计学和 Bayes 理论的基础之上。

- **主要特点:** 从统计的角度表示数据的分布情况, 能够反映同类数据本身的相似度。
- **主要优点:** 实际上所带的信息要比判别式模型丰富
研究单类问题比判别式模型灵活性强, 模型可以通过增量学习得到, 且能用于数据不完整(missing data)情况。
- **主要缺点:** 学习和计算过程比较复杂。

➤ 判别(区分)式模型 (Discriminative Model)

如果对条件概率(后验概率) $p(q|o)$ 进行建模, 就是判别式模型。基本思想是: 有限样本条件下建立判别函数, 不考虑样本的产生模型, 直接研究预测模型。表性理论为统计学习理论。

- **主要特点:** 寻找不同类别之间的最优分类面, 反映的是异类数据之间的差异。
- **主要优点:** 判别式模型比生成式模型较容易学习。
- **主要缺点:** 黑盒操作, 变量间的关系不清楚, 不可视。

基于字的区分模型有利于处理集外词, 而基于词的生成模型更多地考虑了词汇之间以及词汇内部字与字之间的依存关系。因此, 可以将两者的优势结合起来。

✧ **结合方法1:** 将待切分字串的每个汉字用 $[c, t]_i$ 替代, 以 $[c, t]_I$ 作为基元, 利用语言模型选取全局最优(生成式模型)。

✧ **结合方法2: 插值法把两种方法结合起来**

$$Score(t_k) = \alpha \times \log(P([c, t]_k | [c, t]_{k-2}^{k-1})) + (1 - \alpha) \times \log(P(t_k | c_{k-2}^{k+2}))$$

$(0.0 \leq \alpha \leq 1.0)$

Generative scoreDiscriminative score

- **这样做的优点:**

充分结合了基于字的生成模型和基于字的区分式模型的优点。

➤ 基本方法

- 统计模型
- 通过训练语料选取阈值
- 地名初筛选
- 寻找可以利用的上下文信息
- 利用规则进一步确定地名

词性标注方法

基于规则的词性标注方法

- 手工编写词性歧义消除规则
- 机器自动学习规则

基于统计模型的词性标注方法

- 基于错误驱动的机器学习方法

➤ 初始词性赋值

- 对比正确标注的句子，自动学习结构转换规则
- 利用转换规则调整初始赋值

基于 HMM 的词性标注方法

规则和统计方法相结合的词性标注方法

- 规则消歧，统计概率引导
- 或者统计方法赋初值，规则消歧

基于有限状态变换机的词性标注方法

基于神经网络的词性标注方法

当前分词技术存在的主要问题

- 分词模型过于依赖训练样本，而标注大规模训练样本费时费力，且仅局限于个别领域，由此导致分词系统对新词的识别能力差，往往在与训练样本差异较大的测试集上性能大幅度下降。
- 现有的训练样本主要在新闻领域，而实际应用千差万别：网络新闻、微博/微信/QQ等对话文本、不同的专业领域(中医药、生物、化学、能源……)。

关于词性标注

- 进一步研究消歧方法，与其他技术相结合（如分词、句法分析等），提高性能
- 在有些任务或方法中，词性作用并不大，如基于词的统计机器翻译、目前的神经网络机器翻译等

本章小结

- ◆词法分析的任务（英语汉语有所不同）
- ◆英语形态分析
 - 单词识别 ➤形态还原
- ◆汉语自动分词
 - 汉语分词中的主要问题
 - 基本原则和辅助原则
 - 几种基本方法：MM、最少分词法、统计法等
- ◆未登录词识别
 - 人名、地名、组织机构名、特殊符号等
- ◆词性标注
 - 问题(兼类、标注集、规范)
 - 方法(规则方法、统计方法、综合方法)
- ◆分词与词性标注结果评测
 - 正确率、找回率、F-测度值
- ◆分词与词性标注下一步努力的方向

语义分析

◆**语义计算的任务**: 解释自然语言句子或篇章各部分(词、词组、句子、段落、篇章)的含义。

◆**面临的困难**:

- 自然语言句子中存在大量的歧义，涉及指代、同义/多义、量词的辖域、隐喻等；
- 同一句子对于不同的人来说可能有不同的理解；
- 语义计算的理论、方法、模型尚不成熟。

语义理论

词的指称作为意义

心理图像、大脑图像或思想作为意义

说话者的意图作为意义

过程语义

词汇分解学派

条件真理模型

情景语义学

模态逻辑

格语法

基本观点

C. J. Fillmore 指出：诸如主语、宾语等语法关系实际上都是表层结构上的概念，在语言的底层，所需要的不是这些表层的语法关系，而是用施事、受事、工具、受益等概念所表示的句法语义关系。这些句法语义关系，经各种变换之后才在表层结构中成为主语或宾语。

格的定义

格语法中的格是“深层格”，它是指句子中体词(名词、代词等)和谓词(动词、形容词等)之间的及物性关系(transitivity)，如：动作和施事者的关系、动作和受事者的关系等，这些关系是语义关系，它是一切语言中普遍存在的现象。

格语法的三条基本规则

(1) $S \rightarrow M + P$

句子 S 可以改写成情态(Modality)和命题(Proposition)两大部分

(2) $P \rightarrow V + C_1 + C_2 + \dots + C_n$

命题 P 都可以改写成一个动词V 和若干个格 C。

(3) $C \rightarrow K + NP$

K 为格标，是各种格范畴在底层结构中的标记，可以有各种标记形式

格表

- (1) 施事格(Agentive): 动作的发生者;
- (2) 工具格(Instrumental): 对动作或状态而言作为某种因素而牵涉到的无生命的力量或客体。
- (3) 承受格(Dative): 由动词确定的动作或状态所影响的有生物。如, He is tall.
- (4) 使成格(Factitive): 由动词确定的动作或状态所形成的客体或有生物。或理解为：动词意义的一部分的客体或有生物。如: John dreamed about Mary.
- (5) 方位格(Locative): 由动词确定的动作或状态的处所或空间方位。如: He is in the house
- (6) 客体格(Objective): 由动词确定的动作或状态所影响的事物。如: He bought a book.
- (7) 受益格(Benefactive): 由动词确定的动作为之服务的有生命的对象。如: He sang a song for Mary.
- (8) 源点格(Source): 由动词确定的动作所作用到的事物的来源或发生位置变化过程中的起始位置。如: He bought a book from Mary.
- (9) 终点格(Goal): 由动词确定的动作所作用到的事物的终点或发生位置变化过程中的终端位置。如: I sold a car to Mary.
- (10) 伴随格(Comitative): 由动词确定的与施事共同完成动作的伴随者。如: He sang a song with Mary.

格语法描写汉语的局限性

汉语的一些无动句、流水句、连动句、紧缩、动补、省略等结构，无法或不必用一个统率全句的模式来描述，其中连动句和兼语句尤为突出。

语义网络

语义网络的概念

语义网络通过由概念和语义关系组成的有向图来表达知识、描述语义。

- 有向图：图的结点表示概念，图的边表示概念之间的关系。
- 边的类型：(1)“是一种”：A到B的边表示“A是B的一种特例”；(2)“是部分”：A到B的边表示“A是B的一部分”；……

事件的语义关系

- (1) 分类关系：事物之间的类属关系。
- (2) 聚焦关系：多个下位概念构成一个上位概念。
- (3) 推论关系：由一个概念推出另一个概念。
- (4) 时间、位置关系：事实发生或存在的时问、位置。

词义
 {
 内涵: 词本身的意义，是对词代表的概念描述。
外延: 词所指代的物体。

概念依存理论

CD 理论的组成:

- 三个层次之一: **动作基元**
 - (1) 在概念依存层次: 规定了一组动作基元, 其他动作是由这些动作基元组合而成的。如: 抓(Grasp)、移动(Move)、传送(Trans)、去(Go)、推(Propel)、吸收(Ingest)、撞击(Hit)等。
 - (2) 关于精神世界的概念: 心传(MTrans)、概念化(Conceptualize)、心建(MBuild)。
 - (3) 关于手段或工具: 闻(Smell)、看(Look-at)、听(Listen-to)、说(Speak)。
- 三个层次之二: **剧本**
用来描写遇到一些常见场景或场合时所采取的一些固定成套的动作。
- 三个层次之三: **计划**
计划中的每一步都是一个剧本

◆依据CD 理论理解语言

一般文章中一些动作的细节被很多处理方法忽略, 计算机难以发现事件、人物、地点等各种指代之间的联系, 而 CD 理论试图建立这种联系, 正确描述常识, 并利用基本动作推理。

该理论对限定领域内的特定应用比较有效。

缺陷: 对常识的描写过于刻板和定式。

词义消歧

基本方法

- 早期基于规则的消歧方法
- 统计机器学习消歧方法
 - 有监督学习方法
 - 无监督学习方法

基本思路: 一个词的不同语义一般发生在不同的上下文中。

- 基于词典信息的消歧方法

有监督的词义消歧方法

总体思路: 通过建立分类器, 利用划分多义词的上下文类别的方法来区分多义词的词义。

- 基于互信息的消歧方法(Brown et al., 1991)

基本思想：假设我们有一个双语对齐的平行语料库，以法语和英语为例，通过词语对齐模型每个法语单词可以找到对应的英语单词，一个多义的法语单词在不同的上下文中对应多种不同的英语翻译。

利用 **Flip-Flop 算法** 来解决指示器分类问题(假设多义法语词只有两个语义)：

- 基于贝叶斯分类器的词义消歧方法
- 基于最大熵的词义消歧方法

基于词典的词义消歧方法

(1) 基于语义定义的消歧

基本思想：词典中词条本身的定义作为判断其语义的条件。

(2) 基于义类辞典(thesaurus) 的消歧

基本思想：多义词的不同义项在使用时往往具有不同的上下文语义类，即通过上下文的语义范畴可以判断多义词的使用义项。

3) 基于双语词典的消歧

基本思想：需要消歧的语言称为第一语言，把需要借助的另一种语言称为第二语言。建立多义词 x 与相关词 y 之间的搭配关系，然后，在第二种语言的语料库中统计对应 x 不同词义的翻译与相关词 y 的翻译同现的次数，同现次数高的搭配对应的义项即为消歧后的词义。

(4) Yarowsky 消歧算法

基本思想：基于词典的词义消歧算法都是分别处理每个出现的歧义词，且对歧义词有两个限制：

- 每篇文本只有一个意义：在任意给定的文本中，目标词的词义具有高度的一致性；
- 每个搭配只有一个意义：目标词和周围词之间的相对距离、词序和句法关系，为目标词的意义提供了很强的一致性的词义消歧线索。

无监督的词义消歧方法

与(Gale, 1992) 方法类似，对于一个具有 k 个义项的词 w ，估计使用义项 s_i ($k \geq i \geq 1$) 的上下文中出现词 v_j 的概率，即 $p(v_j | s_i)$ 。

H. Schütze (1998) 提出的上下文分组辨识 (context-group discrimination) 方法是无监督的词义消歧方法的典型代表。

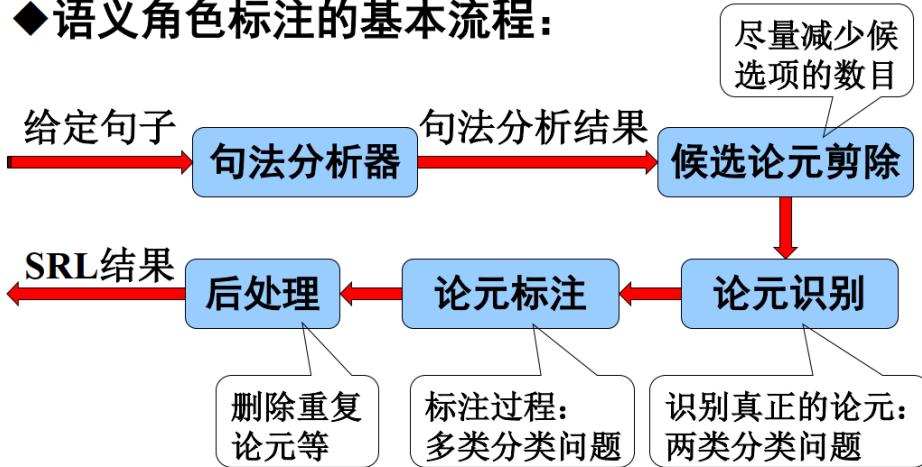
但是，在该方法中参数 $p(v_j | s_i)$ 的估计不是根据有标注的训练语料，而是在无标注的语料上进行，开始时随机地初始化参数，然后根据EM算法重新估计该概率值。

主要问题在于，很多同义词的同一个意义出现的上下文往往有很大的差异，因此，很难保证同一个意义的上下文被划分到同一个等价类中。

语义角色标注

语义角色标注一般是在句法分析的基础上进行的。

◆语义角色标注的基本流程：



机器翻译

机器翻译 (machine translation, MT) 是用计算机把一种语言(源语言, source language) 翻译成另一种语言(目标语言, target language) 的一门学科和技术。



机器翻译技术大体上可以分为三种方法，分别为基于规则的机器翻译、统计机器翻译以及神经机器翻译。

第一代机器翻译技术是主要使用基于规则的机器翻译方法，其主要思想是通过形式文法定义的规则引入源语言和目标语中的语言学知识。

统计机器翻译兴起于上世纪 90 年代[9, 20]，它利用统计模型从单/双语语料中自动学习翻译知识。具体来说，可以使用单语语料学习语言模型，使用双语平行语料学习翻译模型，并使用这些统计模型完成对翻译过程的建模。整个过程不需要人工编写规则，也不需要从实例中构建翻译模板。无论是词还是短语，甚至是句法结构，统计机器翻译系统都可以自动学习。人更多的是定义翻译所需的特征和基本翻译单元的形式，而翻译知识都保存在模型的参数中。

随着机器学习技术的发展，基于深度学习的神经机器翻译逐渐兴起。自 2014 年开始，它在短短几年内已经在大部分任务上取得了明显的优势[21, 22, 23, 24, 25]。在神经机器翻译中，词串被表示成实数向量，即分布式向量表示。这样，翻译过程并不是在离散化的单词和短语上进行，而是在实数向量空间上计算。因此与之前的技术相比，它在词序列表示的方式上有着本质的改变。通常，机器翻译可以被看作一个序列到另一个序列的转化。在神经机器翻译中，序列到序列的转化过程可以由编码器-解码器 (Encoder-Decoder) 框架实现。其中，编码器把源语言序列进行编码，并提取源语言中的信息进行分布式表示，之后解码器再把这种信息转换为另一种语言的表达。

机器翻译的困难

- 自然语言中普遍存在的歧义和未知现象
- 机器翻译不仅仅是字符串的转换
- 机器翻译的解不唯一，而且始终存在的人为的标准

基本翻译方法

- ◆ **直接转换法**
- ◆ **基于规则的翻译方法**
- ◆ **基于中间语言的翻译方法**
- ◆ **基于语料库的翻译方法**
 - **基于事例的翻译方法**
 - **统计翻译方法**
 - **神经网络机器翻译**

直接转换法

从源语言句子的表层出发，将单词、短语或句子直接置换成目标语言译文，必要时进行简单的词序调整。对原文句子的分析仅满足于特定译文生成的需要。这类翻译系统一般针对某一个特定的语言对，将分析与生成、语言数据、文法和规则与程序等都融合在一起。

基于规则的翻译方法

1957年美国学者V. Yingve在《句法翻译框架》(Framework for Syntactic Translation)一文中提出了对源语言和目标语言均进行适当描述、把翻译机制与语法分开、用规则描述语法的实现思想，这就是基于规则的翻译方法。

基于规则的翻译过程分成6个步骤：

- (a) 对源语言句子进行词法分析
- (b) 对源语言句子进行句法/语义分析
- (c) 源语言句子结构到译文结构的转换
- (d) 译文句法结构生成
- (e) 源语言词汇到译文词汇的转换
- (f) 译文词法选择与生成

由于基于规则的翻译方法执行过程为：“独立分析 - 独立生成 - 相关转换”因此，又称基于转换的翻译方法。

对基于规则的翻译方法的评价：

优点：可以较好地保持原文的结构，产生的译文结构与原文的结构关系密切，尤其对于语言现象已知的或句法结构规范的源语言语句具有较强的处理能力和较好的翻译效果。

弱点：规则一般由人工编写，工作量大，主观性强，一致性难以保障，不利于系统扩充，对非规范语言现象缺乏相应的处理能力。

基于中间语言的翻译方法

方法：输入语句→中间语言→翻译结果

· **代表系统：**JANUS (CMU) 早期版本

- 源语言解析器
- 比较准确的中间语言(Interlingua)
- 目标语言生成器(Target Language Generator)

对基于中间语言的翻译方法评价：

优点：中间语言的设计可以不考虑具体的翻译语言对，因此，该方法尤其适合多语言之间的互译。

弱点：如何定义和设计中间语言的表达方式，以及如何维护并不是一件容易的事情，中间语言在语义表达的准确性、完整性等很多方面，都面临若干困难。

基于语料库的翻译方法

基于事例的翻译方法

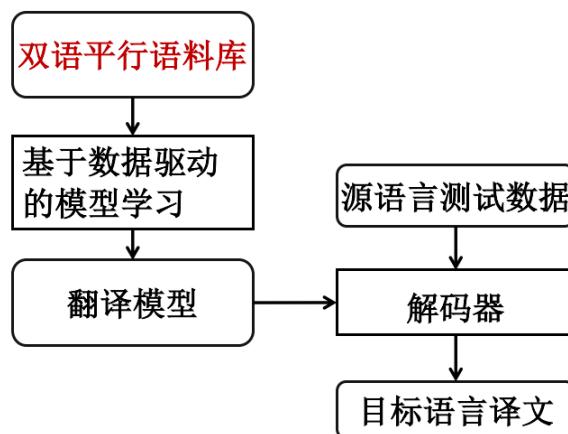
- 方法：输入语句→与事例相似度比较→翻译结果
- 资源：大规模事例库
- 代表系统：ATR-MATRIX (ATR, Japan)

对基于实例的翻译方法评价：

优点：不要求源语言句子必须符合语法规定，翻译机制一般不需要对源语言句子做深入分析。

弱点：两个不同的句子之间的相似性（包括结构相似性和语义相似性）往往难以把握，尤其在口语中，句子结构一般比较松散，成分冗余和成分省略都较严重，这更增加了分析句子与事例句子的比较难度。另外，系统往往难以处理事例库中没有记录的陌生的语言现象，而且当事例库达到一定规模时，其事例检索的效率较低。

统计翻译方法



噪声信道模型

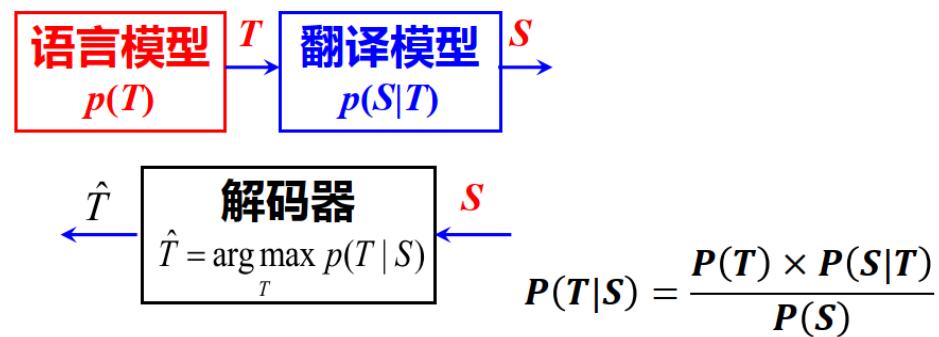
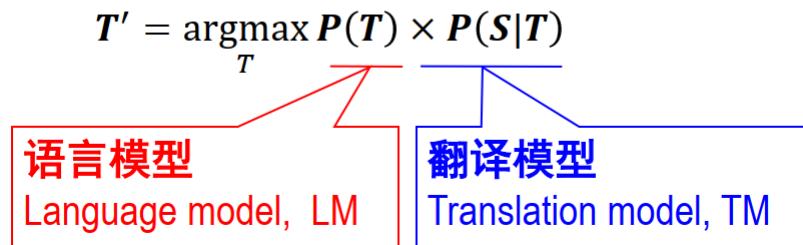
一种语言T由于经过一个噪声信道而发生变形，从而在信道的另一端呈现为另一种语言S(信道意义上的输出，翻译意义上的源语言)。翻译问题实际上就是如何根据观察到的S，恢复最为可能的T问题。这种观点认为，任何一种语言的任何一个句子都有可能是另外一种语言中的某个句子的译文，只是可能有大有小[Brown et. al, 1990]。



源语言句子: $S = s_1^m = s_1 s_2 \cdots s_m$

目标语言句子: $T = t_1^l = t_1 t_2 \cdots t_l$

贝叶斯公式: $P(T|S) = \frac{P(T) \times P(S|T)}{P(S)}$



统计翻译中的三个关键问题:

(1) 估计语言模型概率 $p(T)$;

◆ 估计语言模型概率 $p(T)$

给定句子: $T = t_1^l = t_1 t_2 \cdots t_l$

句子概率: $P(T) = P(t_1)P(t_2|t_1) \cdots P(t_l|t_1 t_2 \cdots t_{l-1})$

(2) 估计翻译概率 $p(S|T)$;

不妨，我们用 $\mathcal{A}(S, T)$ 表示源语言句子 S 与目标语言句子 T 之间所有对位关系的集合。在目标语言句子 T 的长度（单词的个数）为 l ，源语言句子 S 的长度为 m 的情况下， T 和 S 的单词之间有 $2^{l \times m}$ 种不同的对位关系。

$$|\mathcal{A}(S, T)| = 2^{l \times m} \quad A(S, T) \in \mathcal{A}(S, T)$$

用来刻画这些对应关系 $A(S, T)$ 的模型叫做对位模型 (alignment model)。

将对位模型 A 视为隐含变量，则：

$$P(S|T) = \sum_A P(S, A|T)$$

按照约定，源语言句子 $S = s_1^m = s_1 s_2 \cdots s_m$ 有 m 个单词
目标语言句子 $T = t_1^l = t_1 t_2 \cdots t_l$ 有 l 个单词
每一种对位序列表示成：

$$A = a_1^m = a_1 a_2 \cdots a_m \quad a_j \in [0, 1, \dots, l]$$

$a_j=i$ 表示从 S 的第 j 个词 到 T 的第 i 个词 的 对位关系

◆ 翻译概率 $P(S|T)$ 的计算

$$P(S|T) = \sum_A P(S, A|T)$$

→ $P(S, A|T)$?

$$P(S, A|T) = p(m|T) \times \underbrace{P(A|T, m)}_{\text{对位模型}} \times \underbrace{P(S|T, A, m)}_{\text{词汇翻译模型}}$$

(3) 快速有效地搜索 T 使得 $p(T) \times p(S | T)$ 最大。

基于词的机器翻译建模

模型	假设	参数训练	简评
IBM1	翻译模型仅与单词间的直译概率有关，句长概率和对齐概率都是均匀分布。	应用EM算法，从双语语料库中训练获得，可以得到全局最优参数，与初始值无关。	模型简单、易于实现，但仅考虑了单词的影响，没有考虑词序的影响。
IBM2	翻译模型和句长模型同IBM1，对位概率为0阶对齐	应用EM算法，从双语语料库中训练获得，只能收敛到局部最优。	模型简单、易于实现，同时考虑了单词和词序的影响。
IBM3	翻译模型依赖于繁衍率模型和单词间的直译概率，对齐概率取0阶词对齐。	需要首先应用模型IBM1或IBM2对双语语料进行单词级对位，然后训练繁衍概率参数。	引入了描述单词间一对多情况的繁衍概率，参数较多，实现过程较复杂。
IBM4	翻译模型依赖于单词间的直译概率繁衍概率、词类、语言片断中心位置和语言片断内相对位置等因素对齐概率取1阶词对齐。	需要首先应用模型IBM1~IBM3对双语语料进行单词级对位和语言片断划分，然后训练两种位置概率参数。	不仅考虑了一对多的情况，还将语言片断作为一个整体进行考虑。参数较多、不易实现。
IBM5	翻译模型依赖于直译概率、繁衍概率、语言片断中心位置、语言片断内相对位置和对位的历史等因素。	需要在模型IBM1~IBM4参数训练的基础上获得参数	对IBM4进行了修正，同时考虑了当前对位信息和对位历史。模型的表现力最强，但过于复杂，实用性不强。
HMM	句长模型和翻译模型同IBM1，对齐模型为1阶对齐。	应用EM算法，从双语对照语料中训练获得。	模型简单，易于实现，考虑了词序的影响。

基于短语的翻译模型

- 基于词的翻译模型的问题：
 - 很难处理词义消歧问题
 - 很难处理一对多、多对一和多对多的翻译问题

不难发现，基于单词的模型并不能很好地捕捉单词间的搭配关系。相比之下，使用更大颗粒度的翻译单元是一种对搭配进行处理的方法。下面来一起看看，基于单词的模型所产生的问题以及如何使用基于短语的模型来缓解该问题。

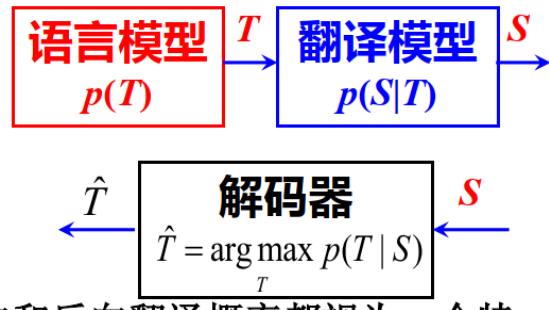
实际上，单词本身也是一种短语。从这个角度说，基于单词的翻译模型是包含在基于短语的翻译模型中的。而这里所说的短语包括多个连续的单词，可以直接捕捉翻译中的一些局部依赖。而且，由于引入了更多样的翻译单元，可选择的翻译路径数量也大大增加。本质上，引入更大颗粒度的翻译单元给模型增加了灵活性，同时增大了翻译假设空间。如果建模合理，更多的翻译路径会增加找到高质量译文的机会。在7.2节还将看到，基于短语的模型会从多个角度对翻译问题进行描述，包括基础数学建模、调序等等。

基于最大熵的方法(判别式)

生成式模型转向判别式模型

➤ 问题:

- 反向的翻译模型用噪声信道模型无法解释。



➤ 解决方法:

- 将语言模型概率、正向和反向翻译概率都视为一个特征，采用最大熵方法建模

最大熵方法的基本思想

➤ 任务

➤ 对于一个随机事件，假设已经有了一组样例，我们希望建立一个统计模型来模拟这个随机事件的分布

➤ 目标

➤ 对于一组特征，使得统计模型在这一组特征上的模型分布与样例中的经验分布完全一致，同时不对未知事件作任何假设，即保证这个模型尽可能的“均匀”(也就是要求模型的熵值达到最大)

基于短语的翻译模型[Koehn, 2003]

$$\begin{aligned} T' &= \underset{T}{\operatorname{argmax}} P(T|S) \\ &= \underset{T, S_1^K}{\operatorname{argmax}} P(T, S_1^K | S) \\ &= \underset{T, S_1^K, T_1^K, T_1^{K'}}{\operatorname{argmax}} P(S_1^K | S) \times P(T_1^K | S_1^K, S) \times \\ &\quad P(T_1^{K'} | T_1^K, S_1^K, S) \times P(T | T_1^{K'}, T_1^K, S_1^K, S) \\ & \quad \boxed{T} \\ T' &= \underset{T}{\operatorname{argmax}} P(T|S) \\ &= \underset{T, S_1^K}{\operatorname{argmax}} P(T, S_1^K | S) \\ &= \underset{T, S_1^K, T_1^K, T_1^{K'}}{\operatorname{argmax}} P(S_1^K | S) \underset{\downarrow}{P(T_1^K | S_1^K, S)} \underset{\downarrow}{P(T_1^{K'} | T_1^K, S_1^K, S)} \underset{\downarrow}{P(T | T_1^{K'}, T_1^K, S_1^K, S)} \\ &\quad \text{短语划分模型} \quad \text{短语翻译模型} \quad \text{短语调序模型} \\ &\quad \downarrow \qquad \downarrow \qquad \downarrow \\ &\quad \text{目标语言模型} \end{aligned}$$

短语划分模型

目标：将一个词序列如何划分为短语序列

方法：一般假设每一种短语划分方式都是等概率的

剩下的三个核心模型：

1. 短语翻译模型： $P(T_1^K | S_1^K, S)$

2. 短语调序模型： $P(T_1^{K'} | T_1^K, S_1^K, S)$

3. 目标语言模型： $P(T | T_1^{K'}, T_1^K, S_1^K, S)$

短语翻译模型： $P(T_1^K | S_1^K, S)$

1. 如何学习短语翻译规则
2. 如何估计短语翻译概率

短语翻译规则抽取

算法：对于源语言句子 S 中的任一短语 S_i^j ，根据词语对齐 A 找到目标语言句子 T 中的对齐片段 $T_{i'}^{j'}$ ，若 S_i^j 与 $T_{i'}^{j'}$ 满足对齐一致性，则 $(S_i^j, T_{i'}^{j'})$ 为一条短语翻译规则。

对齐一致性： S_i^j 中每个词 S_k ，若 $(k, k') \in A$ ，则 $i' \leq k' \leq j'$ ， $T_{i'}^{j'}$ 中每个词 $T_{t'}^{j'}$ ，若 $(t, t') \in A$ ，则 $i \leq t \leq j$ 。

短语翻译概率估计：4个翻译概率（最大似然）

1. 正向、逆向短语翻译概率 $p(t|s), p(s|t)$
2. 正向、逆向词汇化翻译概率 $p_{lex}(t|s), p_{lex}(s|t)$

短语调序模型： $(T_1^{K'} | T_1^K, S_1^K, S)$

两种常用方法：

1. 距离跳转模型

2. 分类模型

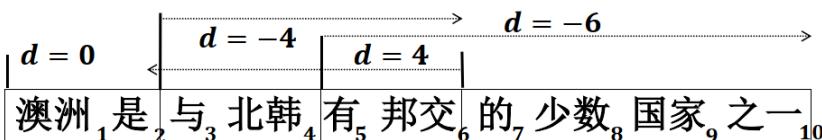
11.2.6 基于短语的翻译模型

◆ 距离跳转模型

当前翻译的源语言短语的第一个词的下标

前一个翻译的源语言短语的最后一个词的下标

$$d = next_{begin} - last_{end} - 1$$



第 <i>i</i> 个源短语	短语跨度	距离跳转	距离
1	1-2	句子开始	0
2	7-10	跳过3-6	+4
3	5-6	向前跳过5-10	-6
4	3-4	向前跳过3-6	-4

◆ 分类模型: Monotone (M) Swap (S) Discontinuous (D)

s	澳洲 ₁ 是 ₂ 与 ₃ 北韩 ₄ 有 ₅ 邦交 ₆ 的 ₇ 少数 ₈ 国家 ₉ 之一 ₁₀	
t	Australia is one of the few countries that have diplomatic relations with North Korea	M 1-0 D S 6-7 S 4-5

$$M: next_{begin} - last_{end} = 1$$

$$S: next_{end} - last_{begin} = -1 \quad \text{MaxEnt Classifier}$$

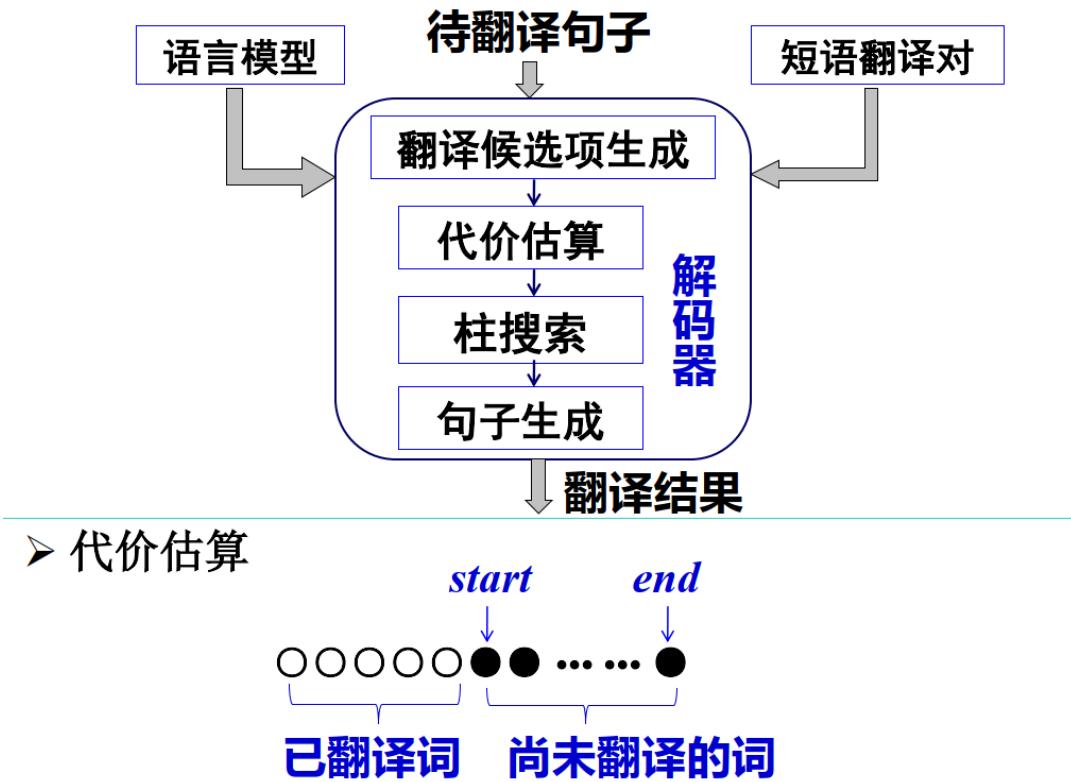
$$D: next_{begin} - last_{end} \neq 1 \text{ and } next_{end} - last_{begin} \neq -1$$

基于短语翻译模型8特征(以汉英翻译为例):

- ✓ 短语翻译概率 $\log p(e|c)$
- ✓ 词汇化的短语翻译概率 $\log p_{lex}(e|c)$
- ✓ 反向的短语翻译概率 $\log p(c|e)$
- ✓ 反向的词汇化短语翻译概率 $\log p_{lex}(c|e)$
- ✓ 短语调序模型 $\log P(E_1^{K'} | E_1^K, C_1^K, C)$
- ✓ 基于n-gram 的英语语言模型 $\log P(E | E_1^{K'}, E_1^K, C_1^K, C)$
- ✓ 英语句子长度惩罚 $\log \text{len}(E)$
- ✓ 汉语短语个数惩罚 $\log K$

基于短语的翻译模型的解码算法

解码算法取决于翻译模型。在基于短语的翻译系统中：



$$Score = \text{已翻译词所耗费的代价} + \text{未翻译部分的估算代价}$$

- ★ 已翻译词所耗费的代价: 已翻译词的模型概率
- ★ 翻译未来词估计需要的代价: 最大概率或其他因素

➤ 最大概率:

- ✓ 尚未翻译部分的正向短语翻译概率 ($\text{汉} \rightarrow \text{英}$);
- ✓ 尚未翻译部分的逆向短语翻译概率 ($\text{英} \rightarrow \text{汉}$);
- ✓ 尚未翻译部分的正向词汇化翻译概率 ($\text{汉} \rightarrow \text{英}$);
- ✓ 尚未翻译部分的逆向词汇化翻译概率 ($\text{英} \rightarrow \text{汉}$)。

$$TP(f_{start}^{end}) = \max \sum \lambda_i \times \log(p_i(e|f))$$

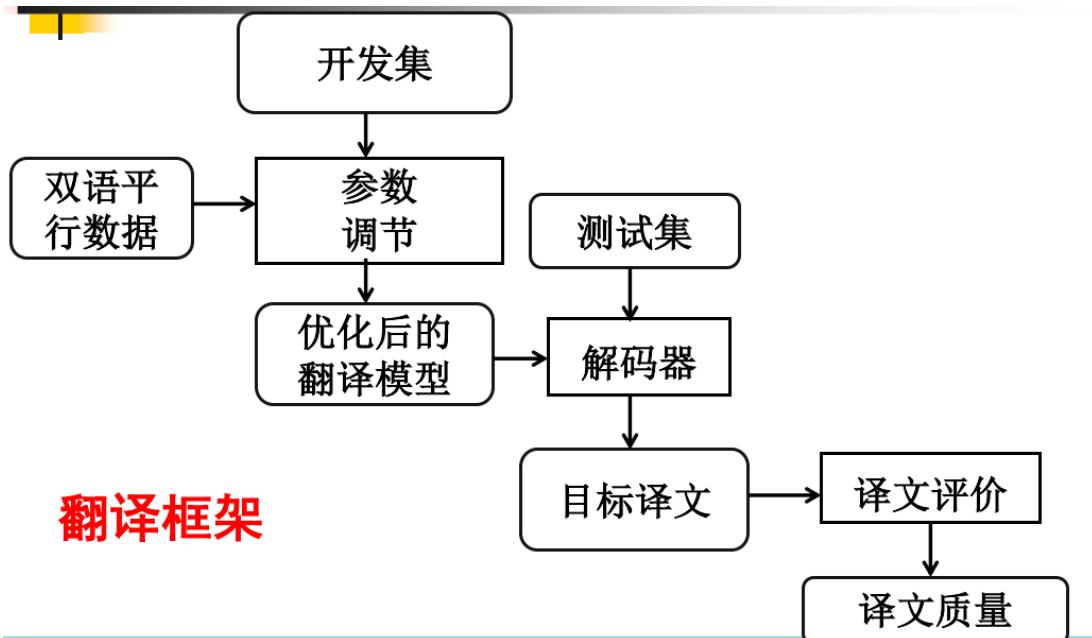
➤ 其他因素:

- ✓ 译文短语的长度、语言模型概率等。

柱搜索(beam search) [Ney, 1992; Tillmann, 2003]

基本思想：给定一个输入句子，生成对应的短语序列，每个短语对应一组翻译候选。短语序列按从左到右的顺序或目标短语生成的先后顺序搜索最可能的翻译假设(hypothesis)。

采用适当的剪枝策略。



基于层次化短语的翻译模型

问题提出：

- (1) 基于短语的翻译模型能够比较鲁棒地翻译较短的子串，当短语长度扩展到3个以上的单词时，翻译系统的性能提高很少，短语长度增大以后，数据稀疏问题变得非常严重。
- (2) 在很多情况下简单的短语翻译模型无法处理短语之间（尤其是长距离）的调序。
- (3) 基于短语翻译模型无法处理非连续短语翻译现象，例如（在 ... 时，when ...）

基于层次短语的翻译过程同步进行双语解析，所使用的同步上下无关文法是从没有做任何句法信息标注的双语对照语料中自动学习获得的。

1. 层次短语翻译规则学习
2. 层次短语模型解码过程

基于层次短语的模型 (Hierarchical Phrase-based Model) 是一个经典的统计机器翻译模型[88, 336]

。

这个模型可以很好地解决短语系统对翻译中长距离调序建模不足的问题。

层次短语模型的核心是把翻译问题归结为两种语言词串的同步生成问题。实际上，词串的生成问题是自然语言处理中的经典问题，早期的研究更多的是关注单语句子的生成，比如，如何使用句法树描述一个句子的生成过程。层次短语模型的创新之处是把传统单语词串的生成推广到双语词串的同步生成上。这使得机器翻译可以使用类似句法分析的方法进行求解。

树翻译模型

- ◆ 树到串的翻译模型
- ◆ 树到树的翻译模型
- ◆ 串到树的翻译模型

问题提出：

- (1) 基于层次短语的翻译模型只使用一个非终结符X，过于泛化。
- (2) 基于层次短语的翻译模型在处理长距离的短语调序问题时能力有限。

树到串的翻译模型

Liu et al. (2006), Huang et al. (2006) 提出树到串的翻译模型。

- 句法分析：将源语言句子分析为一棵句法结构树（短语结构树）
 - 树到串的转换：递归地将源语言句子的句法结构树转换为目标语言句子
1. 树到串翻译规则抽取：给定源语言和目标语言的双语平行句对（经过词语对齐、源语言端句法分析），抽取满足词语对齐的树到串翻译规则
2. 确定满足词语对齐的树节点：源语言句法树节点所能到达的目标语言子串与该树节点覆盖的源语言子串满足词语对齐约束。
3. 对于每个满足词语对齐的树节点，我们可以抽取一条最小规则。

- 树到串模型的优势
 - 搜索空间小、解码效率高
 - 句法分析质量较高的前提下
- 树到串模型的不足
 - 强烈依赖于源语言句法分析的质量
 - 利用源语言端句法结构精确匹配严重
 - 没有使用任何目标语言句法知识目标译文符合文法

树到树的翻译模型

Zhang et al. (2007, 2008) 提出了树到树的翻译模型。

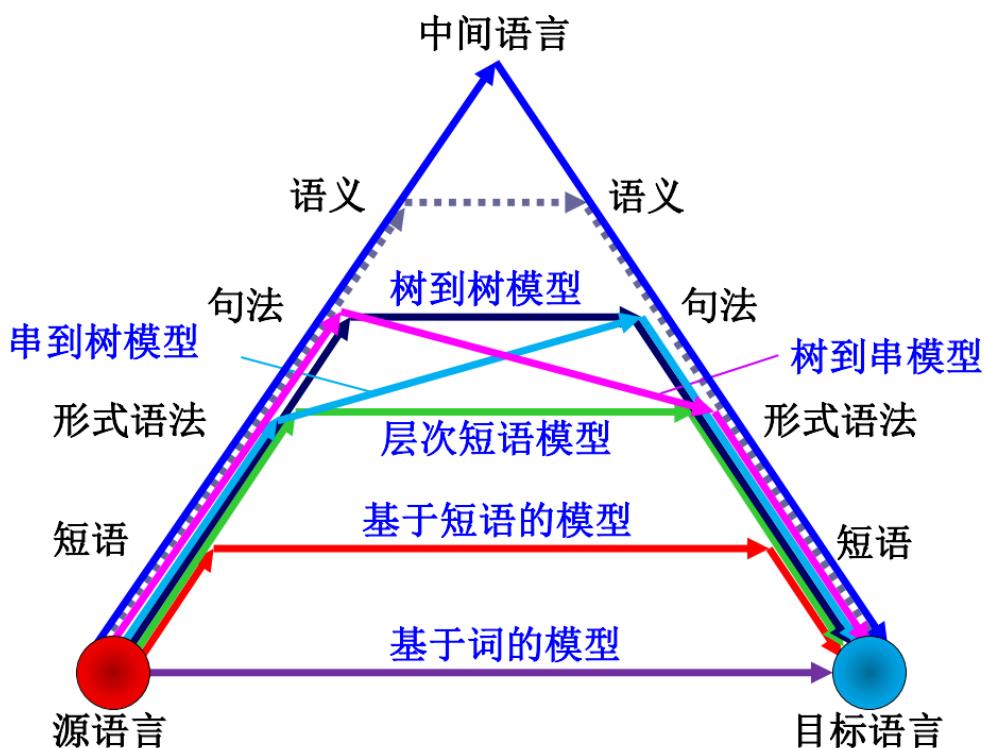
- 句法分析：将源语言句子分析为一棵句法结构树（短语结构树）
 - 树到树的转换：递归地将源语言句子的句法结构树转换为目标语言句子的句法结构树，拼接叶结点得到译文。
- 解码算法：对于源语言句法结构树，自底往上或自顶往下考虑每个节点，为每个节点搜索能够匹配的树到树翻译规则。至所有节点匹配完毕，得到最佳译文。
- 确定满足词语对齐的树节点：源语言句法树节点所覆盖的源语言子串与目标语言句法树节点所覆盖的目标语言子串满足词语对齐约束。为每对节点抽取树到树翻译规则。
- 树到树模型的优势
 - 搜索空间小、解码效率高
 - 树到树模型的不足
 - 强烈依赖于源语言和目标语言句法分析的质量
 - 利用两端句法结构精确匹配严重
 - 翻译质量差

串到树的翻译模型

Galley et al. (2004, 2006), Marcu et al. (2006) 提出了串到树的翻译模型。

- 串到树的转换：利用串到树转换规则，将源语言句子分析为一棵目标语言句法结构树，拼接叶结点得到译文。
- 串到树翻译规则抽取：给定源语言和目标语言的双语平行句对（经过词语对齐、目标语言端句法分析），抽取满足词语对齐的串到树翻译规则
- 确定满足词语对齐的树节点：目标语言句法树节点所能到达的源语言子串与该树节点覆盖的目标语言子串满足词语对齐约束
- 串到树模型的优势
 - 搜索空间大，保证译文符合文法，翻译质量高

- 串到树模型的不足
 - > 解码速度受限
 - > 未使用源语言端句法知识，存在词义消歧问题



系统融合

◆ 系统融合方法：

(1) 句子级系统融合

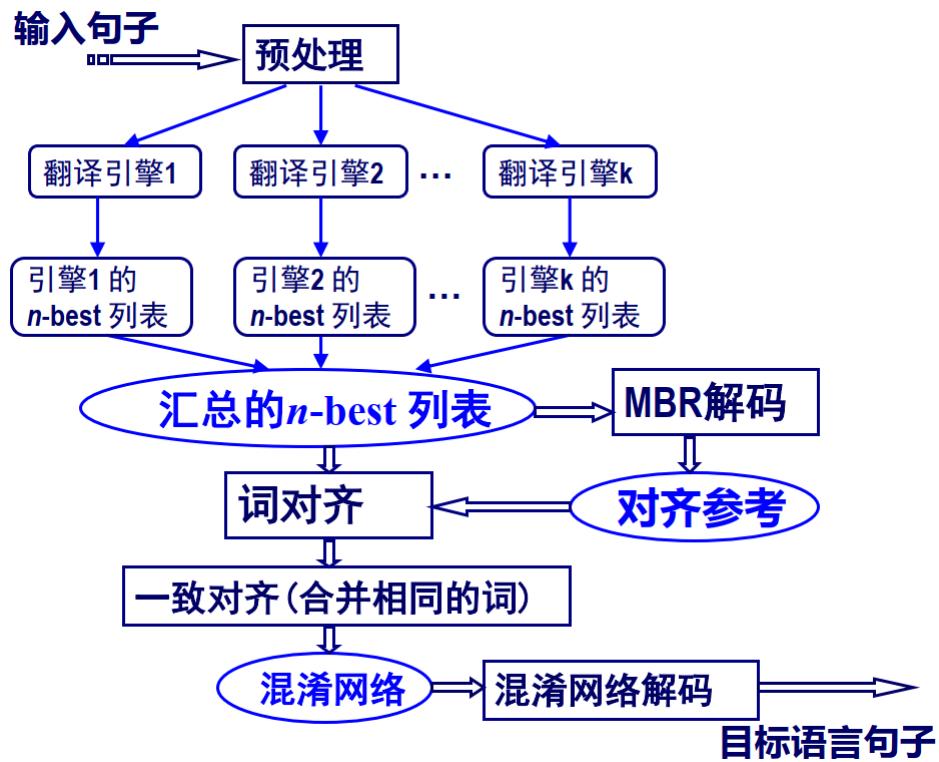
针对同一个源语言句子，利用最小贝叶斯风险解码或重打分方法比较多个机器翻译系统的译文输出，将最优的翻译结果作为最终的一致翻译结果。

(2) 短语级系统融合

利用多个翻译系统的输出结果，重新抽取短语翻译规则集合，并利用新的短语翻译规则进行重新解码。

(3) 词语级系统融合

首先将多个翻译系统的译文输出进行词语对齐，构建一个混淆网络，对混淆网络中的每个位置的候选词进行置信度估计，最后进行混淆网络解码。



译文评估方法

◆ 常用的评测指标

> 主观评测：(1)流畅度；(2)充分性；(3)语义保持性。

◆ 客观评测

(1) **句子错误率**：译文与参考答案不完全相同的句子为错误句子。错误句子占全部译文的比率。

(2) **单词错误率**(Multiple Word Error Rate on Multiple Reference, 记作 mWER)：分别计算译文与每个参考译文的编辑距离，以最短的为评分依据，进行归一化处理

(3) **与位置无关的单词错误率**(Position independent mWER, 记作mPER)：不考虑单词在句子中的顺序

(4) **METEOR** 评测方法

对候选译文与参考译文进行词对齐，计算词汇完全匹配、词干匹配、同义词匹配等各种情况的准确率(P)、召回率(R)和F平均值

(5) **BLEU**评价方法[Papineni, 2002] - BiLingual Evaluation Understudy, IBM

> 基本思想：将机器翻译产生的候选译文与人翻译的多个参考译文相比较，越接近，候选译文的正确率越高。

> 实现方法：统计同时出现在系统译文和参考译文中的n元词的个数，最后把匹配到的n元词的数目除以系统译文的n元词数目，得到评测结果。

(6) **NIST** 评测方法 National Institute of Standards and Technology

> BLEU评分公式中采用的n元语法同现概率的几何平均方法使评分值对于各种n元语法同现的比例具有相同的敏感性，但实际上，这种做法存在着潜在的矛盾，因为n值较大的统计单元出现的概率较低。

> 基本思想：因此，NIST的研究人员提出了另外一种处理方法，就是用n-gram同现概率的算术平均值取代几何平均值。另外，如果一个n元词在参考译文中出现的次数越少，表明它所包含的信息量越大，那么，它对于该n元词就赋予更高的权重。

◆ 统计翻译中的译文错误

(1) 模型错误：概率最高的译文不是正确的

(2) 搜索错误：概率最高的译文是正确的，但搜索算法找不到。这类错误大约占5%。

神经网络机器翻译

单词表示模型

在神经语言建模中，每个单词都会被表示为一个实数向量。这对应了一种单词的表示模型。下面就来看看传统的单词表示模型和这种基于实数向量的单词表示模型有何不同。

One-hot 编码

One-hot 编码（也称独热编码）是传统的单词表示方法。One-hot 编码把单词表示为词汇表大小的 0-1 向量，其中只有该词所对应的那一项是 1，而其余所有项都是 0。举个简单的例子，假如有一个词典，里面包含 10k 个单词，并进行编号。那么每个单词都可以表示为一个 10k 维的 One-hot 向量，它仅在对应编号那个维度为 1，其他维度都为 0，如图9.45所示。

$\cos(\text{'桌子'}, \text{'椅子'}) = 0$		
	桌子	椅子
你 ₁	[0]	[0]
桌子 ₂	[1]	[0]
他 ₃	[0]	[0]
椅子 ₄	[0]	[1]
我们 ₅	[0]	[0]
...	[...]	[...]
你好 _{10k}	[0]	[0]

图 9.45 单词的 One-hot 表示

One-hot 编码的优点是形式简单、易于计算，而且这种表示与词典具有很好的对应关系，因此每个编码都可以进行解释。但是，One-hot 编码把单词都看作是相互正交的向量。这导致所有单词之间没有任何的相关性。只要是不同的单词，在 One-hot 编码下都是完全不同的东西。比如，大家可能会期望诸如“桌子”和“椅子”之类的词具有一些相似性，但是 One-hot 编码把它们看作相似度为 0 的两个单词。

分布式表示

神经语言模型中使用的是一种分布式表示。在神经语言模型里，每个单词不再是完全正交的 0-1 向量，而是在多维实数空间中的一个点，具体表现为一个实数向量。很多时候，也会把单词的这种分布式表示叫做词嵌入。

单词的分布式表示可以被看作是欧式空间中的一个点，因此单词之间的关系也可以通过空间的几何性质进行刻画。如图9.46所示，可以在一个 512 维空间上表示不同的单词。在这种表示下，“桌子”与“椅子”之间是具有一定的联系的。

$\cos(\text{'桌子'}, \text{'椅子'}) = 0.5$		
	桌子	椅子
属性 ₁	[0.1]	[1]
属性 ₂	[-1]	[2]
属性 ₃	[2]	[0.2]
...	[...]	[...]
属性 ₅₁₂	[0]	[-1]

图 9.46 单词的分布式表示(词嵌入)

那么，分布式表示中每个维度的含义是什么？可以把每一维度都理解为一种属性，比如一个人的身高、体重等。但是，神经网络模型更多的是把每个维度看作是单词的一种抽象“刻画”，是一种统计意义上的“语义”，而非简单的人工归纳的事物的一个个属性。使用这种连续空间的表示的好处在于，表示的内容（实数向量）可以进行计算和学习，因此可以通过模型训练得到更适用于自然语言处理的单词表示结

果。



语言模型的词嵌入是通过词嵌入矩阵进行存储的，矩阵中的每一行对应了一个词的分布式表示结果。图9.48展示了一个词嵌入矩阵的实例。

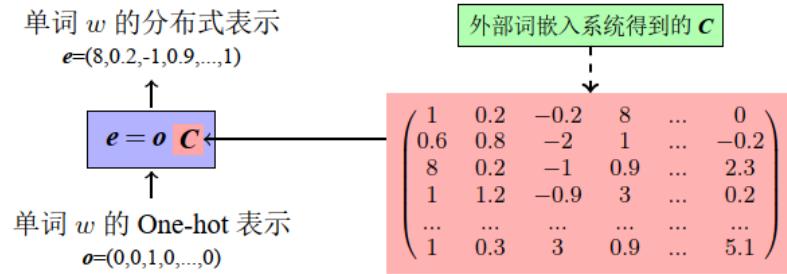
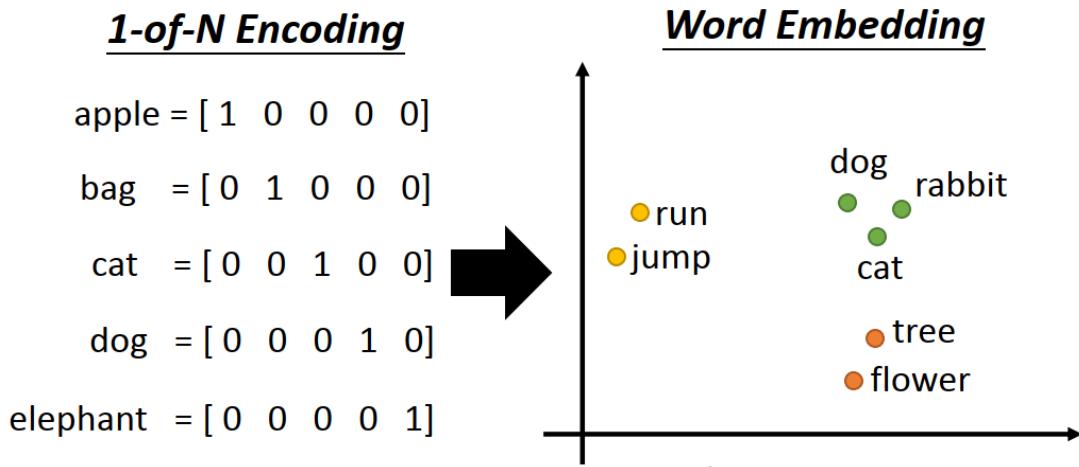


图 9.48 词嵌入矩阵 C

通常，有两种方法得到词嵌入矩阵。一种方法是把词嵌入作为语言模型的一部分进行训练，不过由于语言模型往往较复杂，这种方法非常耗时；另一种方法使用更加轻便的外部训练方法，如 word2vec[423]、Glove[166] 等。由于这些方法的效率较高，因此可以使用更大规模的数据得到更好的词嵌入结果。



word2vec

word2vec是Google在2013年开源的一款将词表征为实数值向量的高效工具。

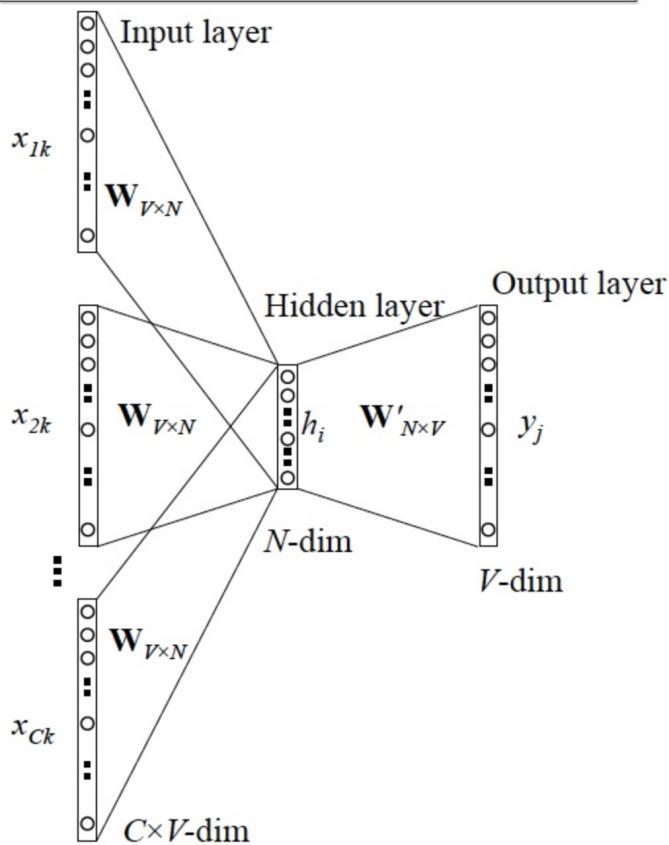
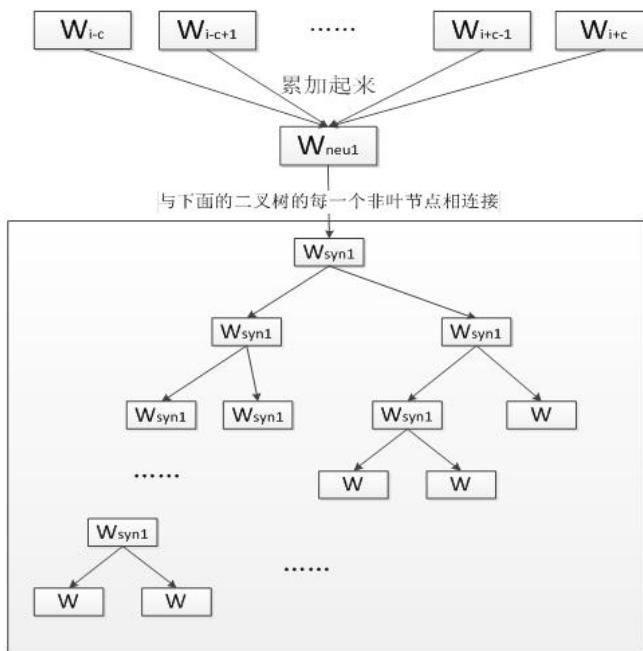
word2vec采用了CBOW(Continuous Bag-Of-Words, 连续词袋模型)和Skip-Gram两种模型。

Word2Vec模型即是一种典型的**分布编码方式**。

CBOW

连续词袋模型(Continuous Bag-of-Word Model, CBOW)是一个三层神经网络。

输入已知上下文，输出对下个单词的预测。



CBOW模型

- 第一层是输入层, 输入已知上下文的词向量.
- 中间一层称为线性隐含层, 它将所有输入的词向量累加.
- 第三层是一棵哈夫曼树, 树的叶节点与语料库中的单词一一对应, 而树的每个非叶节点是一个二分类器(一般是softmax感知机等), 树的每个非叶节点都直接与隐含层相连.
- 将上下文的词向量输入CBOW模型, 由隐含层累加得到中间向量.
- 将中间向量输入哈夫曼树的根节点, 根节点会将其分到左子树或右子树.
- 每个非叶节点都会对中间向量进行分类, 直到达到某个叶节点.
- 该叶节点对应的单词就是对下个单词的预测。

训练步骤

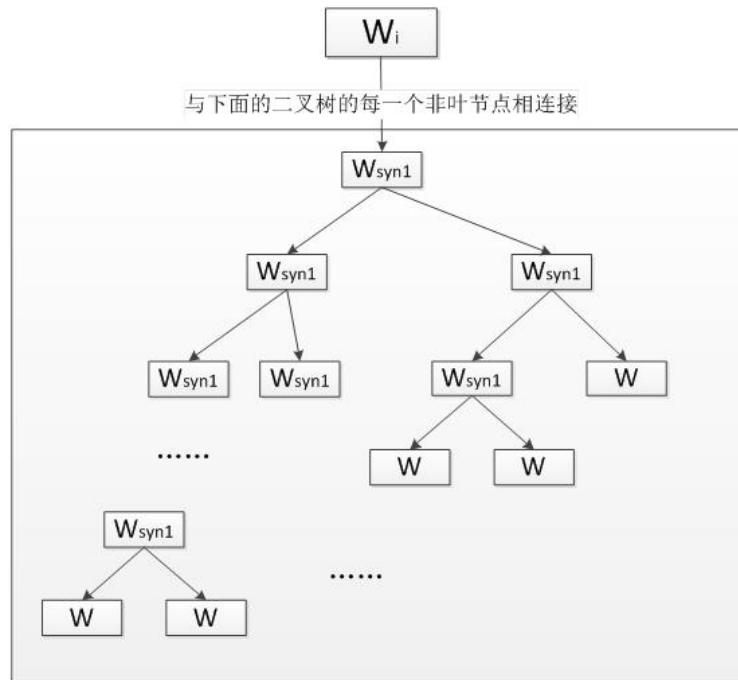
- 首先根据预料库建立词汇表, 词汇表中所有单词拥有一个随机的词向量. 我们从语料库选择一段文本进行训练.
- 将单词W的上下文的词向量输入CBOW, 由隐含层累加, 在第三层的哈夫曼树中沿着某个特定的路径到达某个叶节点, 从给出对单词W的预测.
- 训练过程中我们已经知道了单词W, 根据W的哈夫曼编码我们可以确定从根节点到叶节点的正确路径, 也确定了路径上所有分类器应该作出的预测.
- 我们采用梯度下降法调整输入的词向量, 使得实际路径向正确路径靠拢. 在训练结束后我们可以从词汇表中得到每个单词对应的词向量.

Skip-gram

Skip-gram模型同样是一个三层神经网络。

skip-gram模型的结构与CBOW模型正好相反, 每一个单词从树根开始到达叶节点可以预测出它上下文中的一一个单词, 对每个单词进行N-1次迭代, 得到对它上下文中所有单词的预测, 根据训练数据调整词向量得到足够精确的结果。

skip-gram模型输入某个单词, 输出对它上下文词向量的预测。



transfer learning

迁移学习 (Transfer Learning) 是一种机器学习的方法, 指的是一个预训练的模型被重新用在另一个任务中, 而并不是从头训练一个新的模型[547]。迁移学习的目标是将某个领域或任务上学习到的知识应用到新的领域或问题中。在机器翻译中, 可以用富资源语言的知识来改进低资源语言上的机器翻译性能, 也就是将富资源语言中的知识迁移到低资源语言中。

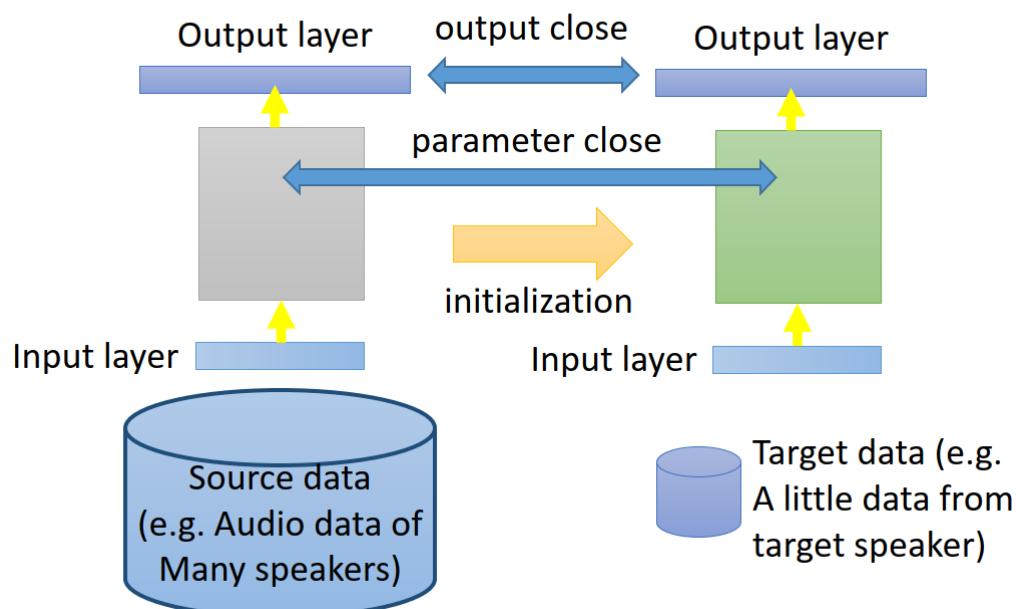
		Source Data (not directly related to the task)	
		labelled	unlabeled
Target Data	labelled	Fine-tuning Multitask Learning	Self-taught learning Rajat Raina , Alexis Battle , Honglak Lee , Benjamin Packer , Andrew Y. Ng, Self-taught learning: transfer learning from unlabeled data, ICML, 2007
	unlabeled	Domain-adversarial training Zero-shot learning	Different from semi-supervised learning Self-taught Clustering Wenyuan Dai, Qiang Yang, Gui-Rong Xue, Yong Yu, "Self-taught clustering", ICML 2008

Model Fine-tuning (labelled source, labelled target)

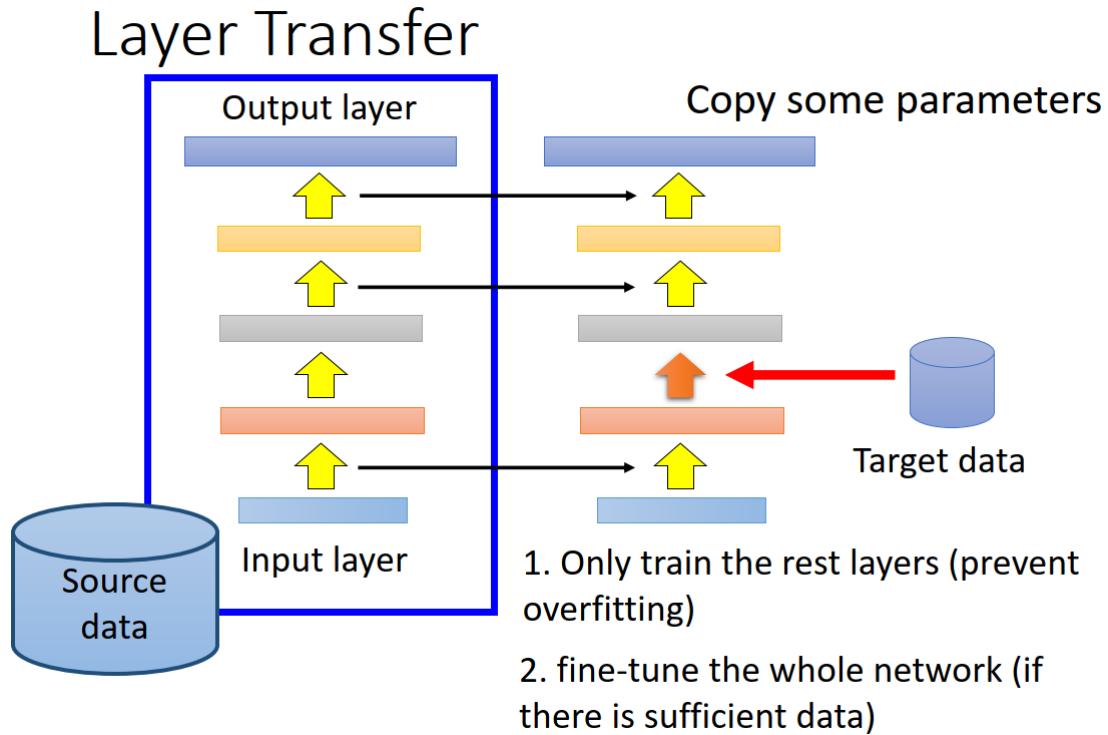
- 任务描述
目标数据量很少，源数据量很多。 (One-shot learning: 在目标域中只有几个或非常少的样例)
- 例子：（有监督）讲话者调整
目标数据：语音数据和某一特定讲话者的稿子。
源数据：语音数据和很多讲话者的稿子。
- 想法：用源数据训练一个模型，然后用目标数据微调模型
 - 难点：只有很有限的目标数据，所以要注意过拟合问题。
 - 一个解决过拟合难点的训练方法：Conservative Training (保留训练)

在微调新模型时加入限制 (regularization)，比如要求微调后的新模型与旧模型针对相同的输入的输出越相似越好，或者说模型的参数越相似越好。

Conservative Training



- 另一种方法：Layer Transfer（层转移器）
将用源数据训练好的模型的某几层取出/拷贝（连带参数），然后用目标数据去训练没有保留（拷贝出来）的层。

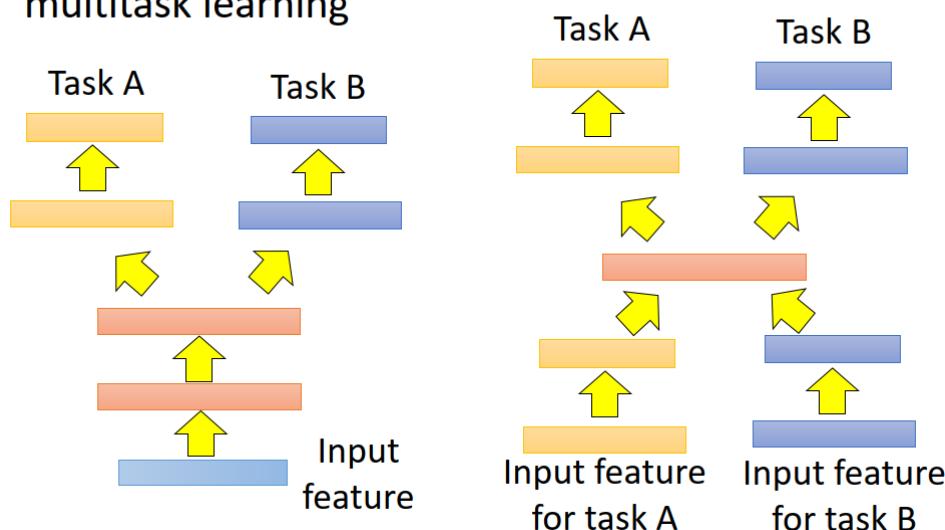


- 那么，这里就产生一个问题：究竟哪些层应该被转移（拷贝）呢？答案是：针对不同的学习任务，往往是需要不一样的层。
 - 在语音识别上，一般拷贝最后几层。（直觉：不同的人由于口腔结构差异，同样的发音方式得到的声音可能不同。而模型的前几层做的事是讲话者的发音方式，后面的几层再根据发音方式就可以获得被辨识的文字，即是跟具体讲话者没有太大关系的，所以做迁移时，只保留后面几层即可，而前面几层就利用新的特定讲话者的目标数据来做训练）
 - 在图像识别上，一般拷贝前几层。（直觉：网络的前几层一般学到的是最简单的模式（比如直线，横线或最简单的几何模型），比较通用，而后几层学到的模式已经很抽象了，很难迁移到底其他的领域）

Multitask Learning (labelled source, labelled target)

在Fine-tuning上，我们其实只关心迁移模型在目标数据域上的学习任务完成的好不好，而不关心在源数据域上表现如何。而Multitask Learning是会同时关注这两点的。

- The multi-layer structure makes NN suitable for multitask learning

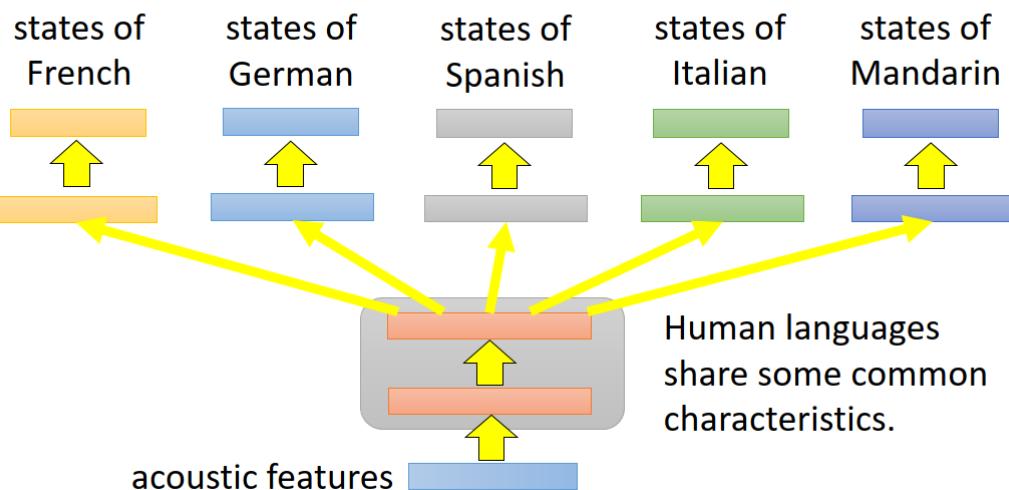


我们针对两种任务A和B，我们用它们的数据一起训练NN模型的前几层，再分别训练模型的后面几层包括输出层，以针对各自任务输出针对性的结果，这样的好处是由于训练数据量的增加，模型的性能可能会更好。关键是要确定两个任务有没有共通性，即是不是能共用前面几层。

另外，也可以在中间几层用共同数据来训练。

多任务学习比较成果的一个应用实例是：多语言识别。

Multitask Learning - Multilingual Speech Recognition

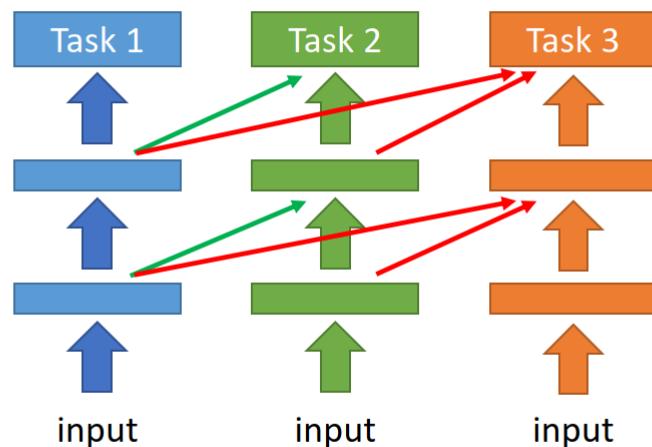


如上图，多国语言语音识别模型的前几层共享一些公共特征。

那么，就又产生了一个问题：语言迁移的范围可以有多广呢？

先针对Task1训练一个NN，在训练Task2的NN时，它的每一个隐层都会借用一个Task1中的NN的隐层（也可以直接设为全0的参数，相当于不借用），这样对Task1的模型性能不会有影响，也可以在Task2中对其已有参数进行借用。

Progressive Neural Networks



Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, Raia Hadsell, "Progressive Neural Networks", arXiv preprint 2016

Domain-adversarial training (labelled source, unlabelled target)

- 任务描述

源数据对应的学习任务和目标数据对应的学习任务是比较相似的，都是做数字识别，但是两者的输入数据差别很大（目标数据加入了背景），那如何让源数据上学出来的模型也能在目标数据上发挥良好呢？

Task description

- Source data: $(x^s, y^s) \rightarrow$ Training data
 - Target data: $(x^t) \longrightarrow$ Testing data
- } Same task, mismatch



而既然源数据的标签已知，目标数据的标签未知，那么我们可以将源数据当做训练数据，目标数据作测试数据来进行迁移。

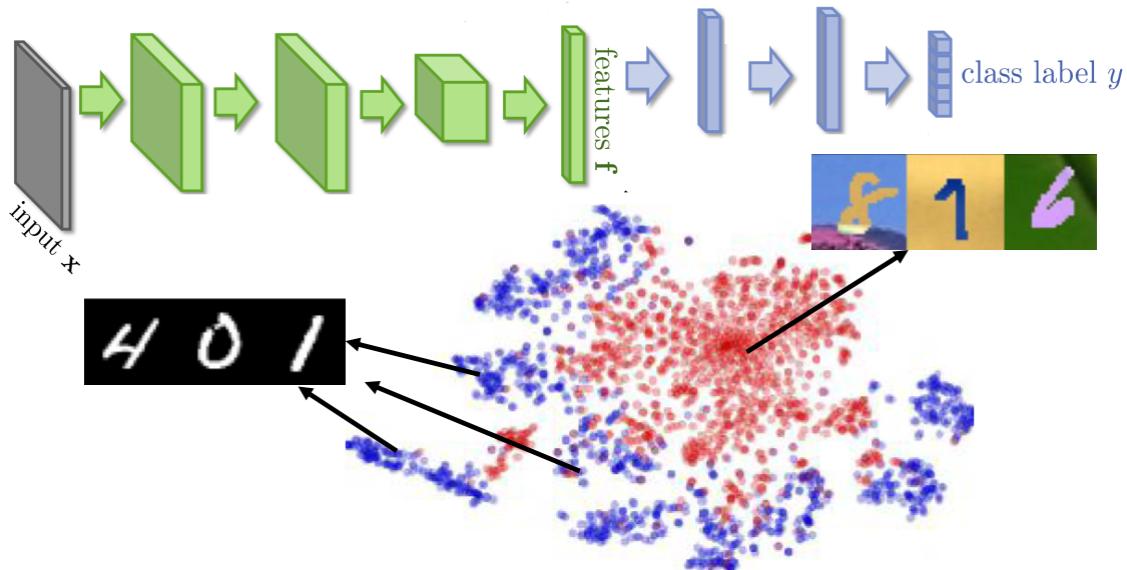
- Domain-adversarial training

而如果我们直接用MNIST的数据学一个model，去做MNIST-M的识别，效果是会非常差的。

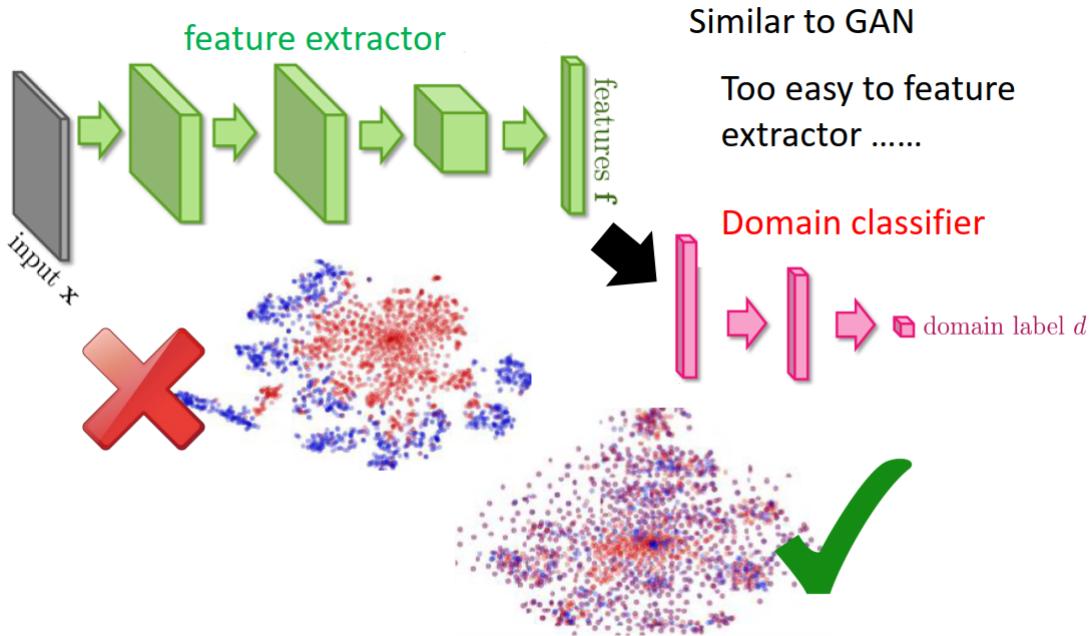
而我们知道，一个NN前几层做的事情相当于特征抽取 (feature extraction)，后几层做的事相当于分类 (Classification)，所以我们将前面几层的输出结果拿出来看一下会是什么样子。

如下图，MNIST的数据可以看做蓝色的点，MNIST-M的点可以看做红色的点，所以它们分别对应的特征根本不一样，所以做迁移时当然效果很差。

Domain-adversarial training



于是思考：能不能让NN的前几层将不同域的各自特性消除掉，即把不同的域特性消除掉。

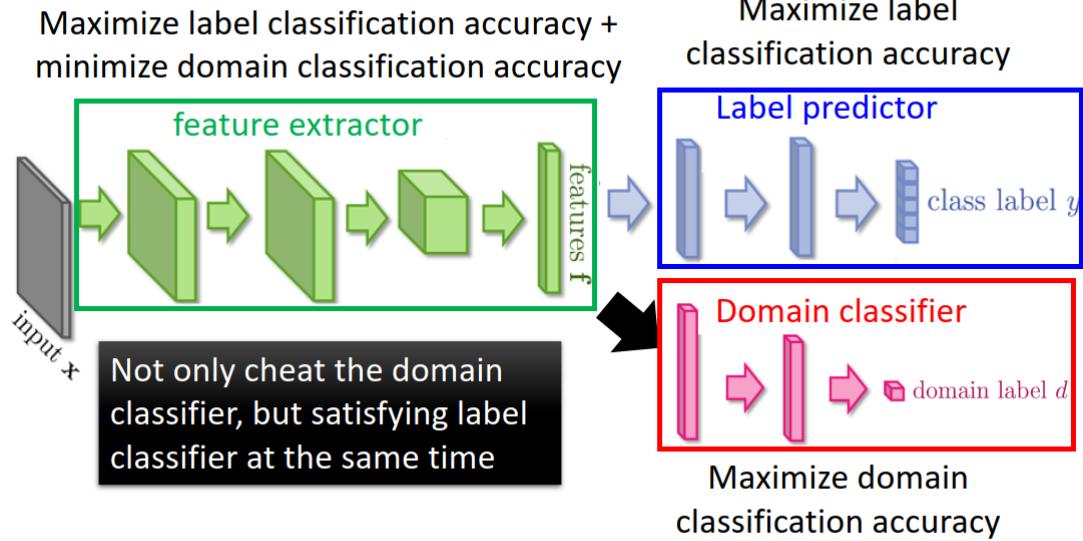


怎么训练这样一个NN呢？

训练一个域分类器，让它无法准确地对特征提取器的结果进行分类。（“骗过”分类器，类似GAN的思想）。但是，在GAN里面，生成器是要生成一张图片来骗过判别器，这个是很困难的，而这里如果只是要骗过域分类器就太简单了，直接让域分类器对任意输入都输出0就行了。这样学出来的特征提取器很可能是完全无效的。所以，我们应该给特征提取器增加学习难度。

所以，我们还要求特征提取器的输出能够满足标签预测器（label predictor）的高精度判别需求。

Domain-adversarial training



This is a big network, but different parts have different goals.

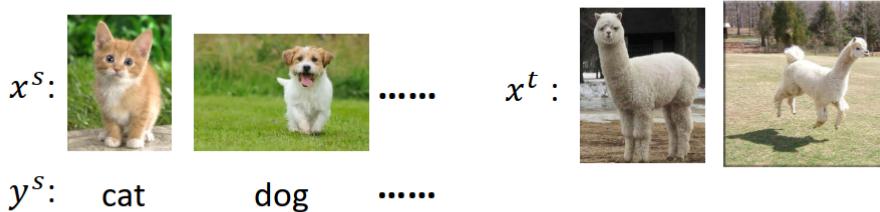
那么如何让特征提取器的训练满足我们上述两个要求呢？也不难，针对标签预测器的反向传播的误差，我们按照误差的方向进行正常的参数调整，而针对域分类器反向传回的误差，我们则按照误差的反方向进行误差调整（即域分类器要求调高某个参数值以提高准确度，而我们就故意调低对应参数值，以“欺负”它）。所以在域分类器的误差上加个负号就可以。

Zero-shot Learning (labelled source, unlabelled target)

在Zero-shot Learning里面，相对有比Domain-adversarial training更严苛的定义，它要求迁移的两种任务差别也是很大的。

如下图，源数据是猫狗图片，而目标数据是草泥马图片，这两种分类任务属于不同的任务了。

- Source data: $(x^s, y^s) \rightarrow$ Training data
 - Target data: $(x^t) \rightarrow$ Testing data
- } Different tasks



In speech recognition, we can not have all possible words in the source (training) data.

How we solve this problem in speech recognition?

- 影像识别上的做法

我们首先将分类任务中对应的所有分类目标的属性存在数据库里，并且必须保证每个分类目标的属性组合独一无二（属性组合重复的两者将无法在下述方法中区分）。

如下图，那么，我们在学习NN模型时，不再要求NN的输出直接是样例的分类，而要求是对应的分类目标包含哪些属性。那么，我们再在目标数据/测试数据上做分类时，也只要先用模型获提取出样例的属性，再查表即可。

而如果数据特征变得复杂（比如是图片作为输入），那么我们就可以做embedding，即尝试训练一个NN，将样例特征映射到一个低维的embedding空间，然后让映射后的结果和样例对应的属性尽可能地相近。

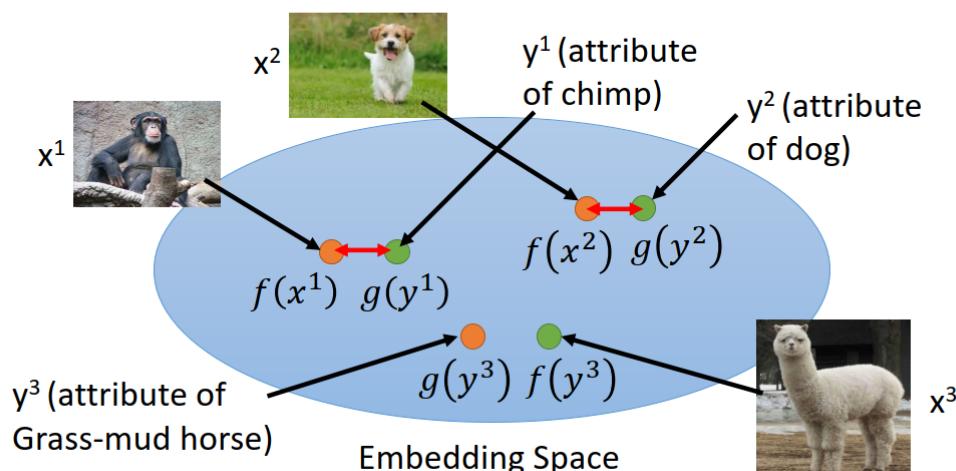
Zero-shot Learning

$f(*)$ and $g(*)$ can be NN.

Training target:

$f(x^n)$ and $g(y^n)$ as close as possible

- Attribute embedding



但是，如果我们没有数据库（没有属性数据）呢？

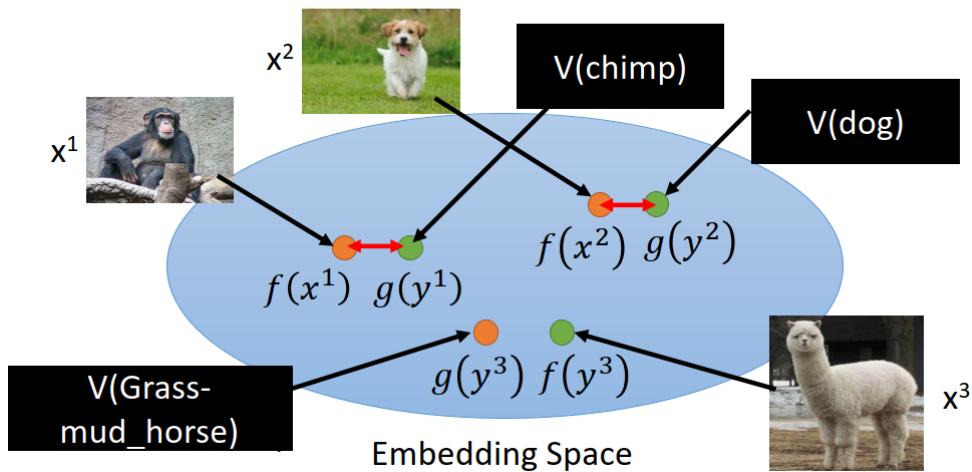
借用Word Vector的概念：word vector的每一个dimension代表了当前这个word的某一个属性，所以其实我们也不需要知道具体每个动物对应的属性是什么，只要知道每个动物对应的word

vector就可以了。即将属性换成word vector，再做embedding。

Zero-shot Learning

What if we don't have database

- Attribute embedding + word embedding



- Zero-shot Learning 的训练

思想：让同一对的 x 与 y 尽量靠近，让不同对的 x 与 y 尽量远离。

$$f^*, g^* = \arg \min_{f,g} \sum_n \|f(x^n) - g(y^n)\|_2 \quad \text{Problem?}$$

$$f^*, g^* = \arg \min_{f,g} \sum_n \max \left(0, k - f(x^n) \cdot g(y^n) + \max_{m \neq n} f(x^n) \cdot g(y^m) \right)$$

↑
Margin you defined

$$\text{Zero loss: } k - f(x^n) \cdot g(y^n) + \max_{m \neq n} f(x^n) \cdot g(y^m) < 0$$

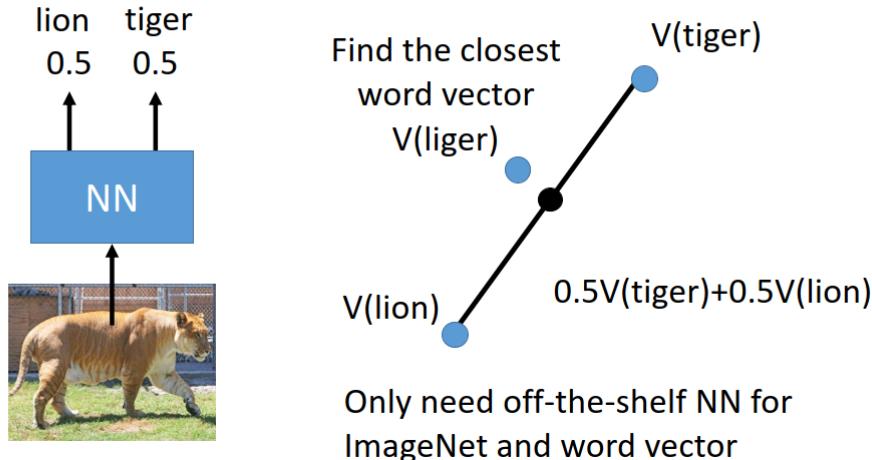
$$\frac{f(x^n) \cdot g(y^n)}{\text{f}(x^n) \text{ and } g(y^n) \text{ as close}} - \frac{\max_{m \neq n} f(x^n) \cdot g(y^m)}{\text{f}(x^n) \text{ and } g(y^m) \text{ not as close}} > k$$

还有一种训练方法更简单：Convex Combination of Semantic Embedding

假设训练出一个狮虎分类器，它对于一张图片给出了“50%是狮子，50%是老虎”的结果，那么我们就将狮子和老虎对应的word vector分别乘以0.5再加和，获得新的vector，看这个vector和哪个动物对应的vector最相近（比如狮虎兽liger最相近）。

而做这个只要求我们有一组word vector和一个语义辨识系统即可。

- Convex Combination of Semantic Embedding



Self-taught learning 和 Self-taught Clustering

这两种都是源数据无标签的。

self-taught learning可以看做一种半监督学习 (semi-supervised learning)，只是源数据和目标数据关系比较“疏远”。我们可以尝试利用源数据去提取出更好的representation (无监督方法)，即学习一个好的Feature extractor，再用这个extractor去帮助有标签的目标数据的学习任务。

Self-taught learning

- Learning to extract better representation from the source data (unsupervised approach)
- Extracting better representation for target data

Domain	Unlabeled data	Labeled data	Classes	Raw features
Image classification	10 images of outdoor scenes	Caltech101 image classification dataset	101	Intensities in 14x14 pixel patch
Handwritten character recognition	Handwritten digits ("0"–"9")	Handwritten English characters ("a"–"z")	26	Intensities in 28x28 pixel character/digit image
Font character recognition	Handwritten English characters ("a"–"z")	Font characters ("a"–"A" – "z"–"Z")	26	Intensities in 28x28 pixel character image
Song genre classification	Song snippets from 10 genres	Song snippets from 7 different genres	7	Log-frequency spectrogram over 50ms time windows
Webpage classification	100,000 news articles (Reuters newswire)	Categorized webpages (from DMOZ hierarchy)	2	Bag-of-words with 500 word vocabulary
UseNet article classification	100,000 news articles (Reuters newswire)	Categorized UseNet posts (from "SRAA" dataset)	2	Bag-of-words with 377 word vocabulary

前馈神经网络语言模型

最具代表性的神经语言模型是前馈神经网络语言模型 (Feed-forward Neural Network Language Model, FNNLM)。这种语言模型的目标是用神经网络计算 $P(w_m | w_{m-n+1} \dots w_{m-1})$ ，之后将多个 n-gram 的概率相乘得到整个序列的概率[72]。为了有一个直观的认识，这里以 4-gram 的 FNNLM 为例，即根据前三个单词 $w_{i-3}, w_{i-2}, w_{i-1}$ 预测当前单词 w_i 的概率。模型结构如图9.42所示。从结构上看，FNNLM 是一个典型的多层神经网络结构。主要有三层：

- 输入层（词的分布式表示层），即把输入的离散的单词变为分布式表示对应的实数向量；
- 隐藏层，即将得到的词的分布式表示进行线性和非线性变换；
- 输出层（Softmax 层），根据隐藏层的输出预测单词的概率分布。

这三层堆叠在一起构成了整个网络，而且也可以加入从词的分布式表示直接到输出层的连接（红色虚线）

箭头)。

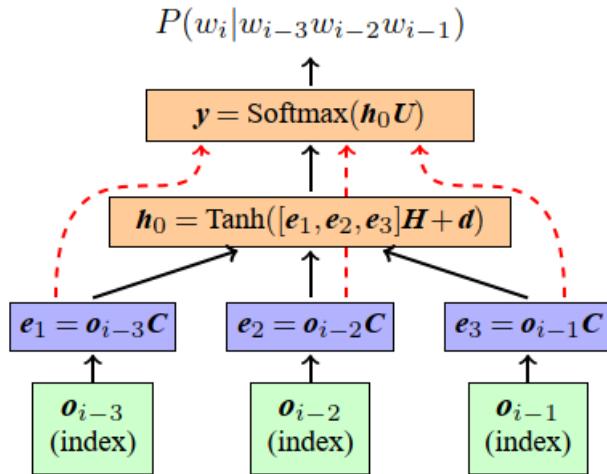


图 9.42 4-gram 前馈神经网络语言架构

编码器-解码器模型

编码器-解码器框架的创新之处在于，将传统基于符号的离散型知识转化为分布式的连续型知识。比如，对于一个句子，它可以由离散的符号所构成的文法规则来生成，也可以直接被表示为一个实数向量记录句子的各个“属性”。这种分布式的实数向量可以不依赖任何离散化的符号系统，简单来说，它就是一个函数，把输入的词串转化为实数向量。更为重要的是，这种分布式表示可以被自动学习。或者从某种意义上说，编码器-解码器框架的作用之一就是学习输入序列的表示。表示结果学习的好与坏很大程度上会影响神经机器翻译系统的性能。

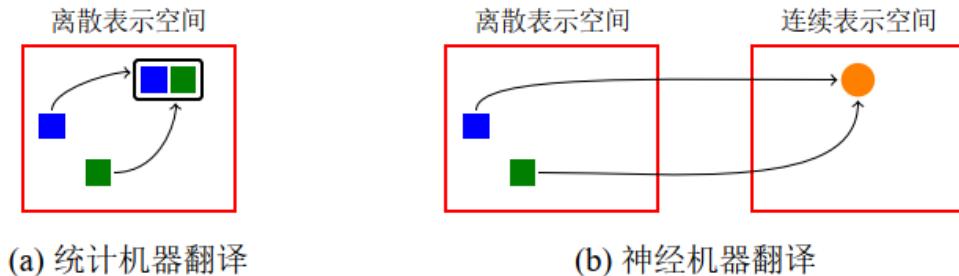


图 10.6 统计机器翻译和神经机器翻译的表示空间

实际上，编码器-解码器模型也并不是表示学习实现的唯一途径。比如，在第九章提到的神经语言模型实际上也是一种有效的学习句子表示的方法，它所衍生出的预训练模型可以从大规模单语数据上学习句子的表示形式。这种学习会比使用少量的双语数据进行编码器和解码器的学习更加充分。相比机器翻译任务，语言模型相当于一个编码器的学习 4，可以无缝嵌入到神经机器翻译模型中。不过，值得注意的是，机器翻译的目的是解决双语字符串之间的映射问题，因此它所使用的句子表示是为了更好地进行翻译。从这个角度说，机器翻译中的表示学习又和语言模型中的表示学习有不同。

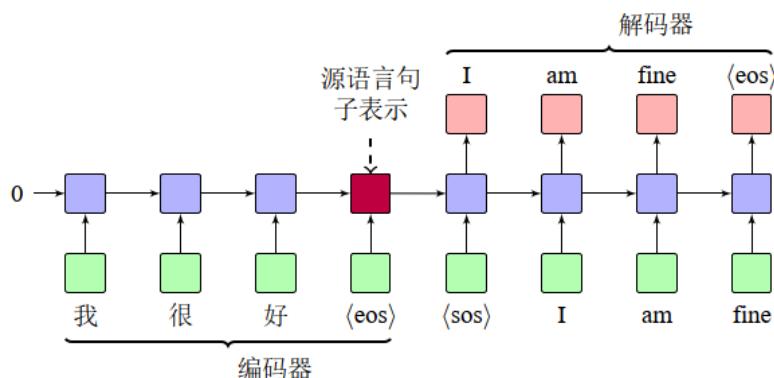


图 10.7 神经机器翻译的运行实例

翻译过程的神经网络结构如图10.7所示，其中左边是编码器，右边是解码器。

编码器会顺序处理源语言单词，将每个单词都表示成一个实数向量，也就是每个单词的词嵌入结果（绿色方框）。在词嵌入的基础上运行循环神经网络（蓝色方框）。在编码下一个时间步状态的时候，上一个时间步的隐藏状态会作为历史信息传入循环神经网络。这样，句子中每个位置的信息都被向后传递，最后一个时间步的隐藏状态（红色方框）就包含了整个源语言句子的信息，也就得到了编码器的编码结果——源语言句子的分布式表示。

解码器直接把源语言句子的分布式表示作为输入的隐层状态，之后像编码器一样依次读入目标语言单词，这是一个标准的循环神经网络的执行过程。与编码器不同的是，解码器会有一个输出层，用于根据当前时间步的隐层状态生成目标语言单词及其概率分布。可以看到，解码器当前时刻的输出单词与下一个时刻的输入单词是一样的。从这个角度说，解码器也是一种神经语言模型，只不过它会从另外一种语言（源语言）获得一些信息，而不是仅仅做单语句子的生成。具体来说，当生成第一个单词“I”时，解码器利用了源语言句子表示（红色方框）和目标语言的起始词“”。在生成第二个单词“am”时，解码器利用了上一个时间步的隐藏状态和已经生成的“I”的信息。这个过程会循环执行，直到生成完整的目标语言句子。从这个例子可以看出，神经机器翻译的流程其实并不复杂：首先通过编码器神经网络将源语言句子编码成实数向量，然后解码器神经网络利用这个向量逐词生成译文。现在几乎所有的神经机器翻译系统都采用类似的架构。

循环神经网络模型

RNN

虽然 RNN 的结构很简单，但是已经具有了对序列信息进行记忆的能力。实际上，基于 RNN 结构的神经语言模型已经能够取得比传统 n-gram 语言模型更优异的性能。在机器翻译中，RNN 也可以做为入门或者快速原型所使用的神经网络结构。后面会进一步介绍更加先进的循环单元结构，以及搭建循环神经网络中的常用技术。

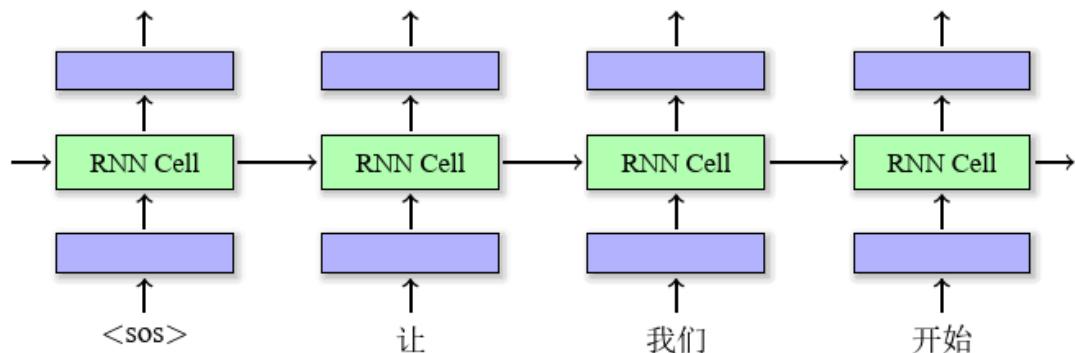


图 10.8 循环神经网络处理序列的实例

图10.8展示了一个循环神经网络处理序列问题的实例。当前时刻循环单元的输入由上一个时刻的输出和当前时刻的输入组成，因此也可以理解为，网络当前时刻计算得到的输出是由之前的序列共同决定的，即网络在不断地传递信息的过程中记忆了历史信息。以最后一个时刻的循环单元为例，它在对“开始”这个单词的信息进行处理时，参考了之前所有词（“让我们”）的信息。

在神经机器翻译里使用循环神经网络也很简单。只需要把源语言句子和目标语言句子分别看作两个序列，之后使用两个循环神经网络分别对其进行建模。

这个过程如图10.9所示。图中，下半部分是编码器，上半部分是解码器。编码器利用循环神经网络对源语言序列逐词进行编码处理，同时利用循环单元的记忆能力，不断累积序列信息，遇到终止符 后便得到了包含源语言句子全部信息的表示结果。解码器利用编码器的输出和起始符 开始逐词地进行解码，即逐词翻译，每得到一个译文单词，便将其作为当前时刻解码器端循环单元的输入，这也是一个典型的神经语言模型的序列生成过程。解码器通过循环神经网络不断地累积已经得到的译文的信息，并继续生成下一个单词，直到遇到结束符，便得到了最终完整的译文。

LSTM

RNN 结构使得当前时刻循环单元的状态包含了之前时间步的状态信息。但是这种对历史信息的记忆并不是无损的，随着序列变长，RNN 的记忆信息的损失越来越严重。在很多长序列处理任务中（如长文本生成）都观测到了类似现象。对于这个问题，研究人员提出了**长短时记忆**（Long Short-term Memory）模型，也就是常说的 LSTM 模型^[467]。

LSTM 模型是 RNN 模型的一种改进。相比 RNN 仅传递前一时刻的状态 h_{t-1} ，LSTM 会同时传递两部分信息：状态信息 h_{t-1} 和记忆信息 c_{t-1} 。这里， c_{t-1} 是新引入的变量，它也是循环单元的一部分，用于显性地记录需要记录的历史内容， h_{t-1} 和 c_{t-1} 在循环单元中会相互作用。LSTM 通过“门”单元来动态地选择遗忘多少以前的信息和记忆多少当前的信息。LSTM 中所使用的门单元结构如图10.11所示，包括遗忘门，输入门和输出门。图中 σ 代表 Sigmoid 函数，它将函数输入映射为 0-1 范围内的实数，用来充当门控信号。

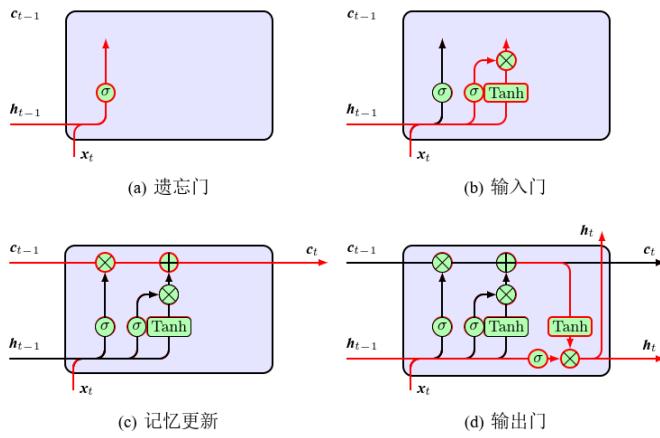


图 10.11 LSTM 中的门控结构

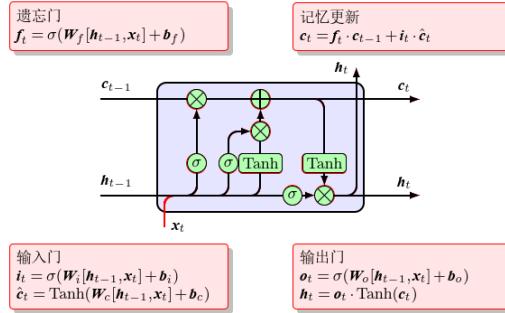


图 10.12 LSTM 的整体结构

GRU

LSTM 通过门控单元控制传递状态，忘记不重要的信息，记住必要的历史信息，在长序列上取得了很好的效果，但是其进行了许多门信号的计算，较为繁琐。**门循环单元**（Gated Recurrent Unit, GRU）作为一个 LSTM 的变种，继承了 LSTM 中利用门控单元控制信息传递的思想，并对 LSTM 进行了简化^[468]。它把循环单元状态 h_t 和记忆 c_t 合并成一个状态 h_t ，同时使用了更少的门控单元，大大提升了计算效率。

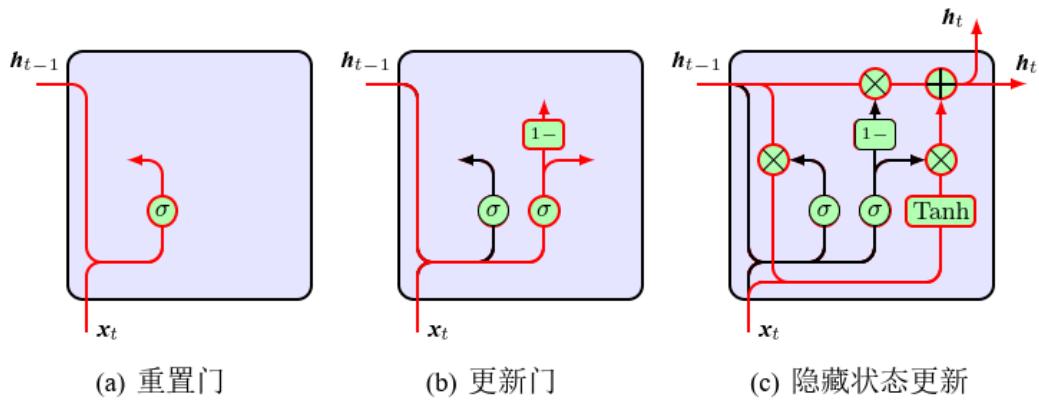


图 10.13 GRU 中的门控结构

双向模型

前面提到的循环神经网络都是自左向右运行的，也就是说在处理一个单词的时候只能访问它前面的序列信息。但是，只根据句子的前文来生成一个序列的表示是不全面的，因为从最后一个词来看，第一个词的信息可能已经很微弱了。为了同时考虑前文和后文的信息，一种解决办法是使用双向循环网络，其结构如图10.14所示。这里，编码器可以看作由两个循环神经网络构成，第一个网络，即红色虚线框里的网络，从句子的右边进行处理，第二个网络从句子左边开始处理，最终将正向和反向得到的结果都融合后传递给解码器。

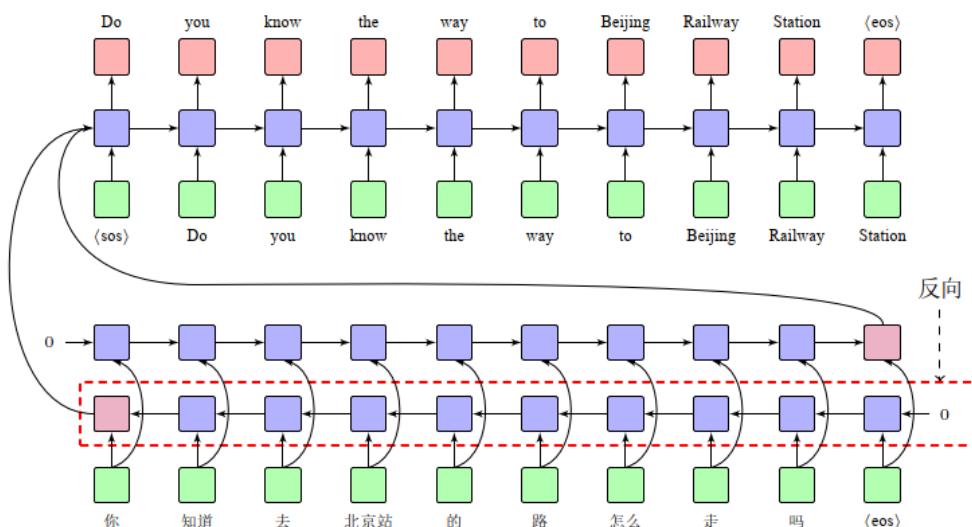


图 10.14 基于双向循环神经网络的机器翻译模型结构

双向模型是自然语言处理领域的常用模型，包括前几章提到的词对齐对称化、语言模型等中都大量地使用了类似的思路。实际上，这里也体现了建模时的非对称思想。也就是，建模时如果设计一个对称模型可能会导致问题复杂度增加，因此往往先对问题进行化简，从某一个角度解决问题。之后再融合多个模型，从不同角度得到相对合理的最终方案。

注意力机制

早期的神经机器翻译只使用循环神经网络最后一个单元的输出作为整个序列的表示，这种方式有两个明显的缺陷：

- 首先，虽然编码器把一个源语言句子的表示传递给解码器，但是一个维度固定的向量所能包含的信息是有限的，随着源语言序列的增长，将整个句子的信息编码到一个固定维度的向量中可能会造成源语言句子信息的丢失。显然，在翻译较长的句子时，解码器可能无法获取完整的源语言信息，降低翻译性能；
- 此外，当生成某一个目标语言单词时，并不是均匀地使用源语言句子中的单词信息。更普遍的情况是，系统会参考与这个目标语言单词相对应的源语言单词进行翻译。这有些类似于词对齐的作用，

即翻译是基于单词之间的某种对应关系。但是，使用单一的源语言表示根本无法区分源语言句子的不同部分，更不用说对源语言单词和目标语言单词之间的联系进行建模了。

神经机器翻译中的注意力机制并不复杂。对于每个目标语言单词 y_j ，系统生成一个源语言表示向量 \mathbf{C}_j 与之对应， \mathbf{C}_j 会包含生成 y_j 所需的源语言的信息，或者说 \mathbf{C}_j 是一种包含目标语言单词与源语言单词对应关系的源语言表示。相比用一个静态的表示 \mathbf{C} ，注意机制使用的是动态的表示 \mathbf{C}_j 。 \mathbf{C}_j 也被称作对于目标语言位置 j 的**上下文向量**（Context Vector）。图10.18对比了未引入注意力机制和引入了注意力机制的编码器-解码器结构。可以看出，在注意力模型中，对于每一个目标语言单词的生成，都会额外引入一个单独的上下文向量参与运算。

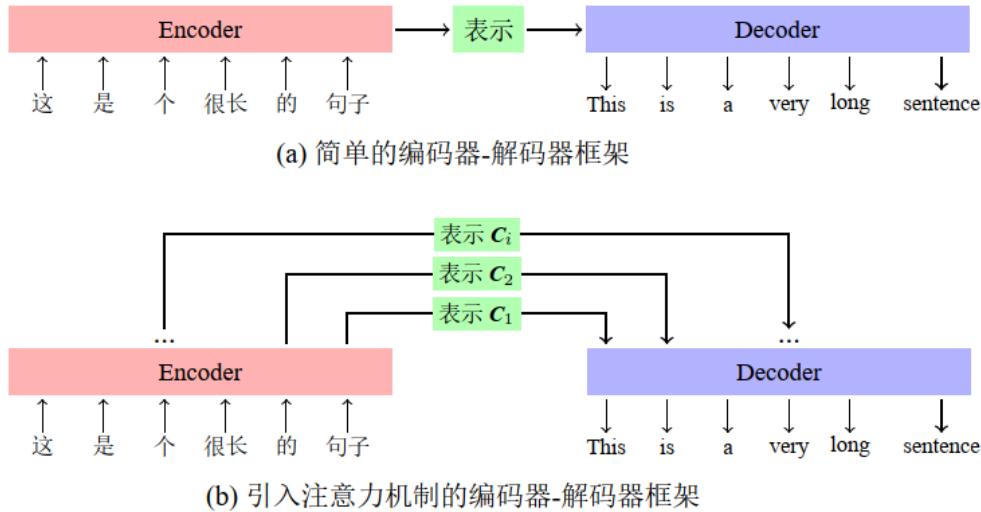


图 10.18 不使用 (a) 和使用 (b) 注意力机制的翻译模型对比

GNMT

GNMT 使用了编码器-解码器结构，构建了一个 8 层的深度网络，每层网络均由 LSTM 组成，且在编码器-解码器之间使用了多层次注意力连接。其结构如图10.24，编码器只有最下面 2 层为双向 LSTM。GNMT 在束搜索中也加入了长度惩罚和覆盖度因子来确保输出高质量的翻译结果。

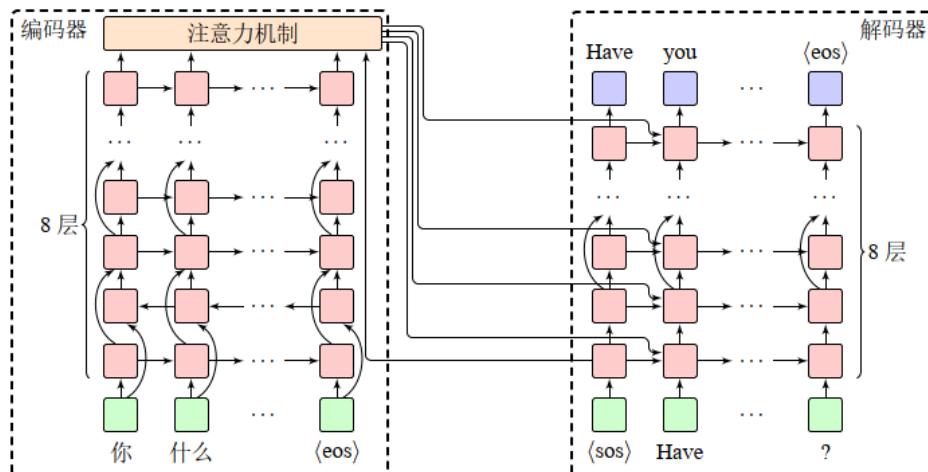


图 10.24 GNMT 结构

自注意力机制

首先回顾一下循环神经网络处理文字序列的过程。如图12.1所示，对于单词序列 $\{w_1, \dots, w_m\}$ ，处理第 m 个单词 w_m 时（绿色方框部分），需要输入前一时刻的信息（即处理单词 w_{m-1} ），而 w_{m-1} 又依赖于 w_{m-2} ，以此类推。也就是说，如果想建立 w_m 和 w_1 之间的关系，需要 $m-1$ 次信息传递。对于长序列来说，词汇之间信息传递距离过长会导致信息在传递过程中丢失，同时这种按顺序建模的方式也使得系统对序列的处理十分缓慢。



图 12.1 循环神经网络中单词之间的依赖关系

那么能否摆脱这种顺序传递信息的方式，直接对不同位置单词之间的关系进行建模，即将信息传递的距离拉近为1？自注意力机制的提出便有效解决了这个问题^[530]。图12.2给出了自注意力机制对序列进行建模的示例。对于单词 w_m ，自注意力机制直接建立它与前 $m-1$ 个单词之间的关系。也就是说， w_m 与序列中所有其他单词的距离都是1。这种方式很好地解决了长距离依赖问题，同时由于单词之间的联系都是相互独立的，因此也大大提高了模型的并行度。

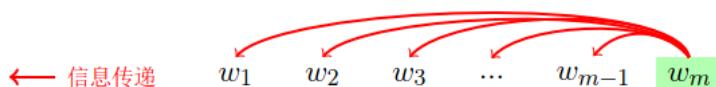


图 12.2 自注意力机制中单词之间的依赖关系

transformer

首先再来看看第十章介绍的循环神经网络，虽然它很强大，但是也存在一些弊端。其中比较突出的问题是，循环神经网络每个循环单元都有向前依赖性，也就是当前时间步的处理依赖前一时间步处理的结果。这个性质可以使序列的“历史”信息不断被传递，但是也造成模型运行效率的下降。特别是对于自然语言处理任务，序列往往较长，无论是传统的RNN结构，还是更为复杂的LSTM结构，都需要很多次循环单元的处理才能够捕捉到单词之间的长距离依赖。由于需要多个循环单元的处理，距离较远的两个单词之间的信息传递变得很复杂。

针对这些问题，研究人员提出了一种全新的模型——Transformer[23]。与循环神经网络等传统模型不同，Transformer模型仅仅使用自注意力机制和标准的前馈神经网络，完全不依赖任何循环单元或者卷积操作。自注意力机制的优点在于可以直接对序列中任意两个单元之间的关系进行建模，这使得长距离依赖等问题可以更好地被求解。此外，自注意力机制非常适合在GPU上进行并行化，因此模型训练的速度更快。表12.1对比了RNN、CNN和Transformer的层类型复杂度1。

注意，Transformer并不简单等同于自注意力机制。Transformer模型还包含了很多优秀的技术，比如：多头注意力、新的训练学习率调整策略等等。这些因素一起组成了真正的Transformer。下面就一起看一看自注意力机制和Transformer是如何工作的。

架构

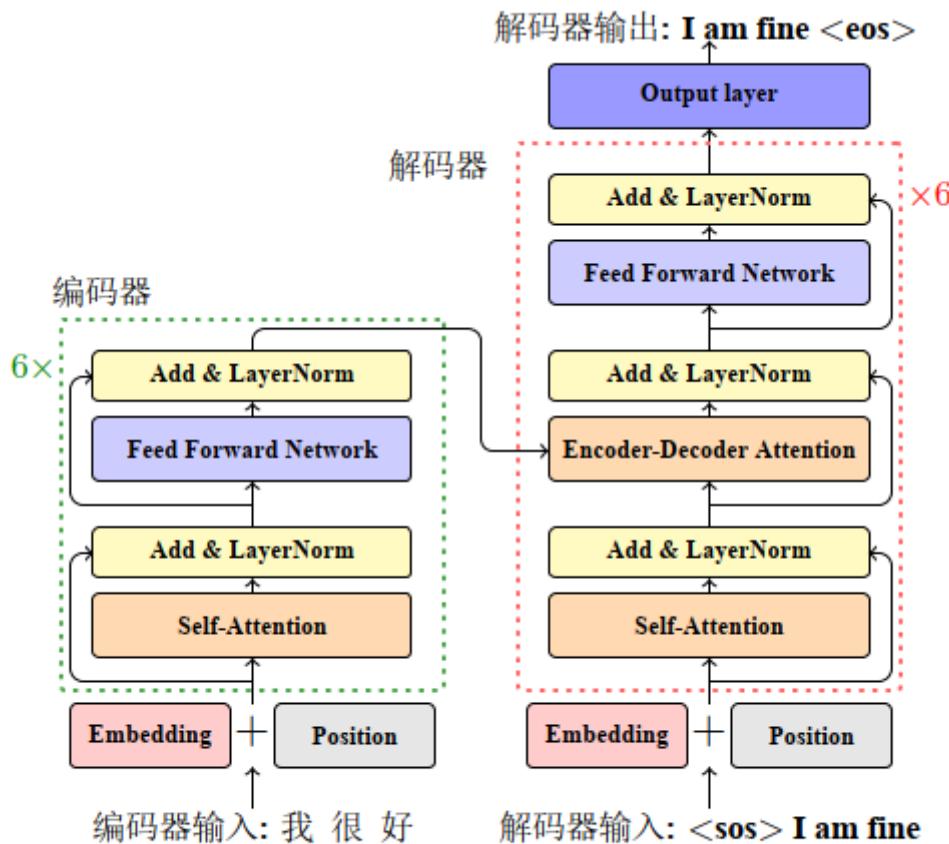


图 12.4 Transformer 结构

图12.4展示了Transformer的结构。编码器由若干层组成（绿色虚线框就代表一层）。每一层（Layer）的输入都是一个向量序列，输出是同样大小的向量序列，而Transformer层的作用是对输入进行进一步的抽象，得到新的表示结果。不过这里的层并不是指单一的神经网络结构，它里面由若干不同的模块组成，包括：

- 自注意力子层（Self-Attention Sub-layer）：使用自注意力机制对输入的序列进行新的表示；
- 前馈神经网络子层（Feed-Forward Sub-layer）：使用全连接的前馈神经网络对输入向量序列进行进一步变换；
- 残差连接（标记为“Add”）：对于自注意力子层和前馈神经网络子层，都有一个从输入直接到输出的额外连接，也就是一个跨子层的直连。残差连接可以使深层网络的信息传递更为有效；
- 层标准化（Layer Normalization）：自注意力子层和前馈神经网络子层进行最终输出之前，会对输出的向量进行层标准化，规范结果向量取值范围，这样易于后面进一步的处理。

编码器

以上操作就构成了Transformer的一层，各个模块执行的顺序可以简单描述为：Self-Attention → Residual Connection → Layer Normalization → Feed Forward Network → Residual Connection → Layer Normalization。编码器可以包含多个这样的层，比如，可以构建一个六层编码器，每层都执行上面的操作。最上层的结果作为整个编码的结果，会被传入解码器。

解码器

解码器的结构与编码器十分类似。它也是由若干层组成，每一层包含编码器中的所有结构，即：自注意力子层、前馈神经网络子层、残差连接和层标准化模块。此外，为了捕捉源语言的信息，解码器又引入了一个额外的编码-解码注意力子层（Encoder-Decoder Attention Sub-layer）。这个新的子层，可以帮助模型使用源语言句子的表示信息生成目标语言不同位置的表示。编码-解码注意力子层仍然基于自注意力机制，因此它和自注意力子层的结构是相同的，只是query、key、value的定义不同。比如，在解码器端，自注意力子层的query、key、value是相同的，它们都等于解码器每个位置的表示。而在编码-

解码注意力子层中，query 是解码器每个位置的表示，此时key 和 value 是相同的，等于编码器每个位置的表示。图12.5给出了这两种不同注意力子层输入的区别。

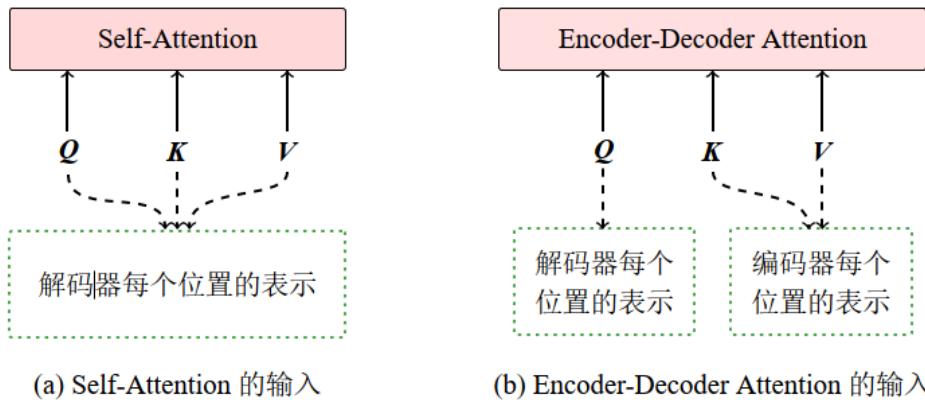


图 12.5 注意力模型的输入（自注意力子层 vs 编码-解码注意力子层）

此外，编码器和解码器都有输入的词序列。编码器的词序列输入是为了对其进行表示，进而解码器能从编码器访问到源语言句子的全部信息。解码器的词序列输入是为了进行目标语言的生成，本质上它和语言模型是一样的，在得到前 $n-1$ 个单词的情况下输出第 n 个单词。除了输入词序列的词嵌入，Transformer 中也引入了位置嵌入，以表示每个位置信息。原因是，自注意力机制没有显性地对位置进行表示，因此也无法考虑词序。在输入中引入位置信息可以让自注意力机制间接地感受到每个词的位置，进而保证对序列表示的合理性。最终，整个模型的输出由一个 Softmax 层完成，它和循环神经网络中的输出层是完全一样的。

Transformer改进

对 Transformer 等模型来说，处理超长序列是较为困难的。一种比较直接的解决办法是优化自注意力机制，降低模型计算复杂度。例如，采用了基于滑动窗口的局部注意力的 Longformer 模型[808]、基于随机特征的 Performer[727]、使用低秩分解的 Linformer[810] 和应用星型拓扑排序的 Star-Transformer[874]。

神经机器翻译结构优化

注意力机制的改进

神经网络连接优化及深层模型

基于句法的神经机器翻译模型

基于结构搜索的翻译模型优化

感想

如何构建一套好的机器翻译系统呢？假设我们需要为用户提供一套翻译品质不错的机器翻译系统，至少需要考虑三个方面：有足够大规模的双语句对集合用于训练、有强大的机器翻译技术和错误驱动的打磨过程。从技术应用和产业化的角度看，对于构建一套好的机器翻译系统来说，上述三个方面缺一不可。

- 从数据角度来看，针对资源稀缺语种的机器翻译技术研究也成了学术界的研究热点。在缺乏足够大规模的双语句对集合作为训练数据的情况下，研究人员也是巧妇难为无米之炊。从技术研究和应用可行性的角度看，解决资源稀缺语种的机器翻译问题非常有价值。
- 从机器翻译技术来看，可实用的机器翻译系统的构建，需要多技术互补融合。做研究可以搞单点突破，但它很难能应对实际问题和改善真实应用中的翻译品质。多技术互补融合有很多研究工作，但是从应用角度来说，构建可实用的机器翻译系统，还需要考虑技术落地可行性。比如大规模知识图谱构建的代价和语言分析技术的精度如何，预训练技术对富资源场景下机器翻译的价值等。

- 错误驱动，即根据用户对机器翻译译文的反馈与纠正，完善机器翻译模型的过程。如果能采用隐性反馈学习方法，在用户不知不觉中不断改善、优化机器翻译品质，就非常酷了，这也许会成为将来的一个研究热点。

情感分析

- ◆ 情感分析研究观点挖掘、倾向性分析等
- ◆ 什么是观点挖掘与倾向性分析？
- ◆ 为什么需要观点挖掘与倾向性分析？

相关定义

- **观点：**人们对事物的看法，具有明显的主观性，不同人对同一事物的看法存在差异 
- **倾向性：**观点中所包含的情感倾向性
- **观点挖掘与倾向性分析：**从海量数据中挖掘观点信息，并分析观点信息的倾向性
 - 非结构化→结构化

情感分析或观点挖掘(in Wikipedia) 是自然语言处理、计算语言学与文本挖掘中的一个研究领域。它的目标在于确定一个说话者或作者对于相关话题的情感、观点或态度。

情感分析发展七项关键技术

- 情感分类
 - 基于传统机器学习方法的情感分类
 - 基于深度学习方法的情感分类
 - 面向评价对象的情感分类
- 情感元素抽取
 - 情感词表示学习
 - 评价对象抽取
 - 评价搭配抽取
- 跨领域情感分析
 - 从源领域到目标领域进行模型的迁移
 - 目的
标注少量或不标注目标领域的语料，利用源领域的语料在目标领域达到较好的性能
 - 情感分析任务的特点
 - 不同领域的评价对象不尽相同
 - 不同领域的评价表达千差万别

- 不同领域中的同一情感表达的极性不同

■ 个性化情感分析

- 在情感分析中加入个性化的元素
- 情感分析的展示变得独特、另类、拥有自己特质的需要，独具一格
- 基于用户用词习惯的方法
 - 不同用户和群体情感倾向具有差异性
 - 由于用户群体立场存在差异，不同的用户群体往往对同一话题的情感倾向不同
 - 不同用户群体表达相同情感时，用词风格不尽相同
- 基于认知理论的方法
 - 用户画像
 - + 属性维度：自然欣喜
 - + 性格维度：大五人格
 - + 行为维度：用户偏好
 - 结合用户信息进行更深入的情感分析与展示
 - + 不同的用户群往往对同一话题的情感倾向不同
 - + 用户群可按性别、年龄、职业等进行区分
- 基于网络结构的方法
 - 传统的情感分类算法仅关注文本（句子、段落）特征
 - + 单条文本的情感可能存在歧义
 - 社交网络上用户之间的连接关系（关注、赞同、@等），这种连接关系表征了相同的情感倾向性
 - + 在用户级别进行情感分析

■ 隐式情感分析

- 社会媒体中文本情感表达方式复杂
 - 多数没有显式情感词
 - 多使用语言修辞表达或事实性陈述
- 事实型隐式情感分析
- 修辞型隐式情感分析

■ 情感原因发现

- 基于文本的情感原因发现
- 基于个体立场的情感原因发现
- 基于群体立场的情感原因发现
- 情感原因通常是由个体共同作用产生的

■ 情感生成

- 评论文本生成
- 情绪对话生成

典型方法

- 情感识别
 - 词级别
 - 任务：
 - 识别词语的情感倾向性，构建词典资源
 - 方法：
 - 基本思路：利用词之间的相似度进行扩展

- 基于词典的方法
 - 基于语料库的方法
- 句子级别
 - 任务：识别句子的情感倾向性
 - 关键问题：如何进行特征表示
 - 分类：
 - 基于语料库的方法
 - 基于词典的方法
 - 融合方法
- 文档级别
 - 任务：识别篇章整体观点倾向性
 - 绝大多数方法与句子级别方法类似
 - 特征+分类器
 - 关键问题
 - 多观点倾向性：一篇商品评论中可能包含对于商品多方面的观点，每个观点的倾向性也可能不同，如何识别篇章整体的观点倾向性
- 篇章级观点倾向性识别仍然可以看做是一个文本分类任务
- 如果仅仅是用词袋子模型，那么文档级别与句子级别在处理方法上没有区别
- 主要问题在多观点混合问题，篇章中局部观点与整体观点不一致
- 观点挖掘
 - 观点对象抽取：抽取观点评价的对象
 - 观点持有者抽取
 - 基本思路(Kim AAAI 2005)
 - 命名实体识别
 - 句法结构特征
Convolution Kernel
 - 分类或者序列标注
SVM, Naïve Bayes, CRFs
 - 需要指代消解
- 观点检索
 - 任务：
 - 从海量文本中根据查询找到观点信息
 - 根据主题相关度(topic relevance)与观点倾向性
 - 关键问题
 - 找到主题相关度得分与观点倾向性得分的折中

情感分析六大趋势

一、从粗粒度到细粒度

二、从单领域到跨领域

三、从文本到社交媒体

四、从显式情感到隐式情感

五、从情感分类到情感原因

六、从情感分析到情感生成

文本自动摘要

文本摘要的定义

◆ 定义：

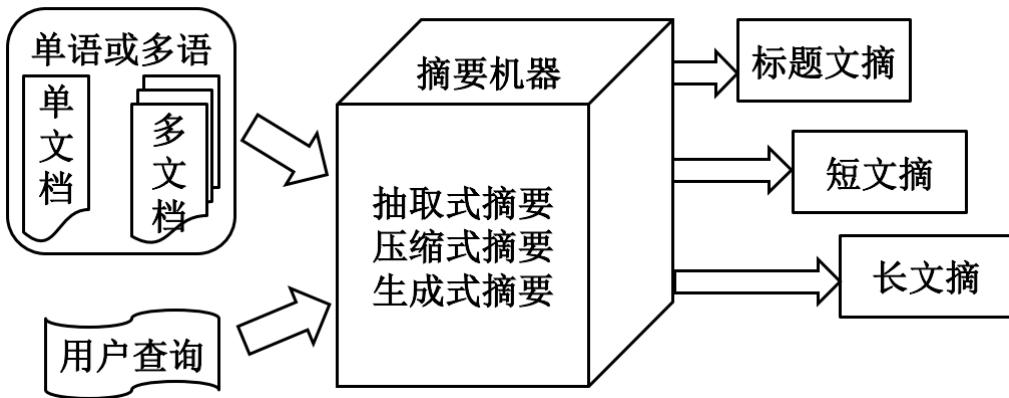
- ◆ 文本自动摘要是利用计算机按照某类应用自动地将文本（或文本集合）转换生成简短摘要的一种信息压缩技术

◆ 要求：

- ◆ 信息量足、覆盖面广、冗余度低和可读性高

文本摘要分类

- ①文档数目：单文档摘要、多文档摘要
- ②输入语言与输出语言的关系：单语摘要、跨语言摘要、多语言摘要
- ③是否有用户输入：通用摘要、用户查询摘要
- ④摘要方法：抽取式摘要、压缩式摘要、理解式摘要
- ⑤摘要长度：标题式摘要、短摘要、长摘要



文本摘要方法

- ◆ **抽取式摘要 (Extractive Summarization)**
 - ◆ 直接从原文中抽取已有的句子组成摘要
 - ◆ 简单易实现，但不符合摘要本质
 - ◆ 众多实际系统中，抽取式方法占主导
- ◆ **压缩式摘要**
 - ◆ 抽取并简化原文中的重要句子构成文摘
 - ◆ **ABACDCDFDSGFGDA → ABADFDSDA**
- ◆ **生成摘要 (Abstractive Summarization)**
 - ◆ 改写或重新组织原文内容形成最终文摘

抽取式摘要

- 三个重要模块
 - 句子重要性评估
 - 启发式规则：句子位置（越靠段首越重要）、词频、与标题相似度以及线索词（总之、总而言之）等
 - 机器学习方法：句子分类、最优化方法
 - 图模型方法：TextRank（PageRank的无向图模型）、HITS算法
 - 信息冗余句子去重复
 - 必要性
 - 多文档摘要中，不同文档通常包含非常相似的句子
 - 为了得到精简的摘要，需要消除冗余的句子
 - 主要方法
 - CSIS
 - MMR



MMR算法

1. 初始化两个集合 $A = \emptyset$ 和 $B = \{s_i | i = 1, \dots, n\}$, 分别表示摘要句子集合与未选句子集合; 初始化每个句子重要性和冗余度的综合得分 (开始时冗余度得分未知, 综合得分仅包含句子重要性的得分), $RS(s_i) = Score(s_i)$, $i = 1, \dots, n$.

2. 根据 $RS(s_i)$ 对集合 B 按照得分从高到底进行排序;

3. 假设 s_i 是得分最高的句子, 即 B 中的第一个句子, 将 s_i 从 B 中移除, 并加入 A 中, 然后按照下面的公式更新 B 中剩余每个句子的综合得分:

$$RS(s_j) = RS(s_j) - \lambda Sim(s_i, s_j) \cdot Score(s_j)$$

4. 返回第二步进行迭代直至集合 B 为空, 或者句子集合 A 达到长度要求。

- 根据长度、字数等约束生成最终摘要

压缩式摘要

- 核心模块: 句子压缩

1. 可视为树结构的精简问题
2. 可视为01序列标注任务

理解式摘要

改写或重新组织原文内容形成文摘

基于AMR的方法 AMR:Abstractive Meaning Representation

基于谓词论元结构的理解式摘要

- 核心思想: 选择并重组概念与行为
- 选择: 基于图的重要性打分+基于约束的整数线性规划

文本摘要评价

- 自动评价

- 给定人工参考摘要, 评价自动摘要结果的质量, 综合考虑内容的忠实度与行文的流畅度
- 省时省力、一致性高、加速方法迭代更新
- ROUGE: 基于N-元组计算自动摘要与人工摘要的匹配率

$$ROUGE - N(sum) = \frac{\sum_{r \in R} \sum_{n-gram \in r} count_{match}(n-gram, sum)}{\sum_{r \in R} \sum_{n-gram \in r} count(n-gram)}$$

系统摘要、参
考摘要匹配的
ngram个数

R -{Reference Summaries}表示参考摘要

参考摘要中的
ngram个数

- BE: 基于语义单元的ROUGE, 语义单元由句法分析得到
- 人工评价
 - 人工评价自动摘要结果的质量
 - 可靠性高、主观性强
 - 内容的忠实度: 金字塔方法
 - 行文的流畅度 (可读性) : 1-5