Sequence Labeling for Part of Speech and Named Entities

Part of Speech Tagging

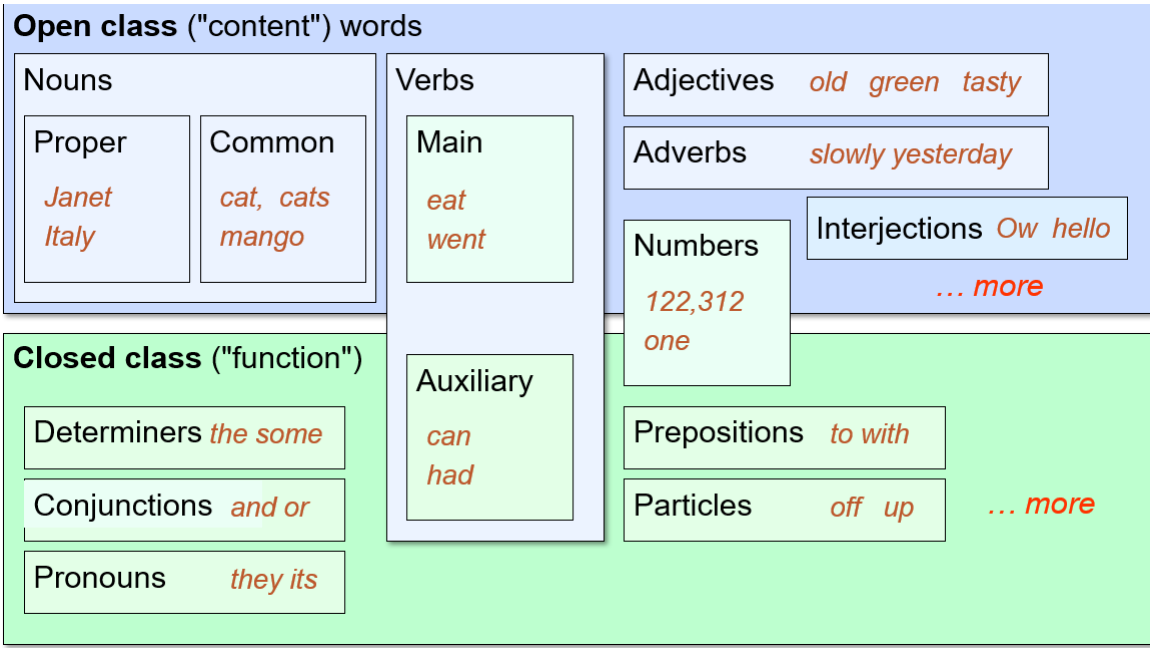
part of Speech

```
noun名词
verb动词
pronoun代词
preposition介词
adverb副词
conjunction连接词
participle分词
article冠词
```

Two classes of words: Open vs. Closed

```
1.Closed class words
Relatively fixed membership
Usually function words: short, frequent words with grammatical function
. determiners: a, an, the
. pronouns: she, he, I
. prepositions: on, under, over, near, by, …

2.Open class words
.Usually content words: Nouns, Verbs, Adjectives, Adverbs(名词、动词、形容词、副词)
.Plus interjections(感叹词): oh, ouch, uh-huh, yes, hello
.New nouns and verbs（新名词和动词） like iPhone or to fax
```



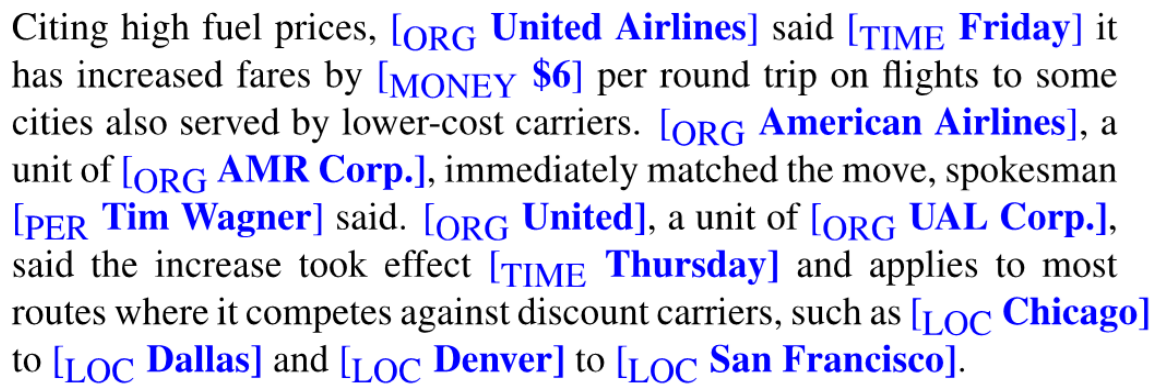
Why Part of Speech Tagging ?

```
Parsing: POS tagging can improve syntactic parsing
MT: reordering of adjectives and nouns (say from Spanish to English)
Sentiment or affective tasks: may want to distinguish adjectives or other POS
Text-to-speech: (how do we pronounce “lead” or "object"?)
```

Named Entity Recognition

```
1. Named Entity:means anything that can be referred to with a proper name\
Most common 4 tags:
PER (Person): “Marie Curie”
LOC (Location): “New York City”
ORG (Organization): “Stanford University”
GPE (Geo-Political Entity): "Boulder, Colorado"
```

NER OUTPUT



why NER

```
Sentiment analysis: consumer’s sentiment toward a particular company or person?
Question Answering: answer questions about an entity?
Information Extraction: Extracting facts about entities from text.
```

Why Hard

```
1.Segmentation
    In POS tagging, no segmentation problem since each word gets one tag.
    In NER we have to find and segment the entities!
2.Type ambiguity
```

Standard algorithms for POS tagging and NER

```
Supervised Machine Learning Algorithms:

Via human created features
.Hidden Markov Models(HMM)
.Conditional Random Fields (CRF)/ Maximum Entropy Markov Models (MEMM)

Via representation learning:  Neural LMs
.Neural sequence models (RNNs or Transformers)
.Large Language Models (like BERT), finetuned
```

HMM

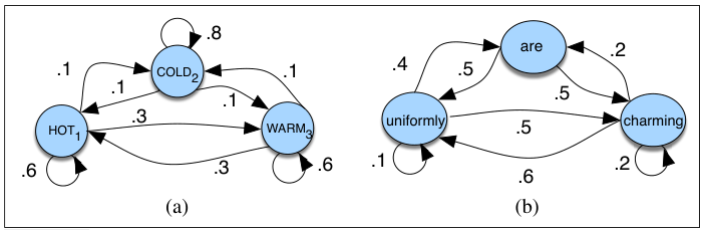
Markov Chains

```
Markov assumption 1
when predicting the future, the past doesn't matter, only the present.

P(qi = a|q1...qi−1) = P(qi = a|qi−1)


a Markov chain is specified by the following components:

Q = q1q2 ...qN            a set of N states
A = a11a12 ...aN1 ...aNN  a transition probability matrix A
π = π1,π2,...,πN          an initial probability distribution over states.
```



The Hidden Markov Model

```
Q = q1q2 ...qN          N个状态
A = a11 ...ai j ...aNN 状态转移函数
O = o1o2 ...oT           可能的观测集合
B = bi(ot)              观测概率矩阵 （代表t时刻处于
 状态qi生成观测ot的概率[也叫生成概率和发射概率])
π = π1,π2,...,πN       初始状态概率向量.


Markov Assumption 2
the probability of an output observation oi depends only on the state that
produced the observation qi and not on any other states or any other
observations

Output Independence: P(oi|q1,...qi,...,qT,o1,...,oi,...,oT) = P(oi|qi)
```

The Viterbi Algorithm

function VITERBI(*observations* of len T,*state-graph* of len N) **returns** *best-path*, *path-prob*

create a path probability matrix *viterbi[N,T]*
for each state s **from** 1 **to** N **do** ; initialization step
  $viterbi[s,1] \leftarrow \pi_s * b_s(o_1)$
  $backpointer[s,1] \leftarrow 0$
for each time step t **from** 2 **to** T **do** ; recursion step
  for each state s **from** 1 **to** N **do**
    $viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
    $backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$
$bestpathprob \leftarrow \max_{s=1}^{N} viterbi[s,T]$; termination step
$bestpathpointer \leftarrow \operatorname{argmax}_{s=1}^{N} viterbi[s,T]$; termination step
$bestpath \leftarrow$ the path starting at state *bestpathpointer*, that follows backpointer[] to states back in time
return *bestpath*, *bestpathprob*

An Example

Input

A 状态转移函数， 初始状态概率向量

	NNP	MD	VB	JJ	NN	RB	DT
<s>	0.2767	0.0006	0.0031	0.0453	0.0449	0.0510	0.2026
NNP	0.3777	0.0110	0.0009	0.0084	0.0584	0.0090	0.0025
MD	0.0008	0.0002	0.7968	0.0005	0.0008	0.1698	0.0041
VB	0.0322	0.0005	0.0050	0.0837	0.0615	0.0514	0.2231
JJ	0.0366	0.0004	0.0001	0.0733	0.4509	0.0036	0.0036
NN	0.0096	0.0176	0.0014	0.0086	0.1216	0.0177	0.0068
RB	0.0068	0.0102	0.1011	0.1012	0.0120	0.0728	0.0479
DT	0.1147	0.0021	0.0002	0.2157	0.4744	0.0102	0.0017

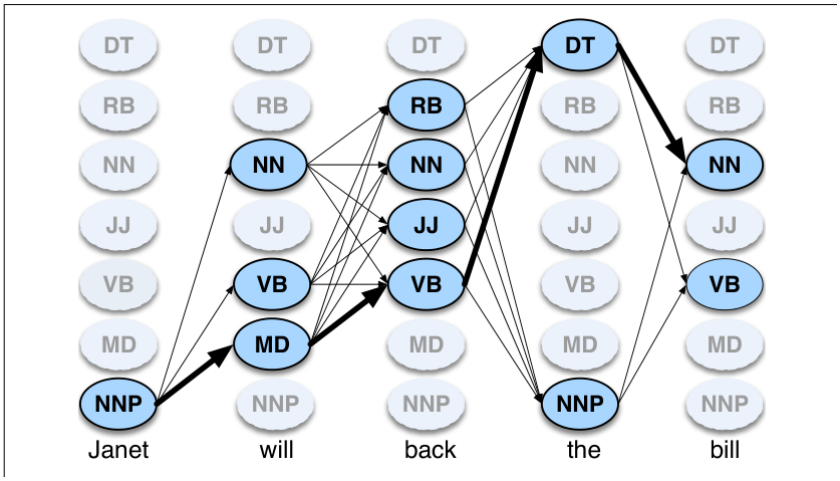
Figure 8.12 The A transition probabilities $P(t_i|t_{i-1})$ computed from the WSJ corpus without smoothing. Rows are labeled with the conditioning event; thus $P(VB|MD)$ is 0.7968.

B 观测概率函数

	Janet	will	back	the	bill
NNP	0.000032	0	0	0.000048	0
MD	0	0.308431	0	0	0
VB	0	0.000028	0.000672	0	0.000028
JJ	0	0	0.000340	0	0
NN	0	0.000200	0.000223	0	0.002337
RB	0	0	0.010446	0	0
DT	0	0	0	0.506099	0

Figure 8.13 Observation likelihoods B computed from the WSJ corpus without smoothing, simplified slightly.

Solution



for each time step t **from** 2 **to** T **do** ; recursion step

  for each state s **from** 1 **to** N **do**

$$viterbi[s,t] \leftarrow \max_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$$

$$backpointer[s,t] \leftarrow \operatorname{argmax}_{s'=1}^{N} viterbi[s',t-1] * a_{s',s} * b_s(o_t)$$

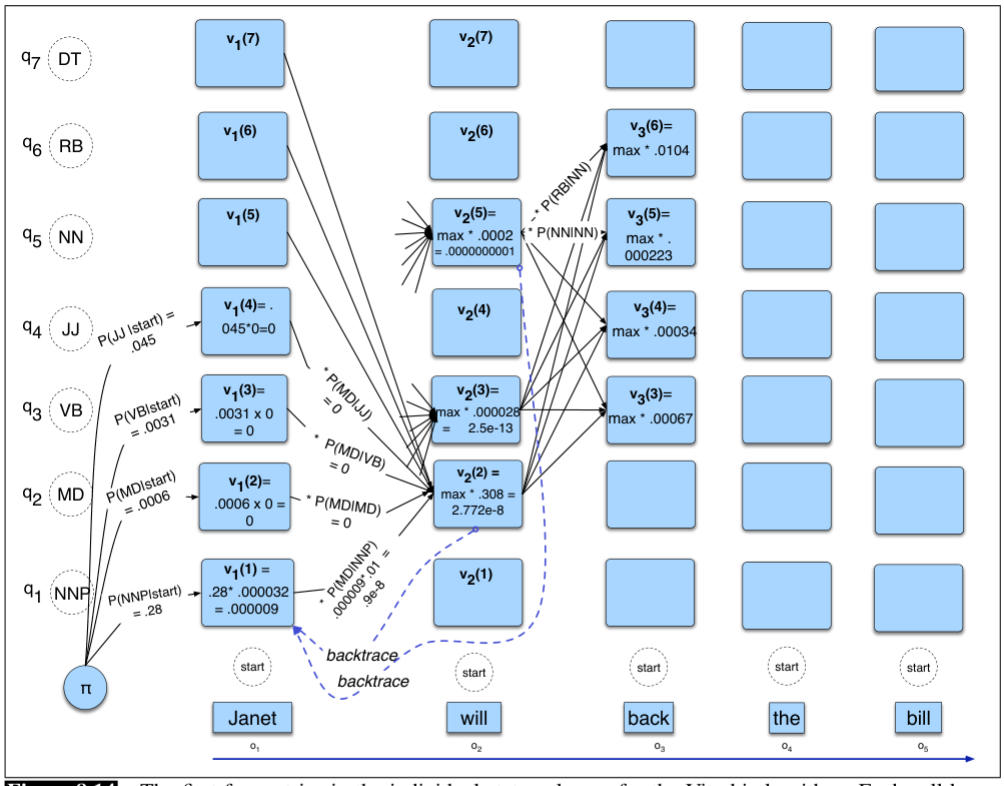


Figure 8.14 The first few entries in the individual state columns for the Viterbi algorithm. Each cell keeps the probability of the best path so far and a pointer to the previous cell along that path. We have only filled out columns 1 and 2; to avoid clutter most cells with value 0 are left empty. The rest is left as an exercise for the reader. After the cells are filled in, backtracing from the *end* state, we should be able to reconstruct the correct state sequence NNP MD VB DT NN.

CRFs : Conditional Random Fields

条件随机场(Conditional Random Fields, 以下简称CRF)是给定一组输入序列,输出序列的条件概率分布模型，即CRF擅长解决相邻上下文相关的问题。

随机场：按照某种分布给一系列变量进行赋值，这些变量构成的整体就叫做随机场。（是由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之后，其全体就叫做随机场）。标注序列 ”OOBSOBMSOOO”就是一个随机场

马尔科夫随机场：满足马尔可夫性的随机场，是随机场的特例，它假设随机场中某一个位置的赋值仅仅与和它相邻的位置的赋值有关，和与其不相邻的位置的赋值无关。

CRF：是马尔科夫随机场的特例，它假设马尔科夫随机场中只有X和Y两种变量，X一般是给定的，而Y一般是在给定X的条件下我们的输出。

CRF中的特征函数

怎么判断一个标注序列靠谱不靠谱呢？
我们可以定义一个特征函数集合，用这个特征函数集合来为一个标注序列打分，并据此选出最靠谱的标注序列。

正式定义什么是CRF的特征函数。他接受四个参数

句子s（就是我们要标注词性的句子）
i，用来表示句子s中第i个单词
l_i，表示要评分的标注序列给第i个单词标注的词性
l_i-1，表示要评分的标注序列给第i-1个单词标注的词性

输出：0或者1，表示要评分的标注序列不符合这个特征，1表示要评分的标注序列符合这个特征。

定义好一组特征函数后，我们要给每个特征函数f_j赋予一个权重λ_j。现在，只要有一个句子s，有一个标注序列l，我们就可以利用前面定义的特征函数集来对l评分。

$$score(l|s) = \sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s, i, l_i, l_{i-1})$$

对这个分数进行**指数化和标准化**，我们就可以得到标注序列l的概率值**p(l|s)**，如下所示

$$p(l|s) = \frac{exp[score(l|s)]}{\sum_{l'} exp[score(l'|s)]} = \frac{exp[\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s,i,l_i,l_{i-1})]}{\sum_{l'} exp[\sum_{j=1}^{m} \sum_{i=1}^{n} \lambda_j f_j(s,i,l'_i,l'_{i-1})]}$$

简单来说

为了建一个条件随机场，首先要定义一个特征函数集，每个特征函数都以整个句子s，当前位置i，位置i和i-1的标签和为输入。然后为每一个特征函数赋予一个权重，然后针对每一个标注序列Y，对所有的特征函数加权求和，必要的话，可以把求和的值转化为一个概率值

Inference and Training for CRFs

the same supervised learning algorithms we presented for logistic regression.

CRF与HMM，逻辑回归的区别

1. 事实上，CRF是逻辑回归的序列化版本。逻辑回归是用于分类的对数线性模型，条件随机场是用于序列化标注的对数线性模型。

2. CRF—判别式模型：直接对$P(Y|X)$建模；

   HMM—生成式模型：训练阶段对$P(X,Y)$建模，inference再对新的sample计算$P(Y|X)$。