COMP90042 LECTURE 7

SEQUENCE TAGGING: UNSUPERVISED HMMS

HMMS FOR POS TAGGING - RECAP

- Previous lecture: **supervised** POS tagging using HMMs
  - Assume we have a corpus with annotated POS tags (for instance, Penn Treebank)
  - Learning via MLE (counting)
  - At test/prediction time, uses Viterbi algorithm to find most probably sequence
- Sometimes we do not have such a corpus
  - Different domains (Twitter, TED talks)
  - Different languages, especially low-resource ones
- Solution: **unsupervised** learning

UNSUPERVISED HMMS

- Suppose you have a corpus of (unannotated) tweets.
- Simple idea: start with an HMM with **random** parameters (emission and transition matrices)
  - Using Viterbi, tag the data using this model.
  - Pretend this is training data. Obtain counts via MLE.
  - Repeat
- Rationale: the model will learn the actual POS distribution based on the word patterns seen in the data
  - This is a simple version of the **Expectation-Maximisation (EM)** algorithm

EXPECTATION MAXIMIZATION

- Janet/NNP will/**NNP** back/**RB** the/DT bill/NN
- $$P(will|\boldsymbol{t}) = \begin{bmatrix} P(will|NNP) \\ P(will|RB) \\ P(will|DT) \\ \vdots \end{bmatrix} = \begin{bmatrix} \frac{c(will,NNP)}{c(NNP)} \\ \frac{c(will,RB)}{c(RB)} \\ \frac{c(will,DT)}{c(DT)} \\ \vdots \end{bmatrix} = \begin{bmatrix} {}^{1}/_{2} \\ {}^{0}/_{1} \\ {}^{0}/_{1} \\ \vdots \end{bmatrix}$$

EXPECTATION MAXIMIZATION

- This approach, sometimes referred as **hard EM**, can perform well depending on the setting.
- However, it is still too naïve, as it does not take into account **distributions** on each word and each tag transition.
- A better approach would be to incorporate such distributions by:
  - Obtaining **marginal** emission and transition distributions.
  - Using **weighted** (expected) counts to train via MLE.

VITERBI EXAMPLE REVISITED

	Janet	will	back	the	bill
NNP	8.8544e-06	0	0	2.5e-17	0
MD	0	3.004e-8	0	0	0
VB	0	2.231e-13	1.6e-11	0	1.0e-20
JJ	0	0	5.1e-15	5.2e-16	0
NN	0	1.034e-10	5.4e-15	5.9e-18	**2.0e-15**
RB	0	0	5.3e-11	0	0
DT	0	0	0	1.8e-12	0

VITERBI EXAMPLE REVISITED

	Janet	will	back	the	bill
NNP	8.8544e-06	0	0	2.5e-17	0
MD	0	3.004e-8	0	0	0
VB	0	2.231e-13	1.6e-11	0	1.0e-20
JJ	0	0	5.1e-15	5.2e-16	0
NN	0	1.034e-10	5.4e-15	5.9e-18	**2.0e-15**
RB	0	0	**5.3e-11**	0	0
DT	0	0	0	1.8e-12	0

MLE WITH EXPECTED COUNTS

- $P(will|\boldsymbol{t}) = \begin{bmatrix} P(will|NNP) \\ P(will|RB) \\ P(will|DT) \\ \vdots \end{bmatrix} = \begin{bmatrix} \frac{\hat{c}(will,NNP)}{\hat{c}(NNP)} \\ \frac{\hat{c}(will,RB)}{\hat{c}(RB)} \\ \frac{\hat{c}(will,DT)}{\hat{c}(DT)} \\ \vdots \end{bmatrix} = ?$
- $\hat{c}(will, NNP) = \sum_{\boldsymbol{w}} \sum_{i=1;i=will}^{l\boldsymbol{w}} \hat{P}(t_i = NNP|\boldsymbol{w})$
  - "i = will" means we iterate only when the observed word is "will"
- $\hat{c}(NNP) = \sum_{\boldsymbol{w}} \sum_{i=1}^{l\boldsymbol{w}} \hat{P}(t_i = NNP|\boldsymbol{w})$
- $\hat{P}(t_i = NNP|\boldsymbol{w}) = \frac{\hat{P}(t_i=NNP,\boldsymbol{w})}{\hat{P}(\boldsymbol{w})}$

MLE WITH EXPECTED COUNTS

- Key quantities for emission probabilities:
  - $\hat{P}(\boldsymbol{w})$
  - $\hat{P}(t_i = NNP, \boldsymbol{w})$
- Let's start with $\hat{P}(\boldsymbol{w})$
  - $\hat{P}(\boldsymbol{w}) = \sum_{\boldsymbol{t}} \hat{P}(\boldsymbol{w}, \boldsymbol{t}) = \sum_{\boldsymbol{t}} \hat{P}(\boldsymbol{w}|\boldsymbol{t}) \hat{P}(\boldsymbol{t})$
  - Exponential number of tag sequences: can we use DP again?

THE FORWARD ALGORITHM

- Exactly like Viterbi, but summing scores instead of taking the max.
  - Also no backpointers since the goal is not prediction.

THE FORWARD ALGORITHM

	Janet	**will**	**back**	**the**	**bill**
NNP					
MD					
VB					
JJ					
NN					
RB					
DT					

THE FORWARD ALGORITHM

	Janet	will	back	the	bill
NNP	P(Janet|NNP) * P(NNP|<s>)				
MD	P(Janet|MD) * P(MD|<s>)				
VB	…				
JJ	…				
NN	…				
RB	…				
DT	…				

THE FORWARD ALGORITHM

	Janet	will	back	the	bill
NNP	8.8544e-06				
MD	0				
VB	0				
JJ	0				
NN	0				
RB	0				
DT	0				

THE FORWARD ALGORITHM

	Janet	**will**	**back**	**the**	**bill**
NNP	8.8544e-06	P(will|NNP) * P(NNP|t_{Janet}) * a(t_{Janet}|Janet)			
MD	0	…			
VB	0	…			
JJ	0	…			
NN	0	…			
RB	0	…			
DT	0	…			

THE FORWARD ALGORITHM

	Janet	will	back	the	bill
NNP	8.8544e-06	$P(will \mid NNP)$ * $P(NNP \mid t_{Janet})$ * $a(t_{Janet} \mid Janet)$			
MD	0	…			
VB	0				
JJ	0	…			
NN	0	…			
RB	0	…			
DT	0	…			

Calculate this for all tags, take the **sum**.

THE FORWARD ALGORITHM

	Janet	will	back	the	bill
NNP	8.8544e-06	0			
MD	0	3.004e-8			
VB	0	2.231e-13			
JJ	0	0			
NN	0	1.034e-10			
RB	0	0			
DT	0	0			

THE FORWARD ALGORITHM

	Janet	will	back	the	bill
NNP	8.8544e-06	0	0		
MD	0	3.004e-8	0		
VB	0	2.231e-13	P(back|VB) * P(VB|t_{will}) * s(t_{will}|will)		
JJ	0	0			
NN	0	1.034e-10			
RB	0	0			
DT	0	0			

THE FORWARD ALGORITHM

	Janet	will	back	the	bill
NNP	8.8544e-06	0	0		
MD	0	3.004e-8	0		
VB	0	2.231e-13	**MD: 1.6e-11** **VB: 7.5e-19** **NN: 9.7e-17**		
JJ	0	0			
NN	0	1.034e-10			
RB	0	0			
DT	0	0			

THE FORWARD ALGORITHM

	Janet	will	back	the	bill
NNP	8.8544e-06	0	0		
MD	0	3.004e-8	0		
VB	0	2.231e-13	1.6e-11		
JJ	0	0			
NN	0	1.034e-10			
RB	0	0			
DT	0	0			

THE FORWARD ALGORITHM

	Janet	will	back	the	bill
NNP	8.8544e-06	0	0		
MD	0	3.004e-8	0		
VB	0	2.231e-13	1.6e-11		
JJ	0	0	5.42e-15		
NN	0	1.034e-10	8.17e-15		
RB	0	0	5.33e-11		
DT	0	0	0		

THE FORWARD ALGORITHM

	Janet	will	back	the	bill
NNP	8.8544e-06	0	0	4.21e-17	0
MD	0	3.004e-8	0	0	0
VB	0	2.231e-13	1.6e-11	0	1.74e-20
JJ	0	0	5.42e-15	6.53e-16	0
NN	0	1.034e-10	8.17e-15	9.76e-18	3.44e-15
RB	0	0	5.33e-11	0	0
DT	0	0	0	3.1e-12	0

THE FORWARD ALGORITHM

	Janet	will	back	the	bill
NNP	8.8544e-06	0	0	4.21e-17	0
					0
					1.74e-20
					0
					3.44e-15
					0
DT	0	0	0	3.1e-12	0

- Final probability also needs to take into account the end state “</s>”
- $P(\mathbf{w}) = \alpha(\text{bill,VB}) * P(<\text{end}>|\text{VB}) + \alpha(\text{bill,NN}) * P(<\text{end}>|\text{NN})$
- $P(\mathbf{w}) = 1.74\text{e-}20 * 0.0004 + 3.44\text{e-}15 * 0.237$
- $P(\mathbf{w}) = 8.15\text{e-}16$

MLE WITH EXPECTED COUNTS

- We got $\hat{P}(\boldsymbol{w})$ using the forward algorithm. Now we need $\hat{P}(t_i = NNP, \boldsymbol{w})$.
  - $\hat{P}(t_i = NNP, \boldsymbol{w}) = \hat{P}(\boldsymbol{w}_{1:i}, t_i = NNP, \boldsymbol{w}_{i+1:l_w})$
  - $\hat{P}(t_i = NNP, \boldsymbol{w}) = \hat{P}(\boldsymbol{w}_{1:i}, t_i = NNP)\ \hat{P}\left(\boldsymbol{w}_{i+1:l_w} | \boldsymbol{w}_{1:i}, t_i = NNP\right)$
  - $\hat{P}(t_i = NNP, \boldsymbol{w}) = \hat{P}(\boldsymbol{w}_{1:i}, t_i = NNP)\ \hat{P}(\boldsymbol{w}_{i+1:l_w} | t_i = NNP)$
- $\hat{P}(\boldsymbol{w}_{1:i}, t_i = NNP) = \alpha(i, NNP)$
  - We got this from the forward algorithm!
- $\hat{P}\left(\boldsymbol{w}_{i+1:l_w} | t_i = NNP\right) = \beta(i, NNP)$
  - We will get these through the **backward** algorithm.

THE BACKWARD ALGORITHM

	Janet	will	back	the	bill
NNP					
MD					
VB					
JJ					
NN					
RB					
DT					

THE BACKWARD ALGORITHM

	Janet	will	back	the	bill
NNP					1
MD					1
VB					1
JJ					1
NN					1
RB					1
DT					1

- $\beta(i, NNP) = \hat{P}\left(\boldsymbol{w}_{i+1:l_w} | t_i = NNP\right)$
- $\beta(5, NNP) = \hat{P}(\boldsymbol{w}_{6:5} | t_5 = NNP)$
- This is a “non-event”. P(Ø) = 1

THE BACKWARD ALGORITHM

	Janet	will	back	the	bill
NNP				$\sum$ P(bill|t_{bill}) * P(NNP|t_{bill}) * ß(bill,t_{bill})	1
MD				$\sum$ P(bill|t_{bill}) * P(MD|t_{bill}) * ß(bill,t_{bill})	1
VB				…	1
JJ				…	1
NN				…	1
RB				…	1
DT				…	1

OBTAINING THE PROBABILITIES

- After running forward and backward, we obtain two matrices containing **α's and β's**,
- $\hat{P}(\boldsymbol{w}) = \sum_t \alpha(l_{\boldsymbol{w}}, t) * \hat{P}(< end > |t)$
- $\hat{P}(t_i = NNP, \boldsymbol{w}) = \alpha(i, NNP) * \beta(i, NNP)$
- $\hat{c}(will, NNP) = \sum_{\boldsymbol{w}} \sum_{i=1;i=will}^{lw} \frac{\alpha(i,NNP) * \beta(i,NNP)}{\sum_t \alpha(l_{\boldsymbol{w}},t) * \hat{P}(<end>|t)}$
- $\hat{c}(NNP) = \sum_{\boldsymbol{w}} \sum_{i=1}^{lw} \frac{\alpha(i,NNP) * \beta(i,NNP)}{\sum_t \alpha(l_{\boldsymbol{w}},t) * \hat{P}(<end>|t)}$
- $P(will|NNP) = \frac{\hat{c}(will,NNP)}{\hat{c}(NNP)}$

TRANSITION PROBABILITIES

- Transition probabilities are also updated using expected counts.
  - We can use the values from forward-backward as well.
- $\xi_i(NN, DT) = \frac{\alpha(i,DT) * a[DT,NN] * b(w_{i+1},NN) * \beta(i+1,NN)}{P(\boldsymbol{w})}$
- $\hat{c}(NN, DT) = \sum_{w} \sum_{i=1}^{l_w} \xi_i(NN, DT)$
- $\hat{c}(DT) = \sum_{w} \sum_{i=1}^{l_w} \sum_{t} \xi_i(t, DT)$
- $P(NN|DT) = \frac{\hat{c}(NN,DT)}{\hat{c}(DT)}$

EM - FINAL ALGORITHM

- Initialise emission and transition matrices
- **E-step:**
  - Run forward-backward, obtaining **α's** and **β's**
- **M-step:**
  - Update emission and transition matrices using the expected counts.

EM – IMPORTANT POINTS

- In our example, we initialised the parameters (matrices) at random. In practice initialisation matters so care must be taken.
  - Take values from a previous known tagger.
  - Use dictionaries to initialise some values ("the" is never a verb).
- Forward scores can be **conditional** instead of joint. Mitigates underflow.
- How to evaluate?
  - Need some annotated data for that.

A FINAL WORD

- Unsupervised learning is key to generalise sequence tagging to other domains and languages
- HMMs can easily do that using EM.
- Some annotated data is always necessary for evaluation.

READINGS

- JM3 Ch 9.3, 9.5
- [Optional] Rabiner's HMM tutorial, for more details
  - http://tinyurl.com/2hqaf8