深蓝学院
shenlanxueyuan.com

音频语音与语言处理研究组
SLP
Audio, Speech and Language Processing Group NPU

语音合成：从入门到精通

第二讲：前端文本分析

主讲人 陈云琳

出门问问算法工程师

毕业于西北工业大学

目录

1. 语音合成基础与流程
2. 文本分析模块构成
3. 条件随机场(CRF)
4. 基于传统方法的前端文本分析模型
5. 基于神经网络的前端文本分析模型
6. 实战

目录

1. 语音合成基础与流程
2. 文本分析模块构成
3. 条件随机场(CRF)
4. 基于传统方法的前端文本分析模型
5. 基于神经网络的前端文本分析模型
6. 实战

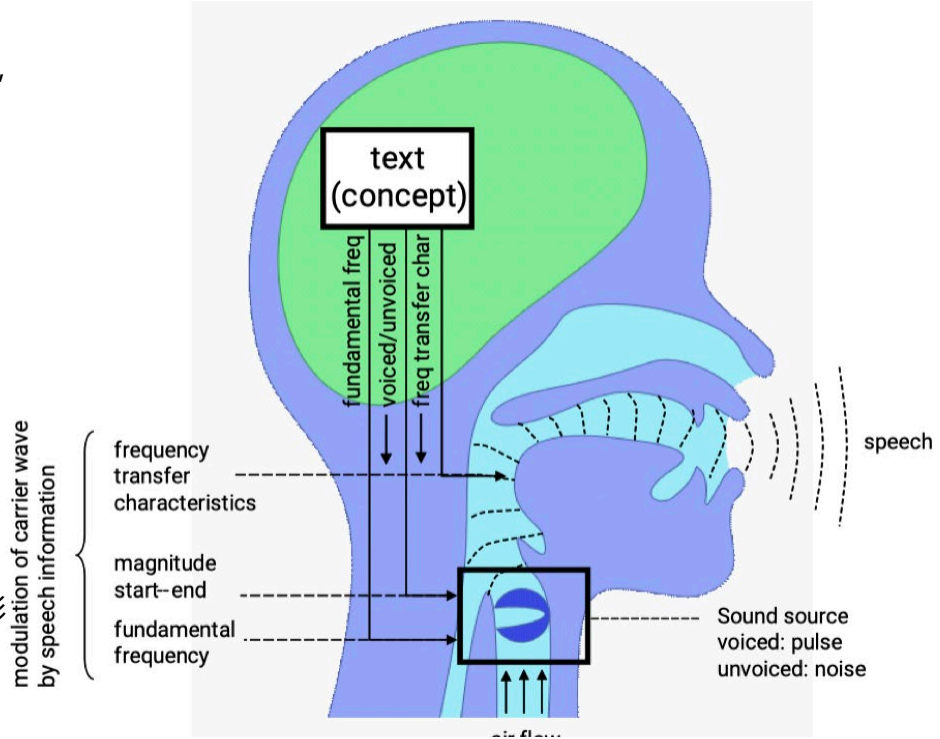
回顾 - 语音产生过程

1.语音产生原理

- 肺部产生气流，引起声带振动，经过喉咙、鼻腔和口腔等，发出声音
- 声道：由声带、声门、口腔、鼻腔等组成，是发声的主要共鸣和调制器官（声门到口唇，约17cm）

2.声音基本概念

- 音调/基音周期 - 声门开启闭合一次时间即振动周期
- 基频 - 基音周期的倒数，声带振动的频率
- 声带振动频率高，音调高；幅度大，响度大。反之则反
- 人的基频范围：50 ~ 550Hz，儿童女性偏高，男性偏低
- 声道功能：谐振腔/产生谐振频率，由每一瞬间的声道外形决定，又称共振峰



3.Text Concept

- 作为语音的条件信息，驱动发音具备规律性、有序性。

语音合成系统框架

输入文本

西北工业大学是唯一 一所以同时发展三航为特色的985好大学。

语言学模型与规则

前端 文本分析Text Analysis

- 文本正则化/Text Normalization(TN)
- 分词/Word Segmentation
- 词性预测/Part-of-speech
- 注音(多音字)/Polyphone
- 韵律预测/Prosody

西北工业大学是唯一一所以同时发展三航为特色的九八五好大学。

西北/工业/大学/是/唯一/一所/以/同时/发展/三航/为/特色/的/九八五/好/大学。

西北/n工业/n大学/n是/v唯一/a一所/a以/v同时/a发展/v三航/n为/v特色/a的/n九八五/n好/a大学/n。

xi1 bei3 gong1 ye4 da4 xue2 shi4 wei2 yi1 yi1 suo3 yi3 tong2 shi2 fa1 zhan3 san1 hang2 wei2 te4 se4 de5 jiu3 ba1 wu3 hao3 da4 xue2 pau0

西北#2工业#1大学#2是#2唯一#1一所#2以#1同时#1发展#2三航为#1特色的#2九八五#1好大学。

后端 语音合成 Acoustic Generation

波形拼接合成

统计参数合成

- 帧级建模
  - 声学模型：文本分析结果 → 声学特征
  - 时长模型
- 声码器：声学特征 → 语音波形采样点

输出声音

输出声音

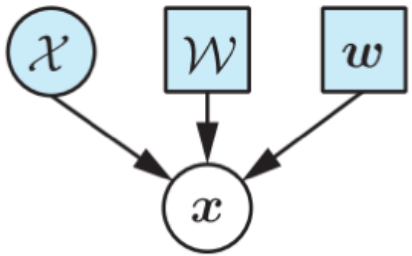
回顾 - 语音合成概率公式

随机变量

$\mathcal{X}$	Speech waveforms(data)	Observed
$\mathcal{W}$	Transcriptions(data)	Observed
w	Given Text	Observed
x	Synthesized speech	Unobserved

Synthesis(合成)

- 估计后验概率分布 -> $p(x \mid w, \mathcal{W}, \mathcal{X})$
- 从后验概率分布采样得到 x



2017 - Heiga et al. - Generative Model-Based Text-to-Speech Synthesis （上文公式来源）

回顾 - 语音合成概率公式

中间变量

O –acoustic feature(声学特征)；

L –linguistic feature(语言学特征)；

λ –模型参数；

o –测试样本acoustic feature；

ℓ –测试样本 linguistic feature

贝叶斯公式

$$p(x, \boldsymbol{\lambda}) = p(x|\boldsymbol{\lambda})p(\boldsymbol{\lambda})$$

后验概率分布

$$p(x \mid w, \mathcal{W}, \mathcal{X}) = \iiint \sum_{\forall l}\sum_{\forall L} p(x, o, \ell, O, L, \lambda \mid w, \mathcal{W}, \mathcal{X}) d_o d_O \, d_\lambda$$

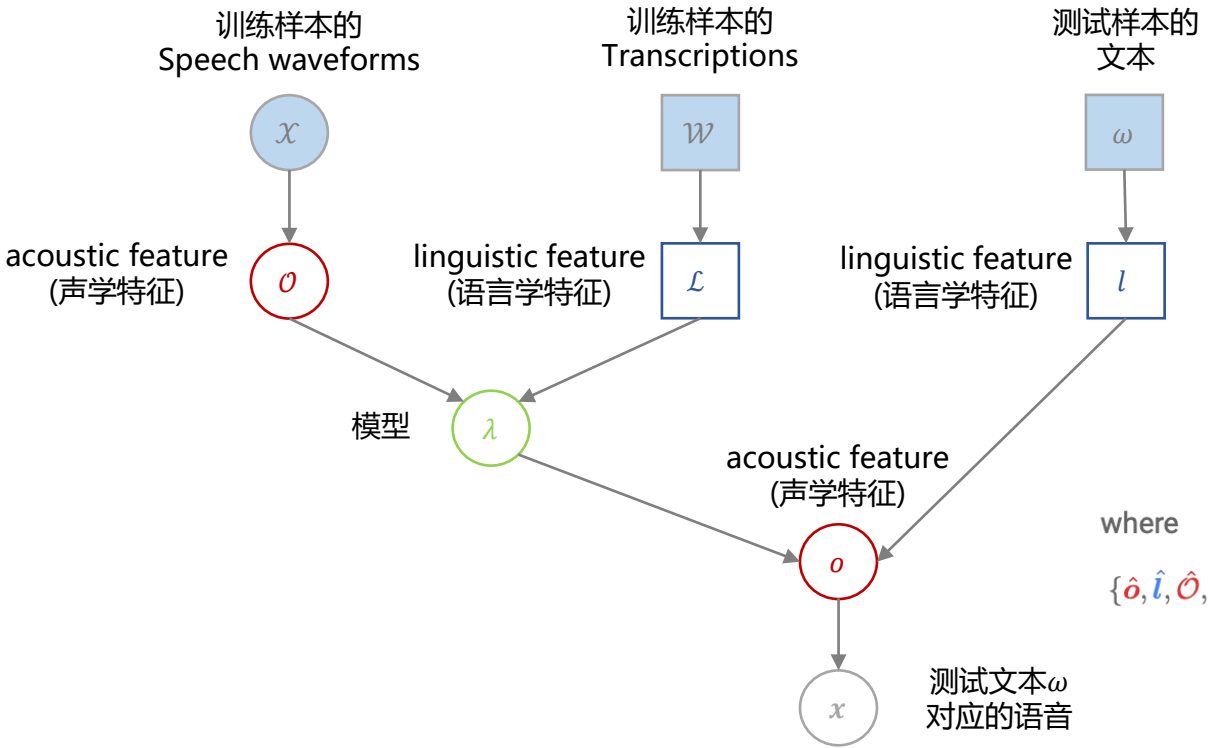
为了推导方便，积分求和忽略：

$$p(x, o, \ell, O, L, \lambda \mid w, \mathcal{W}, \mathcal{X}) = p(x, o, \ell|\lambda, w)p(O, L, \lambda \mid \mathcal{W}, \mathcal{X})$$

$$= p(x|o)p(o|\ell, \lambda)p(\ell|w)\frac{p(\mathcal{X}|O)p(O|L, \lambda)p(\lambda)p(L|\mathcal{W})}{P(\mathcal{X})}$$

回顾 - 语音合成概率公式

$$p(\boldsymbol{x} \mid \boldsymbol{w}, \mathcal{X}, \mathcal{W}) = \iiint \sum_{\forall \boldsymbol{l}} \sum_{\forall \mathcal{L}} \{p(\boldsymbol{x} \mid \boldsymbol{o}) p(\boldsymbol{o} \mid \boldsymbol{l}, \lambda) p(\boldsymbol{l} \mid \boldsymbol{w}) p(\mathcal{X} \mid \mathcal{O}) p(\mathcal{O} \mid \mathcal{L}, \lambda) p(\lambda) p(\mathcal{L} \mid \mathcal{W}) / p(\mathcal{X})\} d\boldsymbol{o} d\mathcal{O} d\lambda$$



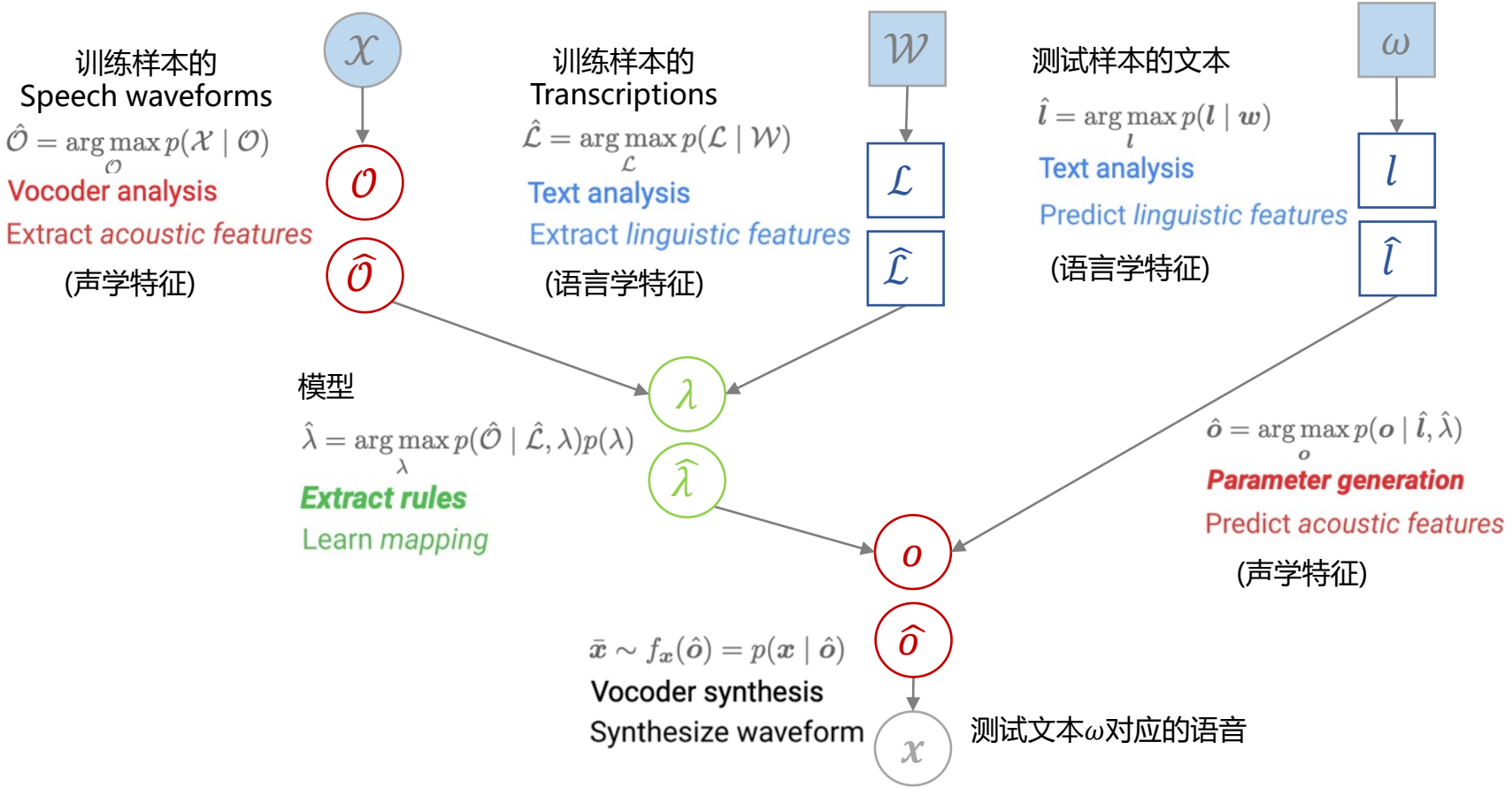
$$p(\boldsymbol{x} \mid \boldsymbol{w}, \mathcal{X}, \mathcal{W}) \approx p(\boldsymbol{x} \mid \hat{\boldsymbol{o}})$$

where

$$\{\hat{\boldsymbol{o}}, \hat{\boldsymbol{l}}, \hat{\mathcal{O}}, \hat{\mathcal{L}}, \hat{\lambda}\} = \arg\max_{\boldsymbol{o}, \boldsymbol{l}, \mathcal{O}, \mathcal{L}, \lambda} \{ p(\boldsymbol{x} \mid \boldsymbol{o}) p(\boldsymbol{o} \mid \boldsymbol{l}, \lambda) p(\boldsymbol{l} \mid \boldsymbol{w}) \; p(\mathcal{X} \mid \mathcal{O}) p(\mathcal{O} \mid \mathcal{L}, \lambda) p(\lambda) p(\mathcal{L} \mid \mathcal{W}) \}$$

2017 - Heiga et al. - Generative Model-Based Text-to-Speech Synthesis （上文公式来源）

回顾 - 语音合成概率公式

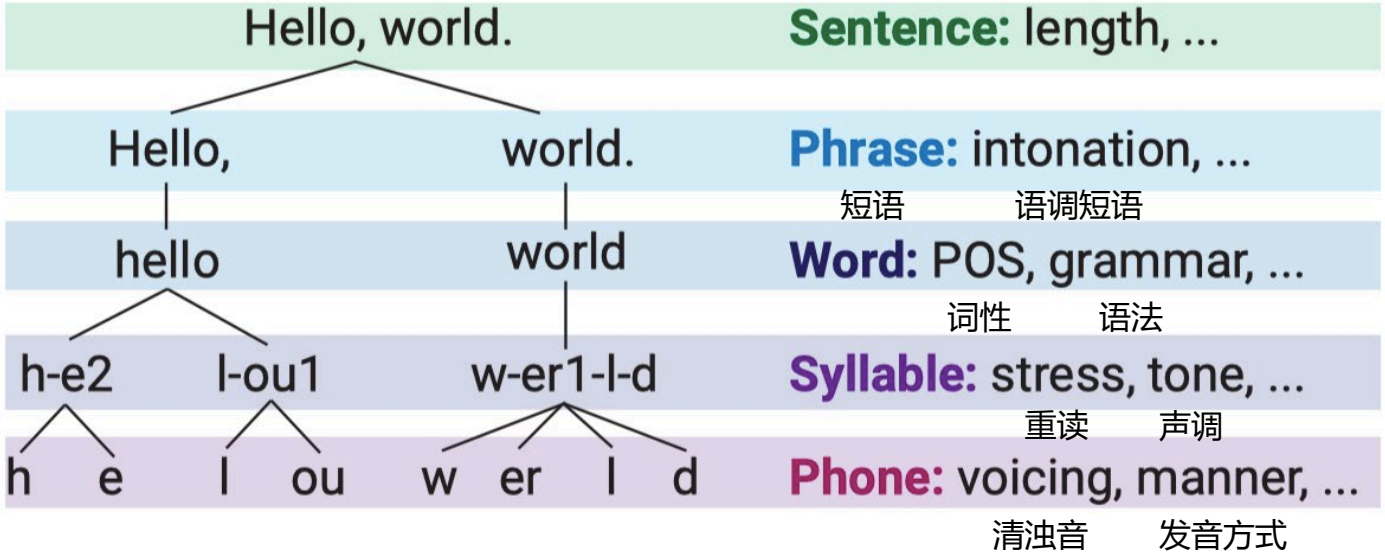


2017 - Heiga et al. - Generative Model-Based Text-to-Speech Synthesis （上文公式来源）

目录

1. 语音合成基础与流程
2. 文本分析模块构成
3. 条件随机场(CRF)
4. 基于传统方法的前端文本分析模型
5. 基于神经网络的前端文本分析模型
6. 实战

Linguistic features(语言学特征)



2017 - Heiga et al. - Generative Model-Based Text-to-Speech Synthesis （上文图来源）

文本分析模块

文本处理以及分析（Text Analysis(TA)）- 前端 (front-end)

- 文本： 90后为中华人民共和国成立70周年准备了大礼
- TN：九零后为中华人民共和国成立七十周年准备了大礼
- 分词：九零后/为/中华人民/共和国/成立/七十/周年/准备/了/大礼
- 词性预测：九零后/n 为/v 中华人民/n 共和国/n 成立/vn 七十/m 周年/n 准备/v 了/u 大礼/n
- 注音：Jiu3 ling2 hou4 wei4 zhong1 hua2 ren2 min2 gong4 he2 guo2…
- 韵律预测：（韵律词，韵律短语，语调短语）

  九零后#1为中华人民#1共和国#2成立七十周年#3准备了大礼#4

文本分析 – 文本正则化

中文文本的非标准词（Non-Standard Words, NSW）处理

- 数字、电话号码: 10086 -> 一千零八十六/幺零零八六
- 时间、比分: 23:20 -> 二十三点二十分/二十三比二十
- 分数、小数、百分比: 3/4 -> 四分之三，3.14 -> 三点一四，15% -> 百分之十五
- 符号、单位: ¥ -> 元， kg -> 千克
- 网址、文件后缀等: www. -> 三w点
- Etc.

英文文本的非标准词（Non-Standard Words, NSW）处理

- 数字: 基数词、序数词 10000 trees -> ten thousand trees
- 单位: 1h2m30s -> one hour two minutes thirty seconds
- 货币: $10 - > ten dollars
- 时间、日期: 12/31/1999 -> december thirty first nineteen ninety nine
- 缩写词: i.e. Mr T vs bros Inc. -> as that is mister t versus brothers incorporated
- 街道地址: 159 W. Popplar Av., Ste. 5, St. George, CA 12345 -> one fifty nine west popplar avenue, suite five, saint george california one two three four five
- Etc.

文本分析 – 分词

例句

- 人要行，干一行行一行，一行行行行行；
  人要是不行，干一行不行一行，一行不行行行不行。
- 佟大为妻子产下一女
- 帝国主义者侵略我们奴役我们，他们要把我们的地瓜分掉！
- 广州市长隆马戏欢迎你
  珠海市长隆马戏欢迎你
- 已结婚的/和尚未结婚的青年
  已结婚的和尚/未结婚的青年

注音 – 字音转换

- 汉语中共有大概1000多个多音字或者多音词
- 多音字：参会、参差；单田芳、千里走单骑
- 变调
  - “一”在非四声前变为四声：一天
  - “一” “不”在四声前变为二声：一寸、不再
  - “一” “不”在词语之间时变为轻声：算一算、行不行
  - 三声相连的词语，前面一个三声变为二声：本领
- 轻声：心里、爸爸、算算、老子
- 儿化音：小孩儿，老头儿
- 方言：四川话/粤语等发音各有不同

注音 – 标注（中文）

- 一般标注文本拼音，“1、2、3、4” 分别对应拼音声调中的“一二三四声”（阴平、阳平、上声、去声），“5”对应轻声，“6”在此文本中表示上声变调后的阳平，如：水果（shui6 guo3）

- “ü”这个音的规定： “ü”统一标为“v”，具体组合如下：

  { j , q , x , y } × { v , van , ve , vn }

  { l , n } × { v , ve }

  Eg：绿色(lǜ sè) - > 绿色(lv4 se4)

- 儿化音的特殊处理：花儿（huar）
- 注：英文一般采用CMU phoneme list

韵律

什么是韵律(prosody)

- In linguistics, prosody is concerned with those elements of speech that are not **individual phonetic segments (vowels and consonants) but are properties of syllables and larger units of speech**, including linguistic functions such as **intonation(语调), tone(音调), stress(重读), and rhythm(节奏/抑扬顿挫)**.
- Prosody may reflect various features of the speaker or the utterance: the **emotional(情感)** state of the speaker; the form of the utterance (statement(陈述), question(疑问), or command(命令)); the presence of irony or sarcasm(讽刺); emphasis(强调), contrast(对比), and focus(重点突出). It may otherwise reflect other elements of language that **may not be encoded by grammar or by choice of vocabulary**. *(from Wikipedia)*

如何表示韵律(prosody)

- ToBI (an abbreviation of tones and break indices) is a set of conventions for transcribing and annotating the prosody of speech. *(from Wikipedia)*
- 中文：主要关注break

韵律 – 中文

九零后#1为中华人民#1共和国#2成立七十周年#3准备了大礼#4

韵律等级结构

- 音素->音节->**韵律词->韵律短语->语调短语**->子句子->主句子->段落->篇章
- LP -> L0 -> **L1(#1) -> L2(#2) -> L3(#3)** -> L4(#4) -> L5 - > L6 -> L7

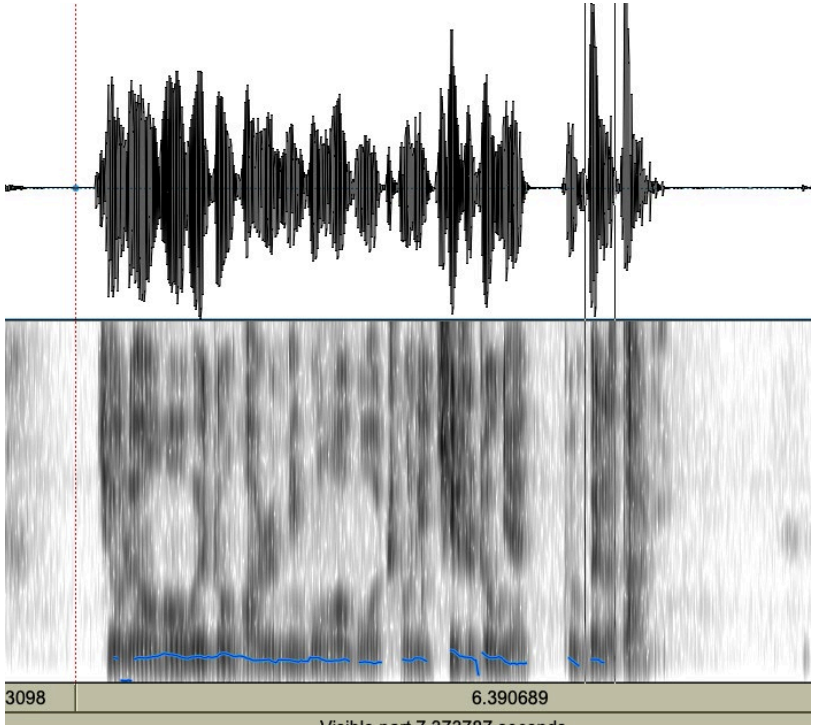
标注韵律等级

	停顿时长	前后音高特征
韵律词边界	不停顿或从听感上察觉不到停顿	无
韵律短语边界	可以感知停顿，但无明显的静音段	音高不下倾或稍下倾，韵末不可做句末
语调短语边界	有较长停顿	音高下倾比较完全，韵末可以作为句末

韵律 – 中文

音高

- 九零后为中华人民共和国唱了一首生日歌



目录

1. 语音合成基础与流程
2. 文本分析模块构成
3. 条件随机场(CRF)
4. 基于传统方法的前端文本分析模型
5. 基于神经网络的前端文本分析模型
6. 实战

条件随机场(CRF)

随机场：若干个位置组成的整体，当给每一个位置中按照某种分布随机赋予一个值。

马尔可夫随机场：随机场的特例，假设随机场中某一个位置的值仅仅与其相邻的位置的值有关，与其不相邻的位置的值无关。

条件随机场：马尔可夫随机场的特例，假设马尔可夫随机场中只有两种变量$\boldsymbol{x}$和$\boldsymbol{y}$，$P(\boldsymbol{y}|\boldsymbol{x})$是给定$\boldsymbol{x}$时$\boldsymbol{y}$的条件概率分布，若$\boldsymbol{y}$构成的是一个马尔可夫随机场，则称条件概率分布$P(\boldsymbol{y}|\boldsymbol{x})$是条件随机场。

线性链条件随机场： CRF定义中，没有要求$\boldsymbol{x}$和$\boldsymbol{y}$有相同的结构。实现中，一般假设$\boldsymbol{x}$和$\boldsymbol{y}$有相同的结构，即

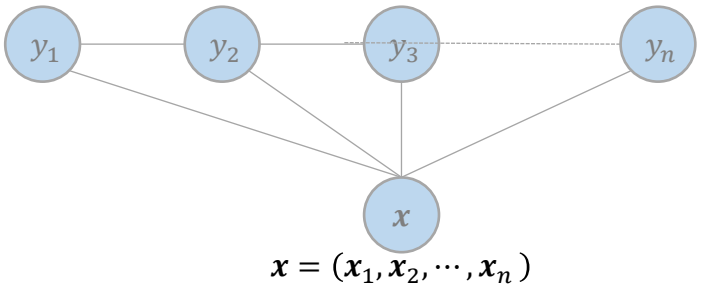
$$\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_n), \qquad \boldsymbol{y} = (y_1, y_2, \cdots, y_n)$$

$\boldsymbol{x}$和$\boldsymbol{y}$有相同结构的CRF就构成了**线性链条件随机场(Linear chain Conditional Random Fields, linear-CRF)**。

$P(\boldsymbol{y}|\boldsymbol{x})$条件概率分布构成的条件随机场，满足马尔可夫性：

$$P(y_i|\boldsymbol{x}, y_1, y_2, \cdots, y_n) = P(y_i|\boldsymbol{x}, y_{i-1}, y_{i+1})$$

其中，$i = 1,2, \ldots, n$，则称P$(\boldsymbol{y}|\boldsymbol{x})$为线性链条件随机场。



条件随机场(CRF)

参数化

目的：将线性链条件随机场$P(\boldsymbol{y}|\boldsymbol{x})$转化为可以学习的机器学习模型。通过特征函数及其权重系数来定义。

特征函数

状态特征函数：定义在$\boldsymbol{y}$节点上的节点特征函数，只和当前节点有关。

$$s_l(y_i, x, i), \quad l = 1, 2, \ldots L$$

其中，L是定义在该节点的状态特征函数的总个数，

i是当前节点在序列的位置。

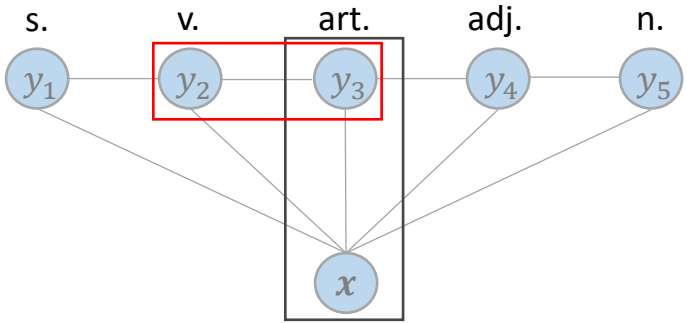
转移特征函数：定义在$\boldsymbol{y}$上下文的特征函数，依赖于当前和前一个节点。

$$t_k(y_{i-1}, y_i, x, i), \quad k = 1, 2, \ldots K$$

其中，K是定义在该节点的局部特征函数的总个数，

i是当前节点在序列的位置。

通常特征函数t_k和s_l取值为1或0; 当满足特征条件时取1，否则为0。



$$\boldsymbol{x} = (\boldsymbol{x_1}, \boldsymbol{x_2}, \boldsymbol{x_3}, \boldsymbol{x_4}, \boldsymbol{x_5})$$

You are a lovely girl

条件随机场(CRF)的特征函数

直观理解

特征函数可以理解为一种序列与标签位置关系的规定，其本质上是一种规则。

以词性标注任务为例，特征函数可以用来定义：对于一个序列来说，如果前一个词的标签为动词，那么后面一个词的标签就是名词。即特征函数定义了一个规则：动词后面跟名词。该规则作为一个函数，需要有输出，最简单的方式为“满足规则输出1，不满足规则输出0”。

我爱语音合成

【，我】　　空→名词，0

【我，爱】　　名词→动词，0

【爱，语音合成】　动词→名词，1

一个特征函数代表了一个规则，我们可以定义多个规则，即多个特征函数。

条件随机场(CRF)

权值

不同规则的组合方式（加权值）

一个训练好的模型，可以看作是多个规则的集合。那么，

（1）每个规则在这个模型中的价值(重要性)都是相同的吗?

（2）如果不相同，怎么得到每个规则的价值?

Linear-CRF的参数化形式

$$P(y|x) = \frac{1}{Z(x)} exp\Big(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i) \Big)$$

其中，$Z(x)$为规范化因子（求和是在所有可能的输出序列上进行）：

$$Z(x) = \sum_{y} exp\Big(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i) \Big)$$

回到特征函数本身，每个特征函数定义了一个linear-CRF的规则，其权重系数定义了这个规则的可信度。所有的规则和其可信度一起构成了linear-CRF的条件概率分布。**条件随机场完全由特征函数t_k，s_l和对应的权值λ_k，μ_l确定。**

例子

假设有一标注问题：输入观测序列为 $\boldsymbol{x}=(\boldsymbol{x}_1,\boldsymbol{x}_2,\boldsymbol{x}_3)$, 输出标记序列为 $\boldsymbol{y}=(y_1,y_2,y_3)$， y_i的取值为 {1, 2}。

特征函数以及对应权值，如下方所示。对给定观测序列$\boldsymbol{x}$，求标记序列为$\boldsymbol{y}$=(1,2,2)的非规范化条件概率（即没有除以规范化因子的条件概率）。

$t_1=t_1(y_{i-1}=1,y_i=2,\boldsymbol{x},i),i=2,3,\quad \lambda_1=1$

$t_2=t_2(y_1=1,y_2=1,\boldsymbol{x},2)\quad \lambda_2=0.6$

$t_3=t_3(y_2=2,y_3=1,\boldsymbol{x},3)\quad \lambda_3=1$

$t_4=t_4(y_1=2,y_2=1,\boldsymbol{x},2)\quad \lambda_4=1$

$t_5=t_5(y_2=2,y_3=2,\boldsymbol{x},3)\quad \lambda_5=0.2$

$s_1=s_1(y_1=1,\boldsymbol{x},1)\quad u_1=1$

$s_2=s_2(y_i=2,\boldsymbol{x},i),i=1,2,u_2=0.5$

$s_3=s_3(y_i=1,\boldsymbol{x},i),i=2,3,u_3=0.8$

$s_4=s_4(y_3=2,\boldsymbol{x},3)\quad u_4=0.5$

$$P(y_1=1,y_2=2,y_3=2|\boldsymbol{x})\propto\exp\left[\sum_{k=1}^{5}\lambda_k\sum_{i=2}^{3}t_k(y_{i-1},y_i,\boldsymbol{x},i)+\sum_{l=1}^{4}u_l\sum_{i=1}^{3}s_l(y_i,\boldsymbol{x},i)\right]$$

$$\propto\exp[\lambda_1t_1(y_1=1,y_2=2,\boldsymbol{x},2)+\lambda_5t_5(y_2=2,y_3=2,\boldsymbol{x},3)+u_1s_1(y_1=1,\boldsymbol{x},1)+u_2s_2(y_2=2,\boldsymbol{x},2)+u_4s_4(y_3=2,\boldsymbol{x},3)]$$

$$\propto\exp[1\times1+1\times0.2+1\times1+1\times0.5+1\times0.5]$$

$$\propto\exp(3.2)$$

条件随机场(CRF)

Linear-CRF的模型参数学习

给定：

训练数据集 $\boldsymbol{x}$

对应的标记序列 $\boldsymbol{y}$

特征函数集合$\{t_k(y_{i-1}, y_i, \boldsymbol{x}, i), s_l(y_i, \boldsymbol{x}, i)\}$

学习：

求解模型参数$\boldsymbol{w}$，即权重系数$\{\lambda_k\}$，$\{\mu_l\}$，使得$P(\boldsymbol{y}|\boldsymbol{x})$取得最大。

对于给定的训练数据集$\boldsymbol{x}$，以及对应的标定序列为$\boldsymbol{y}$的条件概率为：

$$P(\boldsymbol{y}|\boldsymbol{x}) = \prod_{\mathbf{x},y} P_w(y|\mathbf{x})^{\overline{P}(\mathbf{x},y)}$$

经验分布，训练样本中$(\boldsymbol{x}, y)$出现的概率

其中，条件概率为：

$$P_w(y|\mathbf{x}) = P(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} exp\Big(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, \mathbf{x}, i) + \sum_{i,l} \mu_l s_l(y_i, \mathbf{x}, i)\Big)$$

Linear-CRF的模型参数学习

负对数似然函数$L(\boldsymbol{w})$:

$$L(w) = -logP(\boldsymbol{y}|\boldsymbol{x})$$

模型参数学习的目标:

通过学习模型参数（权重系数），使得负对数似然函数取得最小值。采用梯度下降法，求得模型参数 $\boldsymbol{w}$。

Linear-CRF的模型解码

参数学习阶段，求得权重系数。给定一个观测序列$\boldsymbol{x}$，要求出满足$P(\boldsymbol{y}|\boldsymbol{x})$最大的序列$\boldsymbol{y}$，可采用维特比算法（不细讲）。

条件随机场(CRF)用途

定义：已知一组输入随机变量条件下，另一组输出随机变量的条件概率分布模型

CRF用于文本分析的哪些任务?

- 分词
  - 分类标签：词首(B)、词中(M)、词尾(E)、单字词(S)
  - 标注：我/S 爱/S 学/B 习/E 语/B 音/M 合/M 成/E
- 词性标注

  欢迎大家学习深蓝学院的语音合成课程。
  - 标注：欢迎/v 大家/n 学习/v 深蓝学院/n 的/u 语音合成/n 课程/n
  - 标注：欢迎/动词 大家/名词 学习/动词 深蓝学院/名词 的/助词 语音合成/名词 课程/名词
- 注音
- 韵律

  •韵律词(prosody word PW)->韵律短语(prosody phrase PPH)->语调短语(intonational phrase IPH)

目录

1. 语音合成基础与流程
2. 文本分析模块构成
3. 条件随机场(CRF)
4. 基于传统方法的前端文本分析模型
5. 基于神经网络的前端文本分析模型
6. 实战

文本分析 – 文本正则化

正则化的复杂性

- 语种之间基本独立
- 规则种类多（100-200大类规则很常见），维护麻烦
- TN任务在学术界和业界不是那么重要，但是很关键

正则化工具推荐

- https://github.com/google/re2, RE2 is a fast, safe, thread-friendly alternative to backtracking regular expression engines like those used in PCRE, Perl, and Python. It is a C++ library.
- https://github.com/speechio/chinese_text_normalization

文本分析 – 分词

基于最大前向匹配的分词方法

- 顾名思义，就是从待分词句子的左边向右边搜索，寻找词的最大匹配。我们需要规定一个词的最大长度，每次扫描的时候寻找当前开始的这个长度的词来和字典中的词匹配，如果没有找到，就缩短长度继续寻找，直到找到字典中的词或者成为单字。

基于Conditional Random Field(CRF)的分词方法

- CRF 序列标注任务：四个tag（B，E，M，S），B表示词的begin，E表示词的end，M表示词中，S表示单个词single

词性标注：查字典和CRF(和分词一致，不多赘述)

分词 – 基于Trie Tree最大前向匹配

Trie Tree，又称单词字典树、查找树，是一种树形结构

性质：

- 根节点不包含字符，除根节点外每一个节点都只包含一个字符。
- 从根节点到某一节点，路径上经过的字符连接起来，为该节点对应的字符串。
- 每个节点的所有子节点包含的字符都不相同。

例子：

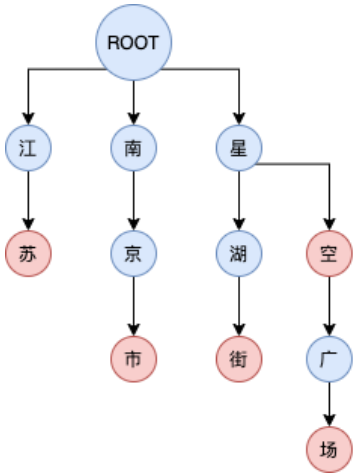
- 江苏/星空/星空广场

优点：

- 最大限度地减少无谓的字符串比较，查询效率为O(m), 其中m是字符串的长度

缺点：

- 空间消耗比较大



IsEnd 表示是否为词的尾节点
蓝色节点：IsEnd = False
红色节点：IsEnd = True

分词 – CRF

分词可以看作分类问题，每个词的类别标签：词首(B)、词中(M)、词尾(E)、单字词(S)。

标注

语句实例：我爱学习语音合成。

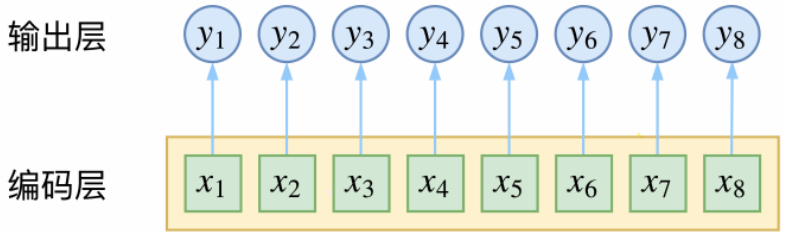
分词结果：我/爱/学习/语音合成。

标注：我/S 爱/S 学/B 习/E 语/B 音/M 合/M 成/E。

如何解决?

思路1： 每个字相互独立，采用常见的分类器；

思路2：考虑相邻字的标注信息，采用CRF等分类器。



分词 – CRF

CRF(条件随机场): 可用于NLP中序列标注，包括分词、词性等标注。

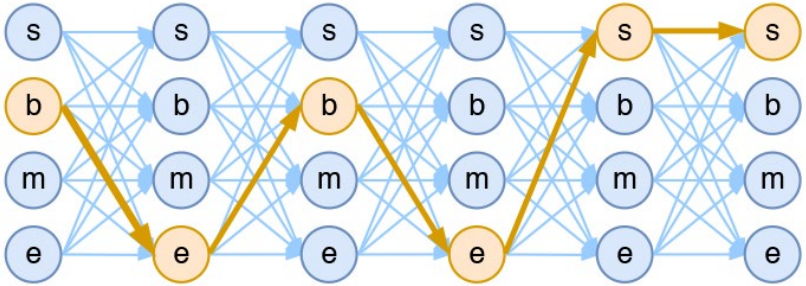
算法概述

(1) 考虑输出序列的上下文关系，比如S后面不能接M和E等

(2) 输入有n帧，标签k种可能，路径条数：k^n次

(3) 根据每个状态的输出概率/状态之间的转移概率，获取最佳路径（维特比算法）

(4) 今天/天气/不/错（对应的输出序列：BEBESS）



参考资料

博客 https://www.cnblogs.com/pinard/p/7048333.html 推荐指数 ★★★★★

资料 https://homepages.inf.ed.ac.uk/csutton/publications/crftut-fnt.pdf

分词 – CRF

CRF 和 词典分词对比

	词典分词	CRF
优点	• 查询效率高 • 可扩展性强 • 可以维护不同的user dict，不同的domain • 快速fix bug	• 基于统计学习模型，依赖上下文语境、较强泛化能力 • 对于歧义词和未登陆词，具备较强的识别能力
缺点	• 过于依赖字典 • 对于未出现的词，分词能力较低	• 增加模型训练时间 • 性能略低于字典分词 • 不能快速fix bug

分词 – CRF

Tools： CRF++

特征模版

数据：每个token包含3列，分别为字本身、字类型（英文数字，汉字，标点等）和词位标记

模版文件：%x[row,col]， row用于确定与当前的token的相对行数；col用于确定绝对列数。

模版类型：Unigram、Bigram

```
IMovie ASCII S
是 CN S
一 CN S
款 CN S
不 CN B  >> 当前token
错 CN E
的 CN S
应 CN B
用 CN E
特征模板形式为：
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-1,0]/%x[0,0]
U06:%x[0,0]/%x[1,0]
U07:%x[-1,0]/%x[1,0]
U08:%x[0,1]
U09:%x[-1,1]/%x[0,1]
# Bigram
B00:%x[0,1]
```

http://crfpp.sourceforge.net/

分词 – CRF

$$P_w(y|\mathbf{x}) = P(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} exp\Big(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, \mathbf{x}, i) + \sum_{i,l} \mu_l s_l(y_i, \mathbf{x}, i) \Big)$$

音频语音与语言处理研究组 SLP Audio, Speech and Language Processing Group NPU

Unigram – 特征函数 $s_l(y_i, \boldsymbol{x}, i)$

当i为右图中当前token时，

func0 = if (output = B and feature= “U00:一") return 1 else return 0

func1 = if (output = B and feature= “U01:款") return 1 else return 0

func2 = if (output = B and feature= “U02:不") return 1 else return 0

…

func5 = if (output = B and feature= “U05:款/不”) return 1 else return 0

…

func9 = if (output = B and feature= “U09:CN/CN") return 1 else return 0

```
IMovie ASCII S
是 CN S
一 CN S
款 CN S
不 CN B  >> 当前token
错 CN E
的 CN S
应 CN B
用 CN E
特征模板形式为:
# Unigram
U00:%x[-2,0]
U01:%x[-1,0]
U02:%x[0,0]
U03:%x[1,0]
U04:%x[2,0]
U05:%x[-1,0]/%x[0,0]
U06:%x[0,0]/%x[1,0]
U07:%x[-1,0]/%x[1,0]
U08:%x[0,1]
U09:%x[-1,1]/%x[0,1]
# Bigram
B00:%x[0,1]
```

Bigram – 特征函数 $t_k(y_{i-1}, y_i, \boldsymbol{x}, i)$

当i为右图中当前token时，

func0 = if ($output_{-1}$ = S and output=B and feature= “U00:CN") return 1 else return 0

http://crfpp.sourceforge.net/

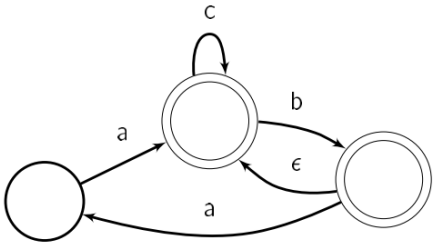
注音 – g2p常用到的概念

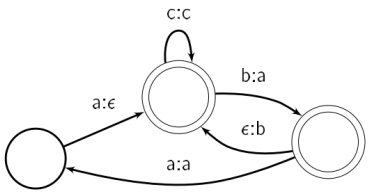
n-gram模型：基于统计语言模型的算法，它的基本思想是将文本里面的内容按照字节进行大小为N的滑动窗口操作，形成了长度是N的字节片段序列。该模型基于假设，第N个词的出现只与前面N-1个词相关。整句的概率就是各个词出现概率的乘积。

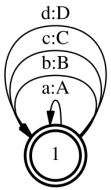
- 一元模型（unigram model）：$P(w_1, w_2, \cdots, w_m) = \prod_{i}^{m} P(w_i)$
- 二元模型（bigram model）：$P(w_1, w_2, \cdots, w_m) = \prod_{i=1}^{m} P(w_i|w_{i-1})$ p(语音) = p(音|语) * p(语)

FST(Finite-state Transducers, 有限状态转换器)

- **FSA(Finite-state Acceptor, 有限状态接收器)**，如右图上
  - 一个开始状态，至少一个结束状态
  - 问题：能表示无穷的字符串吗?
- **FST**，如右图下
  - 相比FSA，边上有输入符号和输出符号
  - 把输入符号替换为输出符号
  - 例子：





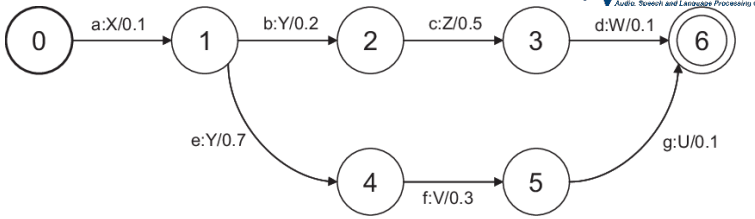


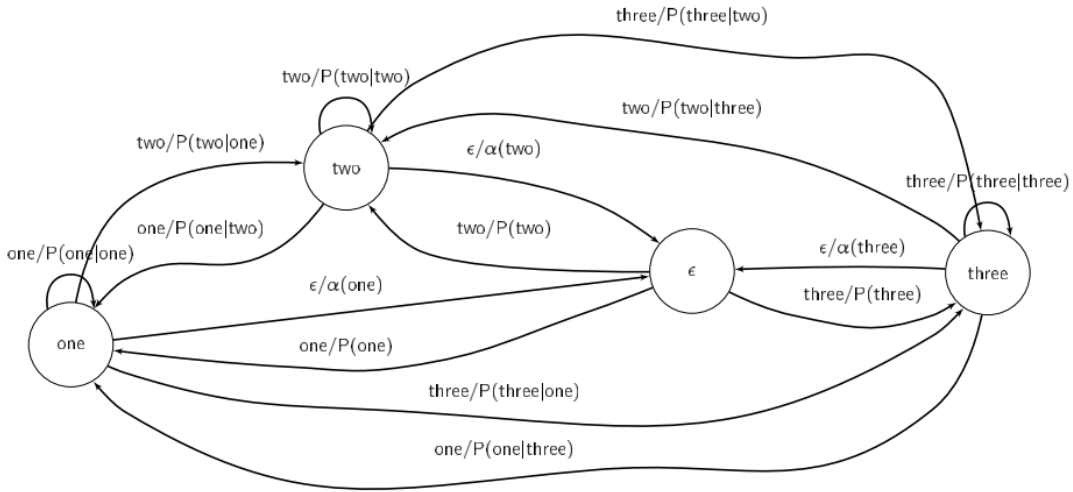
注音 – g2p常用到的概念

Weighted-FST(WFST)

- 比FST在边上多了权重分数，如右图所示
- Bigram – WFST表示， 如下图所示

注：只是示意图，缺少output symbol 以及开始、结束状态。





注音 – g2p (Phonetisaurus)

目标：给定一个词，从character映射到phoneme， PHONIX -> /f i n I k s/

传统方法

- 查字典
- How to solve OOV(Out-of-Vocabulary)?

模型方法

- Phonetisaurus：基于Openfst
- 三个子问题
  - 对齐问题：在PHONIX中PH->f, O->i, N->n, I->I, X->{k,s}
  - 训练问题：根据对齐好的数据，生成 joint sequence的n-gram model.
    - 一元模型（unigram model）
    - 二元模型（bigram model）
  - 解码问题：根据模型，根据fst最短路径算法，生成phoneme

2012. WFST-based Grapheme-to-Phoneme Conversion: Open Source Tools for Alignment, Model-Building and Decoding

2019.binbinzhang - http://robin1001.github.io/2019/09/10/g2p/

Ngram model: https://zhuanlan.zhihu.com/p/32829048

注音 – g2p

实验数据

- 训练数据：35万来自各大新闻网站的中文训练语料。
- 测试数据：common集合和多音字集合，每个集合1万句
  - common集不包含多音字
  - 多音字集，每个句子都有多音字

实验结果

- 纯G2P模型

common	SAR(%)	CAR(%)
WithTone	73.3974	94.5999
IgnoreAllTone	99.0385	99.8527
IgnoreSoftTone	86.859	97.2509

- G2P模型 + 字典 + 规则

common	SAR(%)	CAR(%)
WithTone	79.8077	95.8763
IgnoreAllTone	99.0385	99.8527
IgnoreSoftTone	93.9103	98.4782

- 注：SAR和CAR分别表示句子和字的准确率，WithTone表示测试准确率时加上声调，IgnoreAllTone和IgnoreSoftTone表示忽略所有读音和忽略轻音

注音 – g2p

其他模型方法

- 对于中文的任务，字和拼音一 一对应，可以采用CRF序列标注任务
- Sequence-to-sequence：Transformer g2p

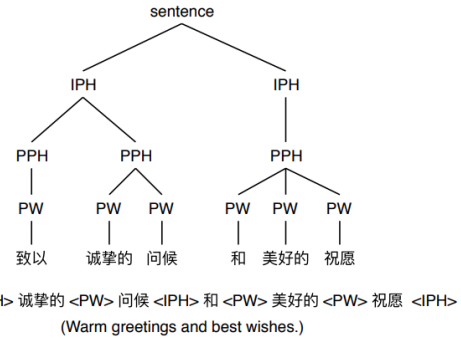
Demo

https://github.com/kakaobrain/g2pM

韵律 – 模型介绍

CRF 模型

- 韵律词(prosody word PW)->韵律短语(prosody phrase PPH)->语调短语(intonational phrase IPH)
- 级联CRF model
- 如果韵律等级预测是PW，会继续预测是否是PPH；如果是PPH，会继续预测IPH
- CRF 模型训练：韵律具体用法和分词一样，template多加一些特征即可（比如词性）



致以 <PPH> 诚挚的 <PW> 问候 <IPH> 和 <PW> 美好的 <PW> 祝愿 <IPH>
(Warm greetings and best wishes.)

Boundary	P (%)	R (%)	F (%)
PW	95.34	96.73	96.03
PPH	83.41	83.68	83.06
IPH	84.85	73.39	78.71

Table 1. The results of CRF-based prosody prediction.

2015 .Ding Et.al. - AUTOMATIC PROSODY PREDICTION FOR CHINESE SPEECH SYNTHESIS USING BLSTM-RNN AND EMBEDDING FEATURES

前端文本分析结果：抄本


```
0 3500000 XX^XX-SIL+k=ao@X_X/A:X_X_X+X/B:X-X=X-X@X-X&X-X#X-X|X/C:4+X+2_X/D:X-X/E:X_X@X+X&X+X#X+X/F:2=1/G:X_X/H:X=X^X=X|X/I:4=3/J:16+10-4$
3500000 4431312 XX^SIL-k+ao=z@1_2/A:X_X_X+X/B:4-X=2-v@1-1&1-4#6-0|ao/C:3+uo+2_f/D:X-X/E:2_1@1+3&X+X#X+X/F:2=1/G:X_X/H:4=3^1=4|X/I:6=4/J:16+10-4$
4431312 5680483 SIL^k-ao+z=uo@2_1/A:X_X_X+X/B:4-X=2-v@1-1&1-4#0-2|ao/C:3+uo+2_f/D:X-X/E:2_1@1+3&X+X#X+X/F:2=1/G:X_X/H:4=3^1=4|X/I:6=4/J:16+10-4$
5680483 6121631 k^ao-z+uo=x@1_2/A:4_ao_2+v/B:3-X=2-f@1-1&2-3#2-0|uo/C:4+ia+2_v/D:2-1/E:2_1@2+2&X+X#X+X/F:4=2/G:X_X/H:4=3^1=4|X/I:6=4/J:16+10-4$
6121631 7306675 ao^z-uo+x=ia@2_1/A:4_ao_2+v/B:3-X=2-f@1-1&2-3#0-2|uo/C:4+ia+2_v/D:2-1/E:2_1@2+2&X+X#X+X/F:4=2/G:X_X/H:4=3^1=4|X/I:6=4/J:16+10-4$
7306675 8153613 z^uo-x+ia=p@1_2/A:3_uo_2+f/B:4-X=2-v@1-2&3-2#2-0|ia/C:1+o+2_v/D:2-1/E:4_2@3+1&X+X#X+X/F:4=2/G:X_X/H:4=3^1=4|X/I:6=4/J:16+10-4$
8153613 9491280 uo^x-ia+p=o@2_1/A:3_uo_2+f/B:4-X=2-v@1-2&3-2#0-1|ia/C:1+o+2_v/D:2-1/E:4_2@3+1&X+X#X+X/F:4=2/G:X_X/H:4=3^1=4|X/I:6=4/J:16+10-4$
9491280 10457192 x^ia-p+o=lp@1_2/A:4_ia_2+v/B:1-X=2-v@2-1&4-1#1-0|o/C:X+X+X_X/D:2-1/E:4_2@3+1&X+X#X+X/F:4=2/G:X_X/H:4=3^1=4|X/I:6=4/J:16+10-4$
10457192 12363202 ia^p-o+lp=q@2_1/A:4_ia_2+v/B:1-X=2-v@2-1&4-1#0-5|o/C:X+X+X_X/D:2-1/E:4_2@3+1&X+X#X+X/F:4=2/G:X_X/H:4=3^1=4|X/I:6=4/J:16+10-4$
12363202 14314374 p^o-lp+q=ian@X_X/A:1_o_2+v/B:X-X=X-X@X-X&X-X#X-X|X/C:2+ian+2_f/D:4-2/E:X_X@X+X&X+X#X+X/F:4=2/G:4_3/H:X=X^X=X|X/I:6=4/J:16+10-4$
14314374 15470586 o^lp-q+ian=f@1_2/A:X_X_X+X/B:2-X=2-f@1-2&1-6#5-0|ian/C:1+ang+2_f/D:4-2/E:4_2@1+4&X+X#X+X/F:2=1/G:4_3/H:6=4^2=3|X/I:2=1/J:16+10-4$
15470586 16636417 lp^q-ian+f=ang@2_1/A:X_X_X+X/B:2-X=2-f@1-2&1-6#0-1|ian/C:1+ang+2_f/D:4-2/E:4_2@1+4&X+X#X+X/F:2=1/G:4_3/H:6=4^2=3|X/I:2=1/J:16+10-4
16636417 17080448 q^ian-f+ang=b@1_2/A:2_ian_2+f/B:1-X=2-f@2-1&2-5#1-0|ang/C:4+ang+2_v/D:4-2/E:4_2@1+4&X+X#X+X/F:2=1/G:4_3/H:6=4^2=3|X/I:2=1/J:16+10-
17080448 18949082 ian^f-ang+b=ang@2_1/A:2_ian_2+f/B:1-X=2-f@2-1&2-5#0-2|ang/C:4+ang+2_v/D:4-2/E:4_2@1+4&X+X#X+X/F:2=1/G:4_3/H:6=4^2=3|X/I:2=1/J:16+1
18949082 19395997 f^ang-b+ang=sh@1_2/A:1_ang_2+f/B:4-X=2-v@1-1&3-4#2-0|ang/C:1+an+2_n/D:4-2/E:2_1@2+3&X+X#X+X/F:2=1/G:4_3/H:6=4^2=3|X/I:2=1/J:16+10-
19395997 20821408 ang^b-ang+sh=an@2_1/A:1_ang_2+f/B:4-X=2-v@1-1&3-4#0-2|ang/C:1+an+2_n/D:4-2/E:2_1@2+3&X+X#X+X/F:2=1/G:4_3/H:6=4^2=3|X/I:2=1/J:16+10
20821408 21366355 b^ang-sh+an=x@1_2/A:4_ang_2+v/B:1-X=2-n@1-1&4-3#2-0|an/C:3+ian+2_n/D:2-1/E:2_1@3+2&X+X#X+X/F:4=2/G:4_3/H:6=4^2=3|X/I:2=1/J:16+10-4
21366355 22816665 ang^sh-an+x=ian@2_1/A:4_ang_2+v/B:1-X=2-n@1-1&4-3#0-2|an/C:3+ian+2_n/D:2-1/E:2_1@3+2&X+X#X+X/F:4=2/G:4_3/H:6=4^2=3|X/I:2=1/J:16+10
22816665 23648058 sh^an-x+ian=l@1_2/A:1_an_2+n/B:3-X=2-n@1-2&5-2#2-0|ian/C:4+u+2_n/D:2-1/E:4_2@4+1&X+X#X+X/F:4=2/G:4_3/H:6=4^2=3|X/I:2=1/J:16+10-4$
23648058 24931136 an^x-ian+l=u@2_1/A:1_an_2+n/B:3-X=2-n@1-2&5-2#0-1|ian/C:4+u+2_n/D:2-1/E:4_2@4+1&X+X#X+X/F:4=2/G:4_3/H:6=4^2=3|X/I:2=1/J:16+10-4$
24931136 25320384 x^ian-l+u=lp@1_2/A:3_ian_2+n/B:4-X=2-n@2-1&6-1#1-0|u/C:X+X+X_X/D:2-1/E:4_2@4+1&X+X#X+X/F:4=2/G:4_3/H:6=4^2=3|X/I:2=1/J:16+10-4$
25320384 27042965 ian^l-u+lp=q@2_1/A:3_ian_2+n/B:4-X=2-n@2-1&6-1#0-5|u/C:X+X+X_X/D:2-1/E:4_2@4+1&X+X#X+X/F:4=2/G:4_3/H:6=4^2=3|X/I:2=1/J:16+10-4$
27042965 29452222 l^u-lp+q=ian@X_X/A:4_u_2+n/B:X-X=X-X@X-X&X-X#X-X|X/C:2+ian+2_f/D:4-2/E:X_X@X+X&X+X#X+X/F:4=2/G:6_4/H:X=X^X=X|X/I:2=1/J:16+10-4$
29452222 30570950 u^lp-q+ian=f@1_2/A:X_X_X+X/B:2-X=2-f@1-2&1-2#5-0|ian/C:1+ang+2_f/D:4-2/E:4_2@1+1&X+X#X+X/F:4=2/G:6_4/H:2=1^3=2|X/I:4=2/J:16+10-4$
30570950 31809666 lp^q-ian+f=ang@2_1/A:X_X_X+X/B:2-X=2-f@1-2&1-2#0-1|ian/C:1+ang+2_f/D:4-2/E:4_2@1+1&X+X#X+X/F:4=2/G:6_4/H:2=1^3=2|X/I:4=2/J:16+10-4
31809666 32418046 q^ian-f+ang=f@1_2/A:2_ian_2+f/B:1-X=2-f@2-1&2-1#1-0|ang/C:3+an+2_f/D:4-2/E:4_2@1+1&X+X#X+X/F:4=2/G:6_4/H:2=1^3=2|X/I:4=2/J:16+10-4
32418046 34336879 ian^f-ang+f=an@2_1/A:2_ian_2+f/B:1-X=2-f@2-1&2-1#0-3|ang/C:3+an+2_f/D:4-2/E:4_2@1+1&X+X#X+X/F:4=2/G:6_4/H:2=1^3=2|X/I:4=2/J:16+10-
34336879 35377757 f^ang-f+an=x@1_2/A:1_ang_2+f/B:3-X=2-f@1-2&1-4#3-0|an/C:4+iang+2_f/D:4-2/E:4_2@1+2&X+X#X+X/F:4=2/G:2_1/H:4=2^4=1|X/I:X=X/J:16+10-4
35377757 36897602 ang^f-an+x=iang@2_1/A:1_ang_2+f/B:3-X=2-f@1-2&1-4#0-1|an/C:4+iang+2_f/D:4-2/E:4_2@1+2&X+X#X+X/F:4=2/G:2_1/H:4=2^4=1|X/I:X=X/J:16+1
36897602 37696282 f^an-x+iang=j@1_2/A:3_an_2+f/B:4-X=2-f@2-1&2-3#1-0|iang/C:2+i+2_n/D:4-2/E:4_2@1+2&X+X#X+X/F:4=2/G:2_1/H:4=2^4=1|X/I:X=X/J:16+10-4$
37696282 39023248 an^x-iang+j=i@2_1/A:3_an_2+f/B:4-X=2-f@2-1&2-3#0-2|iang/C:2+i+2_n/D:4-2/E:4_2@1+2&X+X#X+X/F:4=2/G:2_1/H:4=2^4=1|X/I:X=X/J:16+10-4$
39023248 39767144 x^iang-j+i=w@1_2/A:4_iang_2+f/B:2-X=2-n@1-2&3-2#2-0|i/C:1+an+2_n/D:4-2/E:4_2@2+1&X+X#X+X/F:X=X/G:2_1/H:4=2^4=1|X/I:X=X/J:16+10-4$
39767144 40605217 iang^j-i+w=an@2_1/A:4_iang_2+f/B:2-X=2-n@1-2&3-2#0-1|i/C:1+an+2_n/D:4-2/E:4_2@2+1&X+X#X+X/F:X=X/G:2_1/H:4=2^4=1|X/I:X=X/J:16+10-4$
40605217 41161649 j^i-w+an=SIL@1_2/A:2_i_2+n/B:1-X=2-n@2-1&4-1#1-0|an/C:X+X+X_X/D:4-2/E:4_2@2+1&X+X#X+X/F:X=X/G:2_1/H:4=2^4=1|X/I:X=X/J:16+10-4$
41161649 43194389 i^w-an+SIL=XX@2_1/A:2_i_2+n/B:1-X=2-n@2-1&4-1#0-6|an/C:X+X+X_X/D:4-2/E:4_2@2+1&X+X#X+X/F:X=X/G:2_1/H:4=2^4=1|X/I:X=X/J:16+10-4$
43194389 46695000 w^an-SIL+XX=XX@X_X/A:1_an_2+n/B:X-X=X-X@X-X&X-X#X-X|X/C:X+X+X_X/D:4-2/E:X_X@X+X&X+X#X+X/F:X=X/G:4_2/H:X=X^X=X|X/I:X=X/J:16+10-4$
```

前端文本分析结果：抄本

p1^p2-p3+p4=p5@p6_p7/A:a1_a2&a3_a4/B:b1_b2#b3_b4

b5_b6/C:c1+c2/D:d1_d2/E:e1+e2/F:f1_f2/G:g1+g2+g3/H:

p1: LL phoneme, 当前音素的左边的左边的音素
p2: L(eft) phoneme, 当前音素的左边的音素
p3: C(urrent) phoneme, 当前音素
p4: R(ight) phoneme, 当前音素右边的音素
p5: RR phoneme, 当前音素的右边的右边的音素
p6:从左往右数，当前音素在当前音节中的位置
p7:从右往左数，当前音素在当前音节中的位置
a1:当前音节的前一个音节是否要重读
a2:当前音节的前一个音节的音素数目
a3:从左往右数，当前音节在韵律词中的位置
a4:从右往左数，当前音节在韵律词中的位置
b1:当前音节是否要重读
b2:当前音节拥有的音素数目
b3:从左往右数，当前音节在当前词中的位置
b4:从右往左数，当前音节在当前词中的位置

b5:从上一个重读音节到当前音节之间的音节数目
b6:从当前音节到下一个重读音节之间的音节数目
c1:当前音节的下一个音节是否要重读
c2:下个音节拥有的音素数目
d1:当前词的前一个词的词性
d2:当前词的前一个词拥有的音节数目
e1:当前词的词性
e2:当前词拥有的音节数目
f1:当前词的下一个词的词性
f2:当前词的下一个词拥有的音节数目
g1:当前音节的前一个音节的声调
g2:当前音节的音调
g3:当前音节的下一个音节的声调

总结：音素、音节、词等的位置，词性、声调、重读、韵律等特征

目录

1. 语音合成基础与流程
2. 文本分析模块构成
3. 条件随机场(CRF)
4. 基于传统方法的前端文本分析模型
5. 基于神经网络的前端文本分析模型
6. 实战

基于NN的文本分析模块

g2p

- 基于LSTM的多音消歧
  - *2016. changhao Et.al. - A Bi-directional LSTM Approach for Polyphone Disambiguation in Mandarin Chinese*
- 基于seq-to-seq的g2p
  - *2019. Sevinj Et.al. - Transformer based Grapheme-to-Phoneme Conversion*

分词

- 基于BLSTM + CRF 的分词
  - *2016. chen Et.al. - Long Short-Term Memory Neural Networks for Chinese Word Segmentation*
  - Opensource（基于神经网络的CRF分词）：https://github.com/bojone/crf
- 基于BERT的分词
  - *2019 huang Et.al. - Toward Fast and Accurate Neural Chinese Word Segmentation with Multi-Criteria Learning*

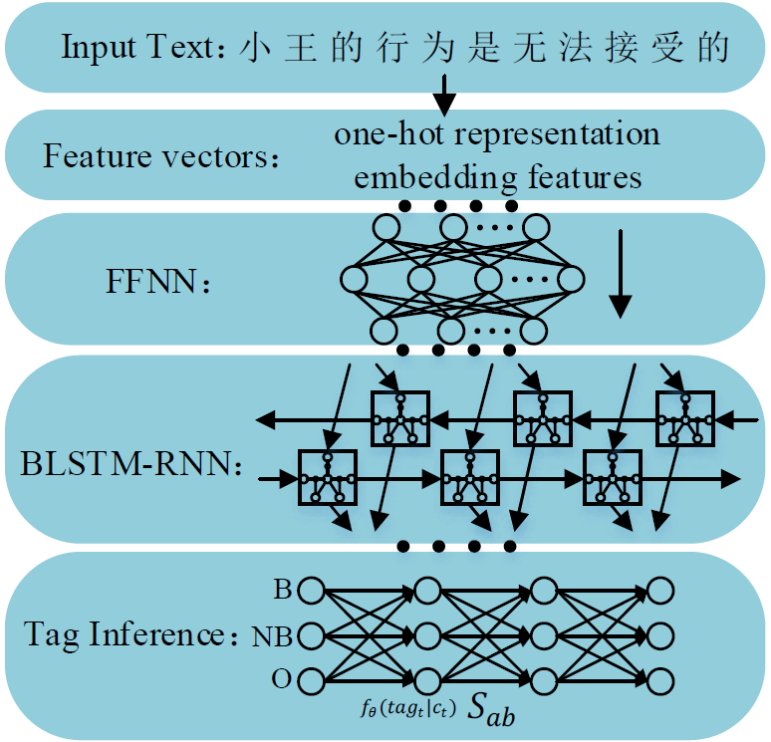
韵律

- 基于BLSTM + CRF 的韵律预测
  - *2015. Ding Et.al. - AUTOMATIC PROSODY PREDICTION FOR CHINESE SPEECH SYNTHESIS USING BLSTM-RNN AND EMBEDDING FEATURES*
- 基于BERT的韵律预测
  - *2020 Zhang Et.al. - Chinese Prosodic Structure Prediction Based on a Pretrained Language Representation Model*

基于NN的文本分析模块 – 韵律

动机

- CRF韵律预测依赖于分词和词性的准确性，分词错误不可避免地会严重影响韵律的准确程度
- 一些人为参与设计的特征模版(feature engineering)都是基于经验性的工程，特征选择的正确与否导致最后模型的好坏



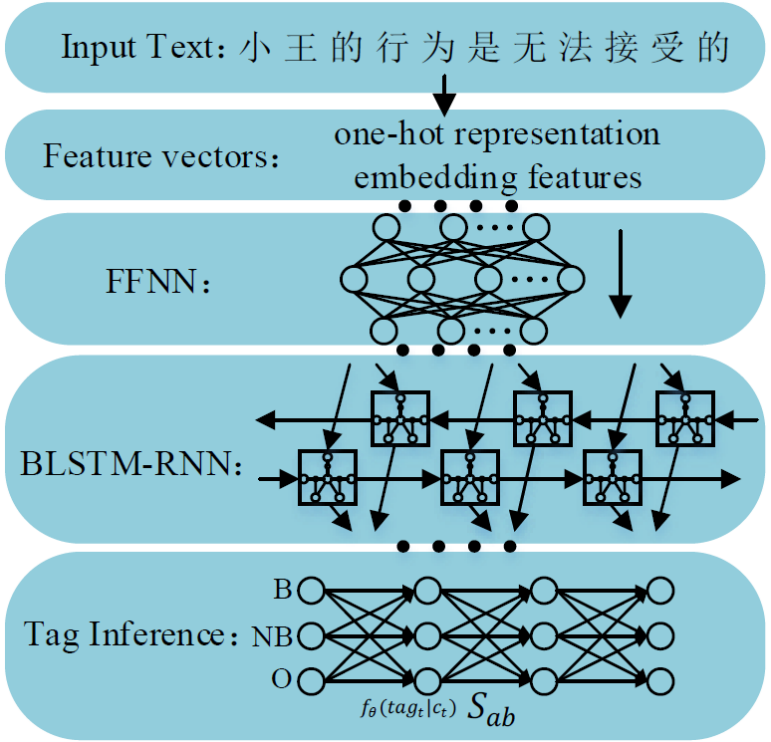
2015 .Ding Et.al. - AUTOMATIC PROSODY PREDICTION FOR CHINESE SPEECH SYNTHESIS USING BLSTM-RNN AND EMBEDDING FEATURES (上图来源)

基于NN的文本分析模块 – 韵律

BLSTM + CRF框架

- 输入：one-hot 特征 / word embedding
- 输出：输入文本中每个字对应的类别，一共3个类别 B/NB/O，分别代表边界、非边界、others
- 网络结构
  - FNN + BLSTM
  - CRF(考虑输出之间的联系)

等级结构		标记	模型
韵律词	prosody word PW	**L1(#1)**	B/NB/O
韵律短语	prosody phrase PPH	**L2(#2)**	B/NB/O
语调短语	intonational phrase IPH	**L3(#3)**	B/NB/O



2015 .Ding Et.al. - AUTOMATIC PROSODY PREDICTION FOR CHINESE SPEECH SYNTHESIS USING BLSTM-RNN AND EMBEDDING FEATURES (上图来源)

基于NN的文本分析模块 – 韵律

输入层

◘ one-hot 特征

v("说")= [0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 0 ...] ∈ $\mathbb{R}^N$

v("话")= [0 0 0 0 0 0 0 0 **1** 0 0 0 0 0 0 0 ...] ∈ $\mathbb{R}^N$

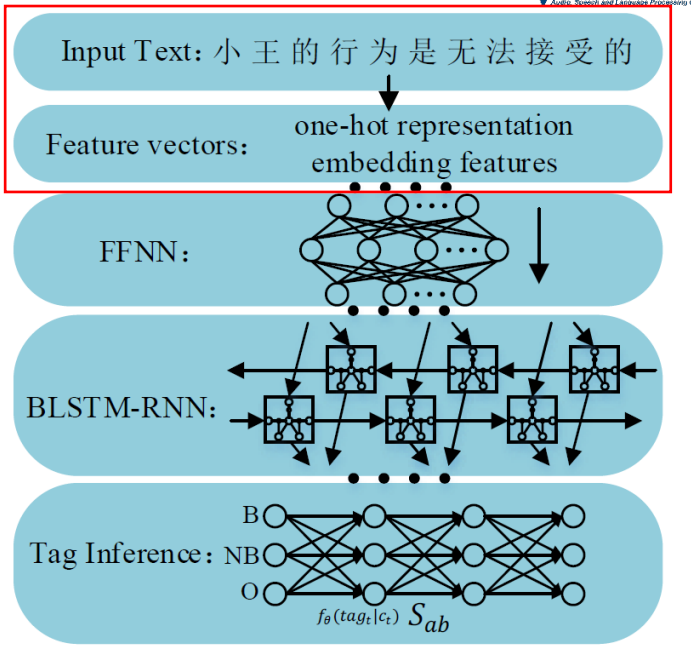
其中，N为字典$\mathcal{D}$的大小，即字典中一共包含N个字。

◘ distributed representation or embedding feature

思想：通过训练，将词典中的每个词映射成一个固定长度的低维向量。所有这些向量构成一个词向量空间，每一个词向量都是该空间的一个点。在词向量空间中引入距离，便于度量词之间的相似性。

代表方法：Word2Vec

示例：某词向量为[0.792, −0.177, −0.107, 0.109, −0.542, ...]



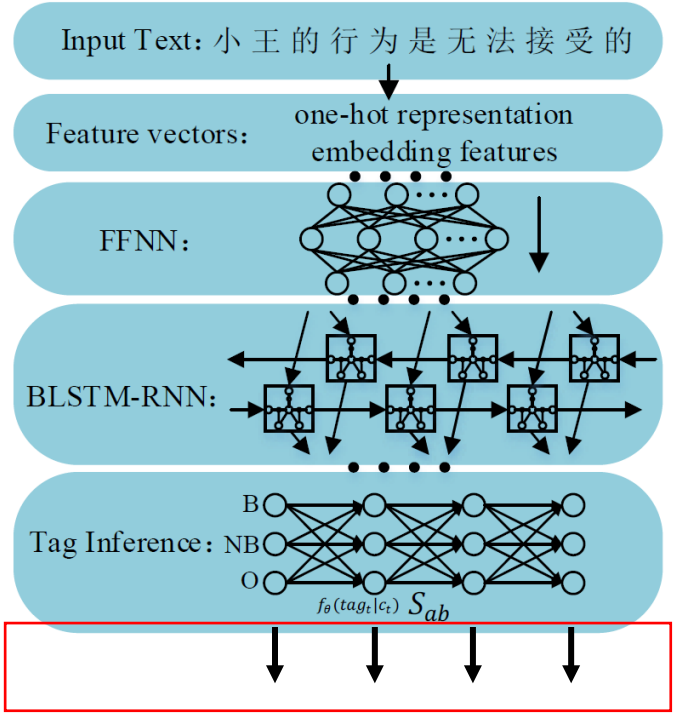
参考论文：*AUTOMATIC PROSODY PREDICTION FOR CHINESE SPEECH SYNTHESIS USING BLSTM-RNN AND EMBEDDING FEATURES*， *2015 .Ding Et.al.*

参考资料：*《word2vec数学原理》*

基于NN的文本分析模块 – 韵律

输出层

输入文本中每个字对应的类别，一共有3种类别，即B/NB/O，分别代表边界、非边界、others。



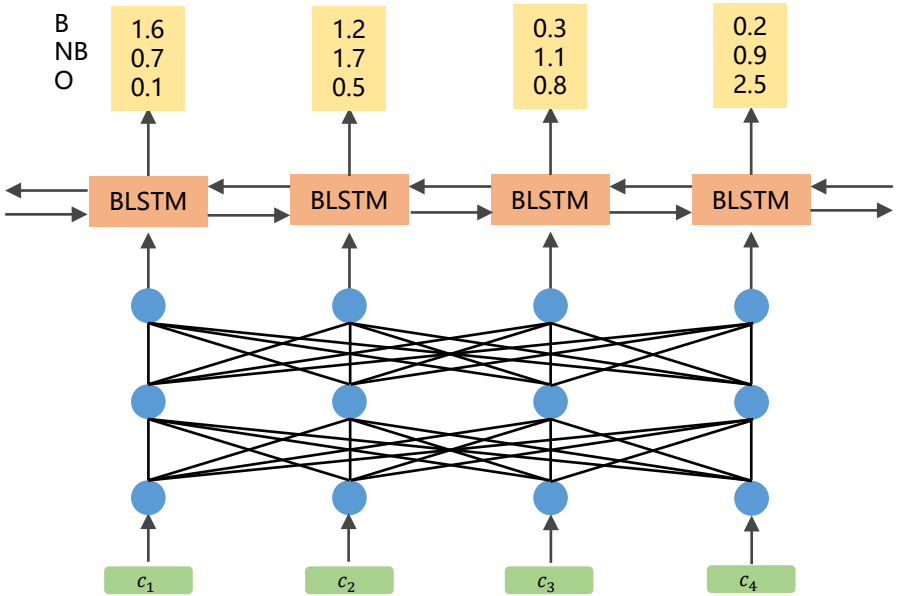
参考论文：*AUTOMATIC PROSODY PREDICTION FOR CHINESE SPEECH SYNTHESIS USING BLSTM-RNN AND EMBEDDING FEATURES，2015 .Ding Et.al.*
参考资料：*《word2vec数学原理》*

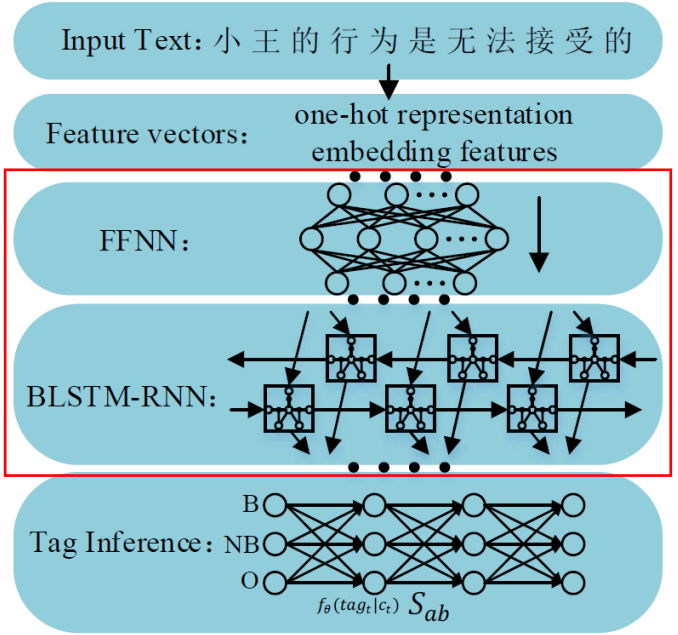
基于NN的文本分析模块 – 韵律

FNN+BLSTM层

BLSTM层的输入：每个字的向量表示

BLSTM层的输出：当前时刻的输入c_t属于每个类别标签的概率





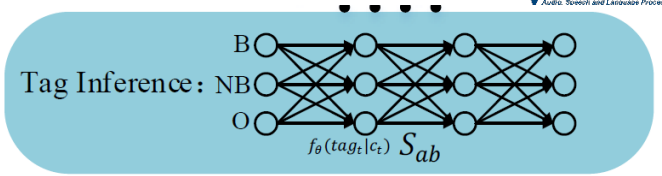
参考论文：*AUTOMATIC PROSODY PREDICTION FOR CHINESE SPEECH SYNTHESIS USING BLSTM-RNN AND EMBEDDING FEATURES*， *2015 .Ding Et.al.*

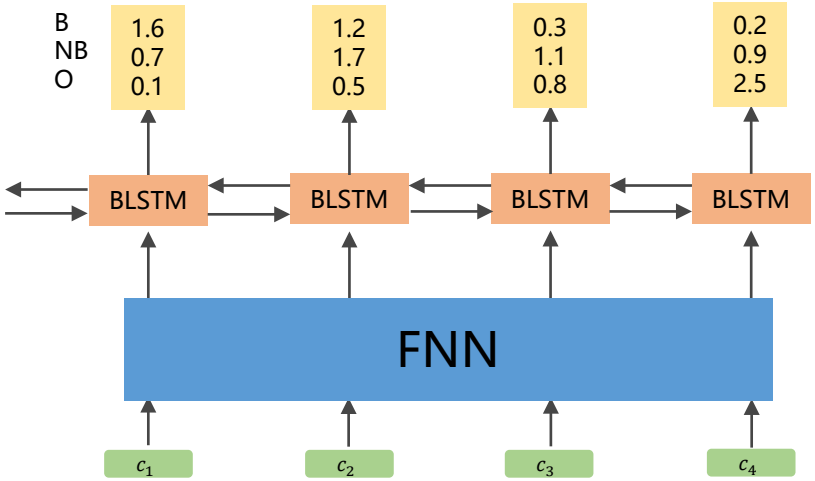
参考资料：*《word2vec数学原理》*

基于NN的文本分析模块 – 韵律

CRF层

为什么需要CRF层？直接用BLSTM的输出作为预测结果不可以吗？





CRF层可以加入一些约束来保证最终预测结果是有效的。这些约束可以在训练数据时被CRF层自动学习得到。

可能的约束条件有：

① 句子的开头应该是“B”或“NB”，而不是“O”

② 对于韵律词预测，韵律词都是在词级别，考虑到词长一般是有限制的，所以不会经常连续出现多个“NB”，即“NB-label，NB-label，NB-label，NB-label，NB-label，NB-label…”很少出现或者是不合理的；因为是在词级别，所以也很少出现“B-label，B-label，B-label， B-label…”这种情况。

③ 对于韵律短语，一个句子，连续出现多个韵律词是不遵循语法规则的(考虑到人读句子是需要换气的)。所以下面这种情况也是不合理的:

中华/B人民/B共和国/B中央/B人民/B政府/B今天/B成立了/B

(省略的字后面标注为NB)，对应标注如下：

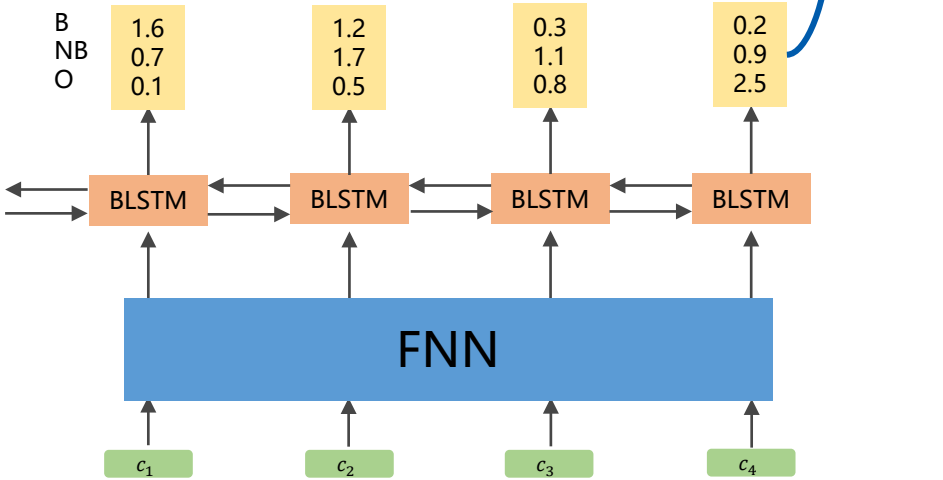
中华#2人民#2共和国#2中央#2人民#2政府#2今天#2成立了#2（不合理）

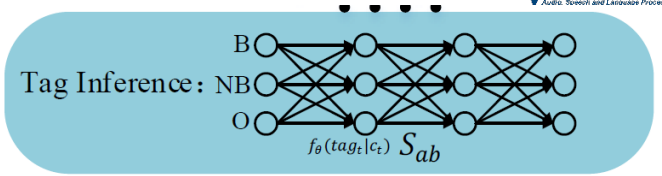
中华人民共和国#2中央人民政府#2今天成立了#2（合理）

基于NN的文本分析模块 – 韵律

CRF层-network score

$$P_w(y|\mathbf{x}) = P(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} exp\Big(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, \mathbf{x}, i) + \sum_{i,l} \mu_l s_l(y_i, \mathbf{x}, i)\Big)$$



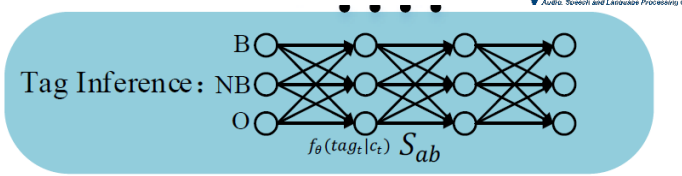


举例：c_1被标记为B的分数为1.6， c_2被标记为NB的分数为1.7 。

基于NN的文本分析模块 – 韵律

CRF层-transition score

$$P_w(y|\mathbf{x}) = P(y|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} exp\Big(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, \mathbf{x}, i) + \sum_{i,l} \mu_l s_l(y_i, \mathbf{x}, i) \Big)$$



下一个字 / 当前字	B	NB	O
B	0.3	0.6	0.1
NB	0.4	0.5	0.1
O	0.3	0.5	0.2

基于NN的文本分析模块 – 韵律

CRF路径得分

对于5个字组成的句子，其可能的类别序列为：

1) B B B B B

2) B NB B B

3) B B NB B

.

.

.

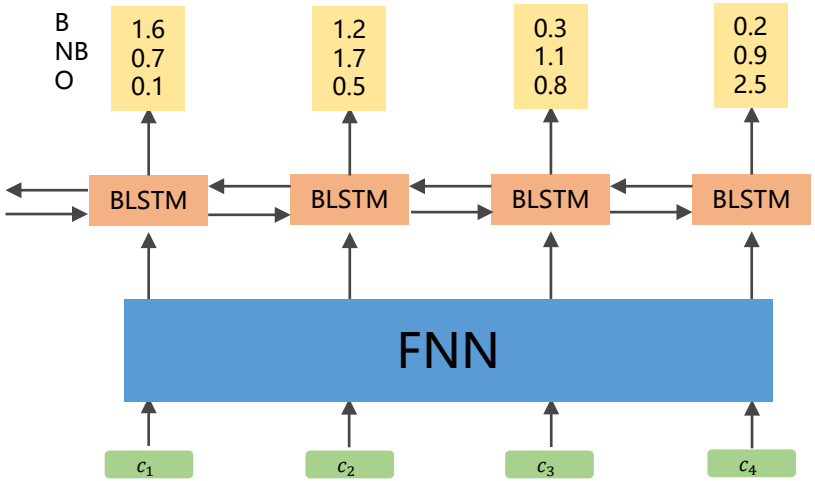
N) NB NB NB

$$P_{total} = P_1 + P_2 + \cdots P_N = e^{s_1} + e^{s_2} + \cdots + e^{s_N}$$

$$s_i = network\ score + transition\ score$$

$$LossFunction = -log\frac{P_{RealPath}}{P_1 + P_2 + \ldots P_N}$$

思考：为什么损失函数不直接定义为$-logP_{RealPath}$？



当前字 \ 下一个字	B	NB	O
B	0.3	0.6	0.1
NB	0.4	0.5	0.1
O	0.3	0.5	0.2

基于NN的文本分析模块 – 韵律

BLSTM + CRF(实验结果)

Boundary	P (%)	R (%)	F (%)
PW	95.34	96.73	96.03
PPH	83.41	83.68	83.06
IPH	84.85	73.39	78.71

Table 1. The results of CRF-based prosody prediction.

Boundary	P (%)	R (%)	F (%)	TP / Num of nodes
PW	96.02	96.69	96.35	FBB / 32
PPH	82.50	86.75	84.57	FBB / 128
IPH	84.06	79.33	81.63	FBB / 64

Table 3. The best performance of each level and the corresponding network topology (TP).

BLSTM + CRF(优点)

- 抛弃繁琐的特征模版，自动学习词的embedding；
- 采用LSTM能够获取更多的上下文信息；
- 结合CRF，同时考虑输出之间的依赖关系；

延伸：不仅仅是韵律，TTS前端文本分析模块分词、词性、多音字，更是依赖更长的上下文，所以神经网络(LSTM/GRU)对TTS前端文本分析整个性能都有很大的提升。

本章总结

- 文本分析的基本组成
  - TN/分词/词性/g2p/韵律

- 文本分析各个模块的方法
  - TN：基于规则的方法
  - 分词：字典/CRF/BLSTM+CRF/BERT
  - 注音：ngram/CRF/BLSTM/seq2seq
  - 韵律：CRF/BLSTM+CRF/BERT
  - Etc.

本章总结

Every time I fire a linguist, the performance of our speech recognition system goes up

By Frederick Jelinek

目录

1. 语音合成基础与流程

2. 文本分析模块构成

3. 条件随机场(CRF)

4. 基于传统方法的前端文本分析模型

5. 基于神经网络的前端文本分析模型

6. 实战

实战1：基于CRF++的中文分词

尝试按照README.md中的步骤，利用CRF++实现中文分词

必做题

在给出的10万行训练集上训练模型，对给定测试集进行分词，并计算准确率、召回率和F值。

选做题

（1）我们同时给出了完整的26万行的数据集，感兴趣的可以按照同样的步骤进行训练，观察结果是否较10万行数据集有所提升。

（2）可以尝试自己设计模板，观察测试集在分词各项指标上是否有提升。

Repo：

https://github.com/nwpuaslp/TTS_Course

实践2：基于ngram/rnnlm的g2p模型

尝试按照README.md中的步骤，利用phonetisaurus进行g2p模型训练

- ngram模型
  - 根据readme熟悉英文pipeline
  - 参考英文pipeline，构建自己的中文g2p pipeline，使用给出的训练集训练模型，在给定测试集上进行解码，并计算准确率。（必做）
- rnnlm模型
  - 参考readme中训练测试过程，使用给出的训练集训练模型，在给定测试集上进行解码，并计算准确率，观察rnnlm和ngrm的模型效果对比。（选做）

Repo：

https://github.com/nwpuaslp/TTS_Course

深蓝学院
shenlanxueyuan.com

音频语音与语言处理研究组
SLP
Audio, Speech and Language Processing Group NPU

感谢聆听！

Thanks for Listening