

# L101: Machine Learning for Language Processing

## Lecture 2

Guy Emerson & Ted Briscoe

# Today's Lecture

---

- Recap
- Part-of-speech tagging
- Hidden Markov Model (HMM)

# Recap – Models

$$f : x \mapsto y$$

- Non-probabilistic:  $f$
- Discriminative:  $P(y|x)$
- Generative:  $P(x, y)$

2

Recall:

A non-probabilistic model deterministically maps inputs to outputs.

Probabilistic models can be broken down into two types. Discriminative models model the conditional distribution over outputs, given an input. Generative models model the joint probability of inputs and outputs. Generative models are more general, since  $P(x, y) = P(y|x)P(x)$ , so they also tell us the probability of inputs  $P(x)$  – but this can also make them more challenging.

# Recap – Naive Bayes

- Fixed vocabulary
- Feature vectors  $x_i$ 
  - Binary (Bernoulli NB)
  - Bag of words (Multinomial NB)
- Parameters  $P(x_i|y)$ ,  $P(y)$
- Train using observed counts

# Vocabulary Size

---

**Zipf's Law** : word frequency follows a power law distribution; many words only appear once (**long tail**)

**Heaps' Law** : no matter how much data we observe (**tokens**), we will never see all words (**types**)

⇒ Some words unseen at test time

# Part-of-Speech Tagging

They can fish .

PNP VM0 VVI PUN

PNP VVB NN2 PUN

PNP VM0 NN2 PUN no full parse!

CLAWS-5 tagset includes:

NN1	singular noun	VVB	verb, base form
NN2	plural noun	VVI	verb, infinitive
PNP	personal pronoun	VM0	verb, modal
PUN	punctuation	VVZ	verb, 3rd pers. sg.

# Part-of-Speech Lexicon Fragment

---

they PNP

can VM0, NN1, VVB, VVI

fish NN2, NN1, VVB, VVI

- Could be hand-written
- ML: aim to learn from data

# Why Do Part-of-Speech Tagging?

- Not often considered until 1990s
- Easier than full parsing
- For applications:
  - Reduce search space for unknown words
  - Input features for other tasks
- For linguistics:
  - Lexicography
  - Corpus linguistics



# Defining The Task

---

- Which language? (dialect?)
- Tagset? Syntactic analysis?
- Genre? Domain?

Nearly all published work is on a limited range of standard datasets, which are relatively small and confined to certain genres (such as newswire text), and certain domains (such as finance). Performance for a real task may not correlate well with performance on standard datasets.

# Defining The Task

---

- Errors in the input?  
They walked into into the room
- Errors in the annotations?
- Rare words / rare usages of words?

Datasets are sometimes inconsistent – what should we expect from a Machine Learning system trained on such data?

# Defining The Task

---

- Sequence labelling
  - Input: sequence
  - Output: sequence of same length
- Usually supervised

# Data



- Limited training data
  - Requires trained annotators
  - Annotation guidelines are lengthy
- High inter-annotator agreement

# Hidden Markov Model

$$\begin{aligned} & \operatorname{argmax}_{t_1 \cdots t_n} P(t_1 \cdots t_n | w_1 \cdots w_n) \\ &= \operatorname{argmax}_{t_1 \cdots t_n} P(t_1 \cdots t_n) P(w_1 \cdots w_n | t_1 \cdots t_n) \\ &\approx \operatorname{argmax}_{t_1 \cdots t_n} \prod_{i=1}^n P(t_i | t_{i-1}) P(w_i | t_i) \end{aligned}$$

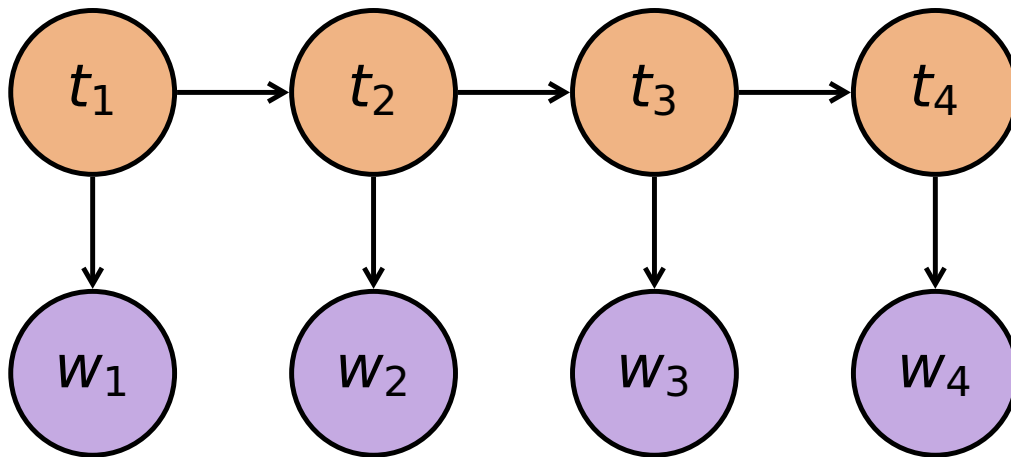
12

This derivation should be reminiscent of Naive Bayes, from last lecture! The term “Markov” is essentially the mathematical way of saying “forgetful”.

For the equation to be well-defined, we also need to define a tag  $t_0$ . If this is a special start-of-sequence tag, this is equivalent to defining a starting distribution for  $t_1$ . It is also common to define a special end-of-sequence tag. The start- and end-of-sequence tags are important, because sentences have clear starts and ends, and the distribution over tags is not constant through a sentence (compare sentences of length  $N$  to sequences of length  $N$  taken out of a longer text).

The equations are for a *bigram* HMM (there is a dependence for sequences of length 2) – each tag  $t_i$  depends on the previous tag  $t_{i-1}$ . We can relax the Markov independence assumption to consider *trigram* HMM (there is a dependence for sequences of length 3) – each tag  $t_i$  depends on the previous tags  $t_{i-1}$  and  $t_{i-2}$ . We can also consider higher  $n$ -grams, but it becomes increasingly difficult to train the parameters.

# Hidden Markov Model



# Hidden Markov Model

- Parameters:
  - $P(t_i|t_{i-1})$  – transition probabilities
  - $P(w_i|t_i)$  – emission probabilities
- Train using observed counts
- Smoothing hyperparameters

Again, this should be reminiscent of Naive Bayes.

# Backoff

For higher  $n$ -grams, can use backoff:

$$\hat{P}(t_i|t_{i-1}, t_{i-2}) = \lambda P_{\text{trigram}}(t_i|t_{i-1}, t_{i-2}) + (1 - \lambda) P_{\text{bigram}}(t_i|t_{i-1})$$

$P_{\text{trigram}}, P_{\text{bigram}}$  calculated using counts



# Inference

- An HMM learns  $P(t_1 \cdots t_n, w_1 \cdots w_n)$
- Dynamic programming:
  - Most likely sequence  
→ Viterbi Algorithm
  - Most likely tag for each word  
→ Forward-Backward Algorithm

# Unknown Words



- Frequency of open-class tags
- Morphology (e.g. “-ing”)
- Capitalisation (e.g. “Bill” vs. “bill”)

# Discriminative POS-Tagging



- Conditional Random Fields
- Recurrent Neural Networks (details in future lecture)

# State of the Art



- Plank et al. (2016), in course readings
- Performance close to ceiling
- Return to question: what is the task?