12. 自然语言处理I

情感分析、机器翻译

WU Xiaokun 吴晓堃

xkun.wu [at] gmail

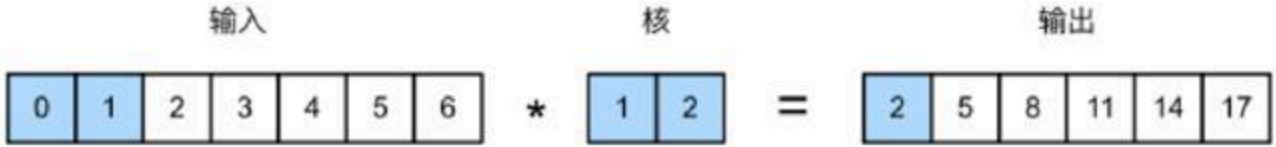
2022/05/09

情感分析

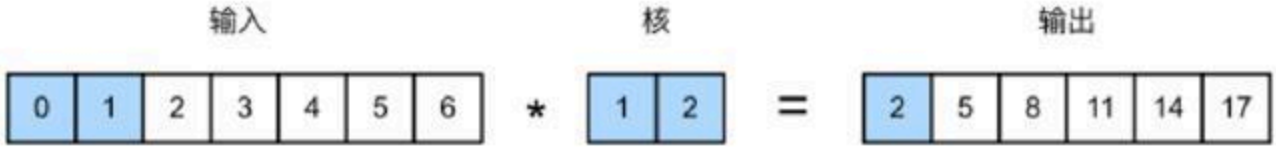
实验：情感分析数据集

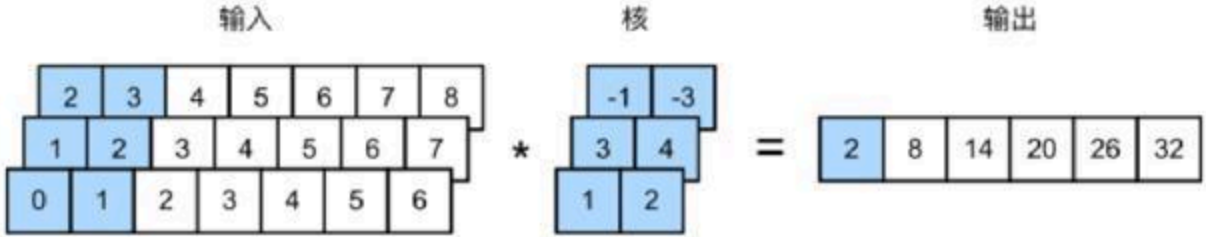
实验：情感分析RNN实现

一维卷积

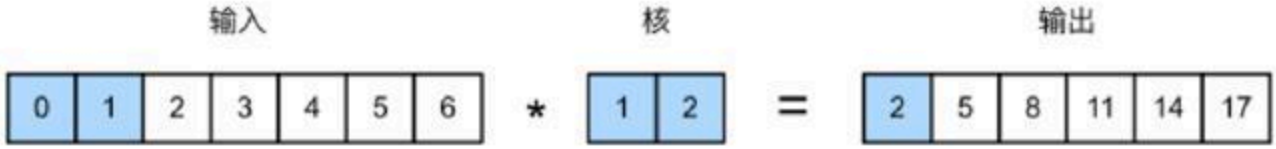


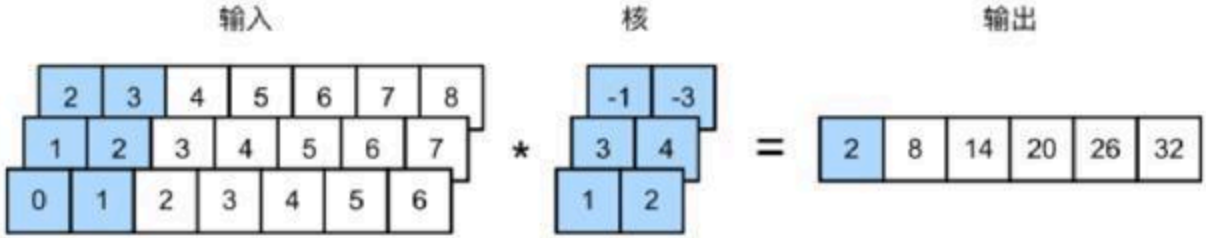
一维卷积

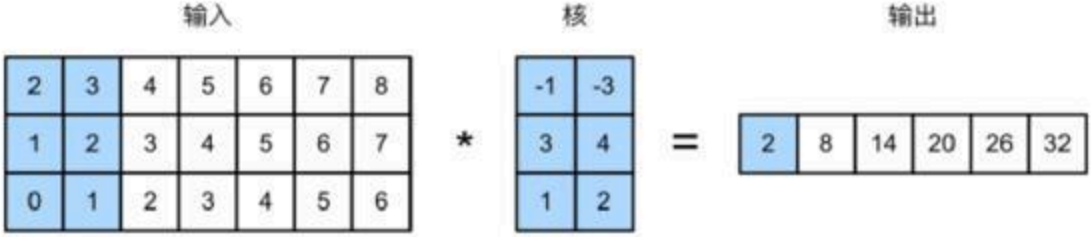

输入
核
输出
0 1 2 3 4 5 6
*
1 2
=
2 5 8 11 14 17


输入
核
输出
2 3 4 5 6 7 8
1 2 3 4 5 6 7
0 1 2 3 4 5 6
*
-1 -3
3 4
1 2
=
2 8 14 20 26 32

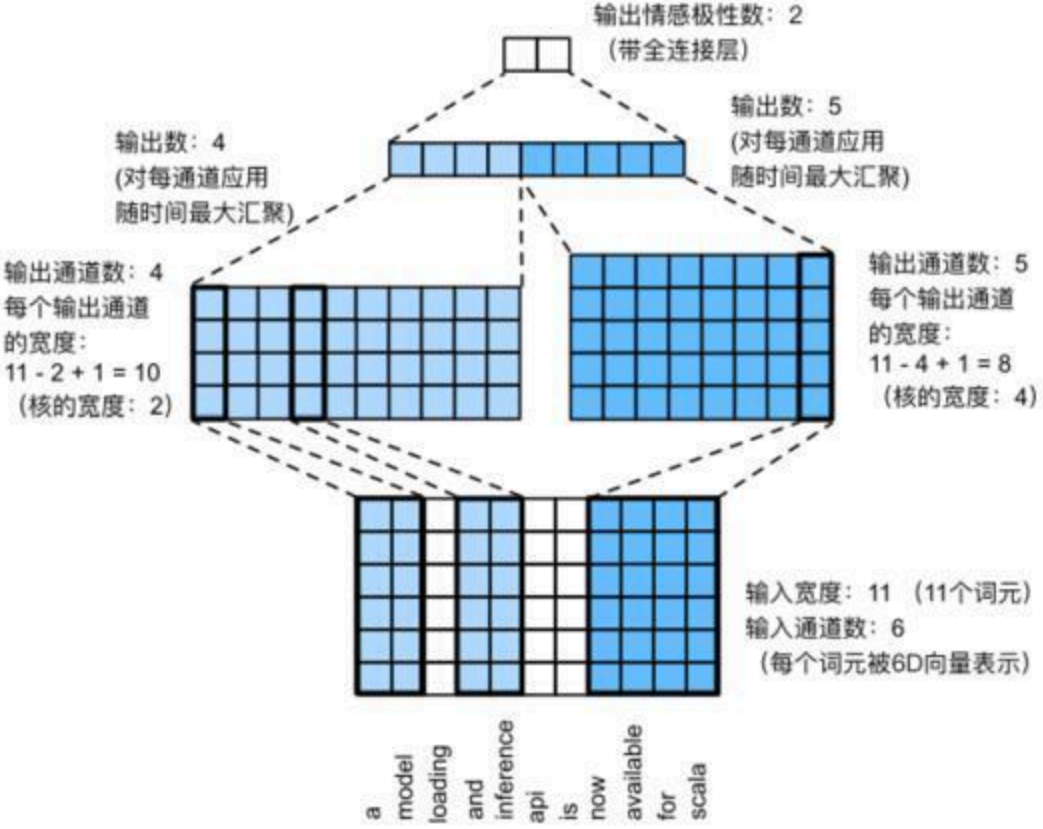
一维卷积


输入
核
输出
0 1 2 3 4 5 6
*
1 2
=
2 5 8 11 14 17


输入
核
输出
2 3 4 5 6 7 8
1 2 3 4 5 6 7
0 1 2 3 4 5 6
*
-1 -3
3 4
1 2
=
2 8 14 20 26 32


输入
核
输出
2 3 4 5 6 7 8
1 2 3 4 5 6 7
0 1 2 3 4 5 6
*
-1 -3
3 4
1 2
=
2 8 14 20 26 32

textCNN模型


输出情感极性数：2
（带全连接层）
输出数：5
(对每通道应用
随时间最大汇聚)
输出数：4
(对每通道应用
随时间最大汇聚)
输出通道数：4
每个输出通道
的宽度：
11 - 2 + 1 = 10
（核的宽度：2）
输出通道数：5
每个输出通道
的宽度：
11 - 4 + 1 = 8
（核的宽度：4）
输入宽度：11 （11个词元）
输入通道数：6
（每个词元被6D向量表示）
a
model
loading
and
inference
api
is
now
available
for
scala

实验：情感分析CNN实现

机器翻译与数据集

机器翻译

机器翻译：**序列转换**的核心问题

- 统计方法：例如字符/词频率对照表，早期曾用于破译密码、翻译古文字

机器翻译

机器翻译：**序列转换**的核心问题

- 统计方法：例如字符/词频率对照表，早期曾用于破译密码、翻译古文字
- 神经网络：**端到端**的学习，提取序列转换规则
  - 训练数据：两种语言的文本序列对

实验：机器翻译数据集

小结：机器翻译与数据集

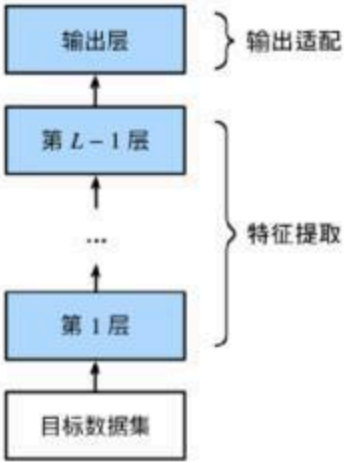
- 机器翻译：语言文本序列的自动转换
- 单词级词元化：词表大、稀疏，故将低频词元视为相同的未知词元
- 小批量加载数据：截断、填充文本序列成相同长度

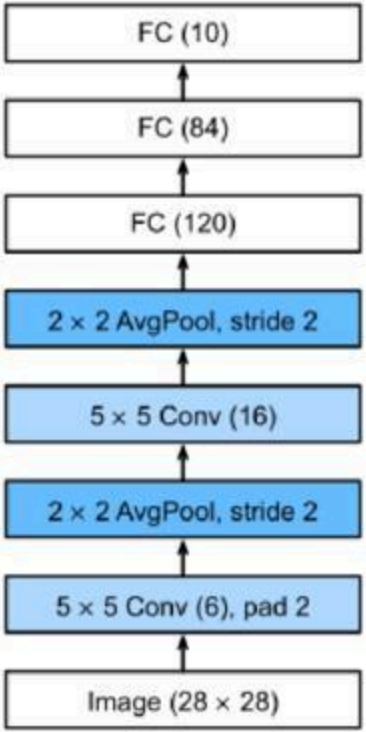
编码器-解码器架构

回顾：CNN

卷积神经网络可以划分成两个组件

1. **特征提取**：看成自动化特征工程
2. **输出适配**：例如分类器输出概率



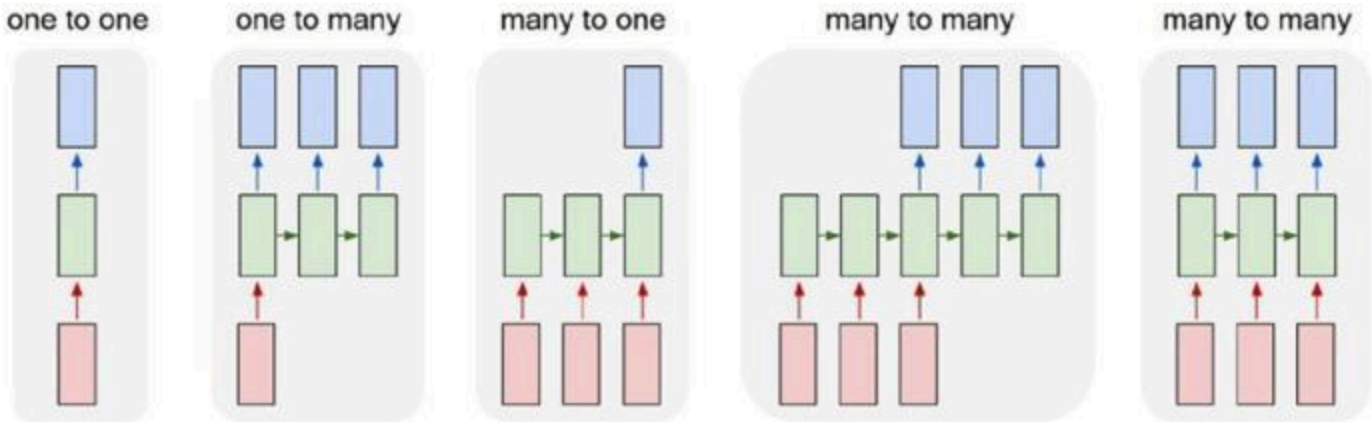


也可以认为是两个**处理阶段**

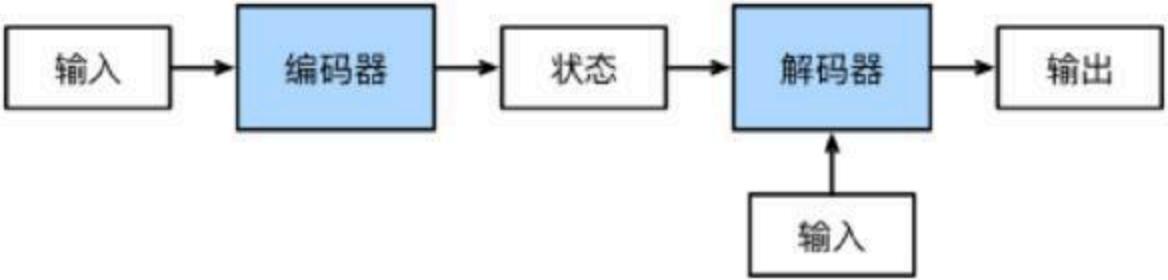
回顾：RNN

循环神经网络也可以划分成两个组件

- **编码器**：输入 -> 定长向量表示，即词嵌入
- **解码器**：定长向量表示 -> 输出



编码器-解码器：架构



- 解码器可能会用到输入

实验：编码器-解码器架构

小结：编码器-解码器架构

- 编码器-解码器：隐藏层提取词的向量表示（词嵌入）
  - 编码器：将可变序列转换成固定形状的编码状态
  - 解码器：将固定形状的编码状态映射为长度可变的输出序列
- 编码器-解码器架构：可以解决不定长输入、输出序列问题
  - 故适用于机器翻译等序列转换问题

序列到序列学习（seq2seq）

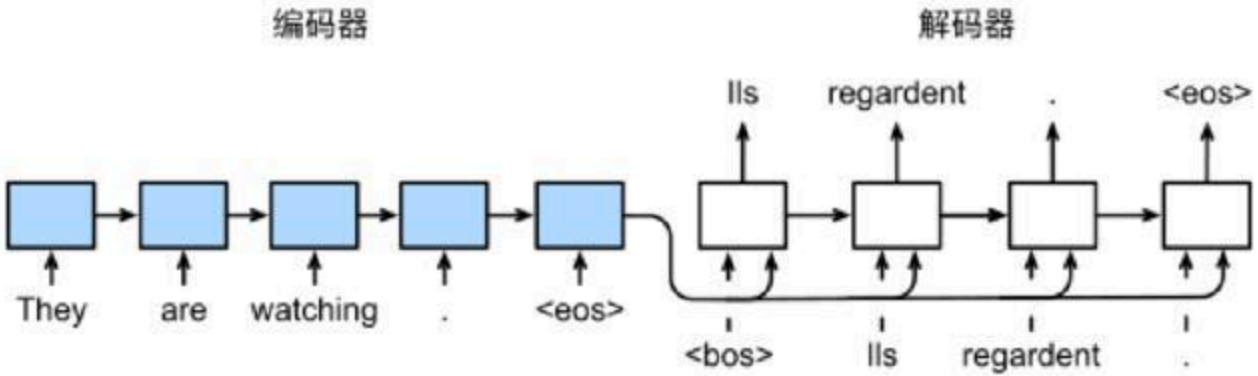
机器翻译：seq2seq

机器翻译：序列转换，**seq2seq**

- 输入、输出序列的**长度可变**

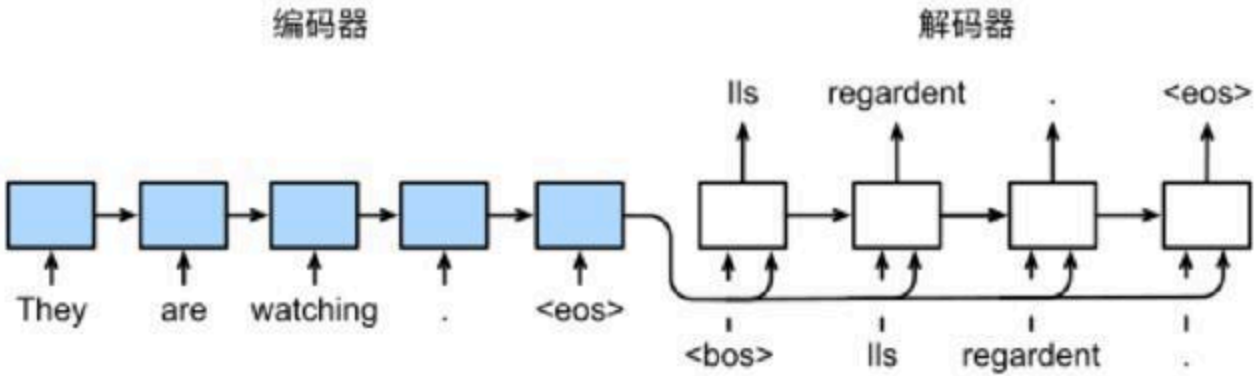
seq2seq：编码器-解码器

编码器-解码器架构：编码成**固定形状**的隐状态



seq2seq：编码器-解码器

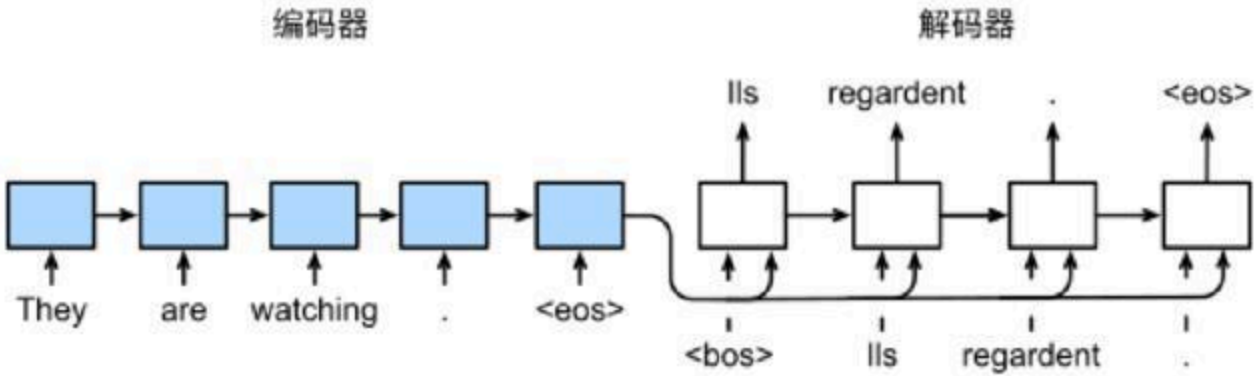
编码器-解码器架构：编码成**固定形状**的隐状态



- 编码器：RNN，输入序列 -> 隐状态
  - 输入序列长度可变？添加<pad>**填充字符**补齐成等长序列
  - 编码器可以是双向RNN

seq2seq：编码器-解码器

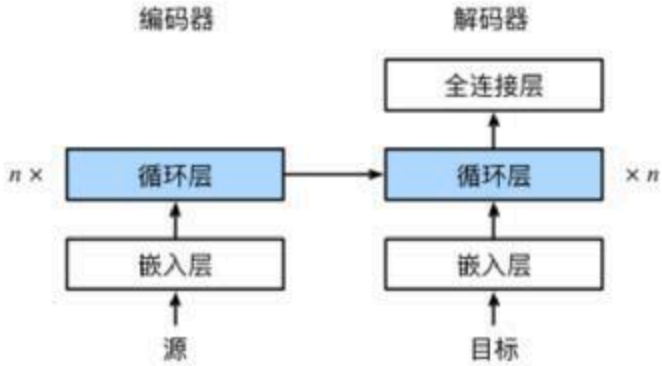
编码器-解码器架构：编码成**固定形状**的隐状态



- 编码器：RNN，输入序列 -> 隐状态
  - 输入序列长度可变？添加<pad>**填充字符**补齐成等长序列
  - 编码器可以是双向RNN
- 解码器：RNN，隐状态 + 输入序列 -> 输出序列
  - 何时终止预测？直到生成<eos>**终止字符**

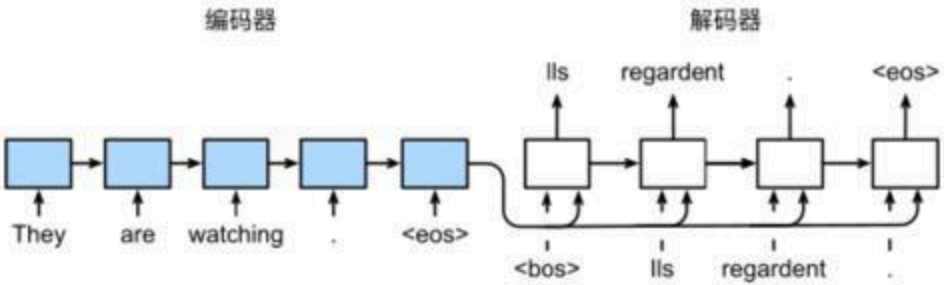
编码器-解码器细节

- 编码器：不输出、只**学习词嵌入**
  - 最后时间步隐状态：初始化解码器
- 解码器：根据上一输出**循环预测**



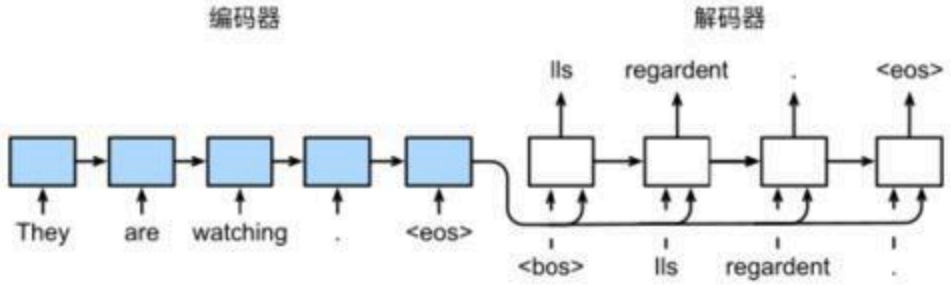
编码器-解码器：训练、预测

训练称为“**强制教学**”：原始输出序列（词元标签）被送入解码器

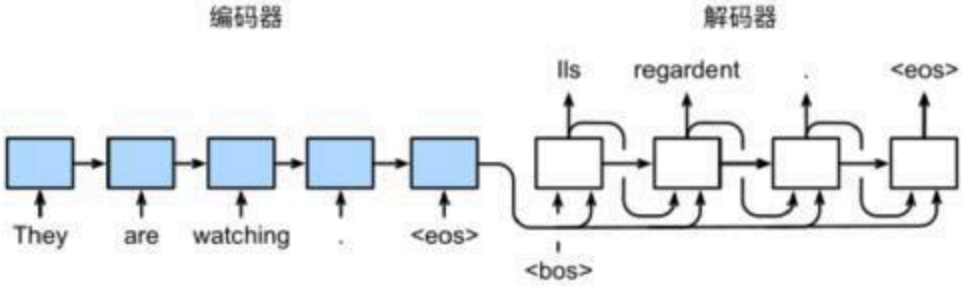


编码器-解码器：训练、预测

训练称为“**强制教学**”：原始输出序列（词元标签）被送入解码器



预测时没有真实标签：**前一时间步的预测**作为当前时间步的输入



机器翻译：评测

[Papineni 2002] **BLEU**（bilingual evaluation understudy）

$$\exp\left(\min\left(0, 1 - \frac{|Y|}{|\hat{Y}|}\right)\right) \prod_n p_n^{\frac{1}{2^n}}$$

- $|Y|$：标注词元数；$|\hat{Y}|$：预测词元数；取值：$[0, 1]$

机器翻译：评测

[Papineni 2002] **BLEU**（bilingual evaluation understudy）

$$\exp\left(\min\left(0, 1 - \frac{|Y|}{|\hat{Y}|}\right)\right) \prod_n p_n^{\frac{1}{2^n}}$$

- $|Y|$：标注词元数；$|\hat{Y}|$：预测词元数；取值：$[0, 1]$
- **惩罚过短的预测**：$\min(\cdot)$部分≤ 0
  - $|Y| \leq |\hat{Y}|$：$\exp(\cdot) = 1$，即无惩罚；否则预测$|\hat{Y}|$越短，惩罚越高

机器翻译：评测

[Papineni 2002] **BLEU**（bilingual evaluation understudy）

$$\exp\left(\min\left(0, 1-\frac{|Y|}{|\hat{Y}|}\right)\right)\prod_n p_n^{\frac{1}{2^n}}$$

- $|Y|$：标注词元数；$|\hat{Y}|$：预测词元数；取值：$[0,1]$
- **惩罚过短的预测**：$\min(\cdot)$部分≤ 0
  - $|Y| \leq |\hat{Y}|$：$\exp(\cdot)=1$，即无惩罚；否则预测$|\hat{Y}|$越短，惩罚越高
- p_n：**预测序列中所有**n-gram的**匹配度**，即匹配数量占总数的比例
  - 例如标注序列“ABCDEF”、预测序列“ABBCD”
    - $p_1=4/5, p_2=3/4, p_3=1/3, p_4=p_5=0$

机器翻译：评测

[Papineni 2002] **BLEU**（bilingual evaluation understudy）

$$\exp\left(\min\left(0, 1-\frac{|Y|}{|\hat{Y}|}\right)\right)\prod_n p_n^{\frac{1}{2^n}}$$

- $|Y|$：标注词元数；$|\hat{Y}|$：预测词元数；取值：$[0,1]$
- **惩罚过短的预测**：$\min(\cdot)$部分≤ 0
  - $|Y| \leq |\hat{Y}|$：$\exp(\cdot)=1$，即无惩罚；否则预测$|\hat{Y}|$越短，惩罚越高
- p_n：**预测序列中所有**n-gram的**匹配度**，即匹配数量占总数的比例
  - 例如标注序列“ABCDEF”、预测序列“ABBCD”
    - $p_1=4/5, p_2=3/4, p_3=1/3, p_4=p_5=0$
- **鼓励长匹配**：N^V种可能性，长匹配难度更大，故分配更高权重
  - $p_n^{\frac{1}{2^n}}$：多项式反函数；固定$0 \leq p_n \leq 1$时，n越大，函数值也越大

机器翻译：评测

[Papineni 2002] **BLEU**（bilingual evaluation understudy）

$$\exp\left(\min\left(0, 1-\frac{|Y|}{|\hat{Y}|}\right)\right)\prod_n p_n^{\frac{1}{2^n}}$$

- $|Y|$：标注词元数；$|\hat{Y}|$：预测词元数；取值：$[0,1]$
- **惩罚过短的预测**：$\min(\cdot)$部分≤ 0
  - $|Y| \leq |\hat{Y}|$：$\exp(\cdot)=1$，即无惩罚；否则预测$|\hat{Y}|$越短，惩罚越高
- p_n：**预测序列中所有**n-gram的**匹配度**，即匹配数量占总数的比例
  - 例如标注序列“ABCDEF”、预测序列“ABBCD”
    - $p_1=4/5, p_2=3/4, p_3=1/3, p_4=p_5=0$
- **鼓励长匹配**：N^V种可能性，长匹配难度更大，故分配更高权重
  - $p_n^{\frac{1}{2^n}}$：多项式反函数；固定$0 \leq p_n \leq 1$时，n越大，函数值也越大

- **惩罚短预测**：预测序列越短，p_n值越高，惩罚也越高

实验：seq2seq

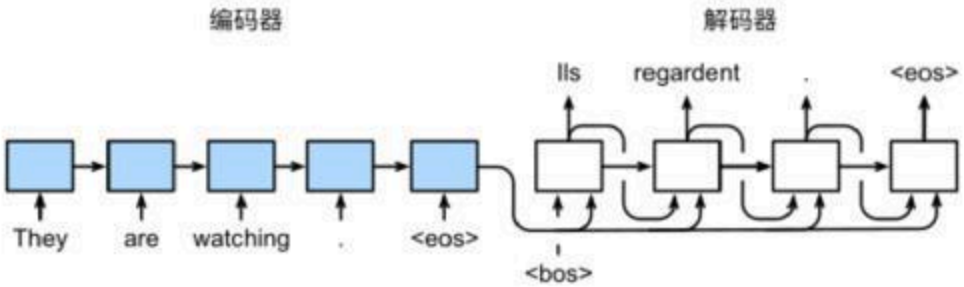
小结：seq2seq

- seq2seq：序列到序列（的转换），用于机器翻译
- 编码器-解码器架构：两个多层RNN
- 解决输入不定长序列：填充，并在计算损失函数时使用掩码过滤
- 训练：强制教学；预测：输入上一时间步预测
- 评价：BLEU，度量n-gram的匹配度，并惩罚短预测

束搜索

回顾：机器翻译

单步、逐词预测：前一时间步的预测作为当前时间步的输入



- 解码输出：词汇表上的概率分布
  - **贪心搜索**：选取概率值最大的词作为当前预测，见下页

贪心搜索

贪心搜索：选取概率值最大的词作为当前预测

- 贪心选取很可能不是最优：上一输出不同时，搜索空间也不同

时间步	1	2	3	4
A	0.5	0.1	0.2	0.0
B	0.2	0.4	0.2	0.2
C	0.2	0.3	0.4	0.2
<eos>	0.1	0.2	0.2	0.6

时间步	1	2	3	4
A	0.5	0.1	0.1	0.1
B	0.2	0.4	0.6	0.2
C	0.2	0.3	0.2	0.1
<eos>	0.1	0.2	0.1	0.6

穷举搜索

穷举搜索：能保证搜索到最优解，但复杂度太高

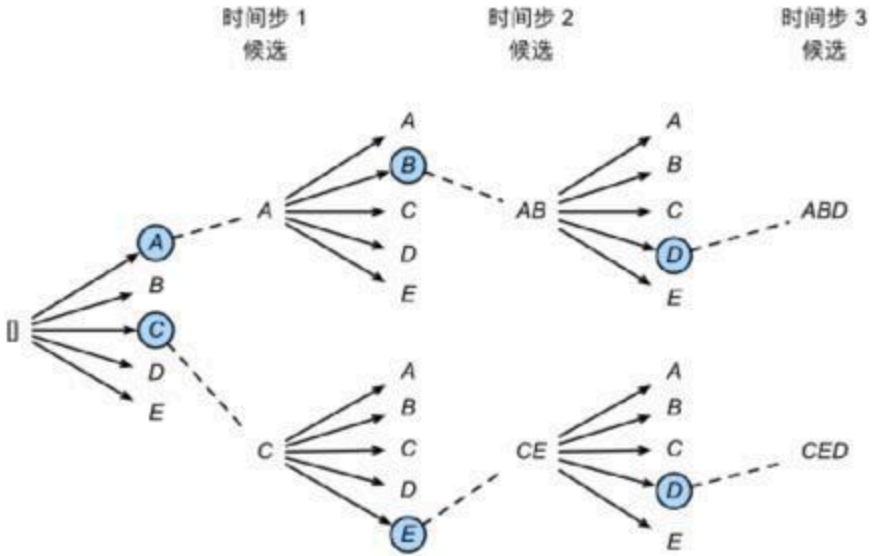
- 穷举：遍历所有可能输出序列
  - 计算概率，选最高的
- 复杂度：$O(|V|^T)$

时间步	1	2	3	4
A	0.5	0.1	0.1	0.1
B	0.2	0.4	0.6	0.2
C	0.2	0.3	0.2	0.1
<eos>	0.1	0.2	0.1	0.6

束搜索

束搜索：贪心、穷举的折衷，每步都只保留k个**候选**

- 第一步后：从个$k|V|$可能中挑选概率最高的k个候选
- 复杂度：$O(k|V|T)$



小结：束搜索

- 束搜索：贪心、穷举的折衷，每步都只保留k个候选
  - $k = 1$：贪心搜索
  - $k = |V|$：穷举搜索

Review

本章内容

情感分析。机器翻译与数据集。编码器-解码器架构。序列到序列学习（seq2seq）。

重点：两种方法实现情感分析；构建机器翻译数据集；编码器-解码器架构；seq2seq；束搜索。

难点：seq2seq 的实现。

学习目标

- 掌握情感分析的两种实现方法：RNN、CNN
- 掌握机器翻译数据集的构建方法
- 理解编码器-解码器架构的设计思路（定长编码）、解决的问题（不定长输入、输出）
- 理解 seq2seq 的数据处理、训练、预测、评测方法
- 理解束搜索的原理

问题

简述编码器-解码器架构的设计思路、解决的问题。

简述 seq2seq 的数据处理、训练、预测、评测方法。

简述束搜索的原理。