NLP

Introduction to NLP

Statistical POS Tagging

Part of Speech Tagging Methods

- Rule-based
- Stochastic
  - HMM (generative)
  - Maximum Entropy MM (discriminative)
- Transformation-based

HMM Tagging

- T = argmax P(T|W)
  - where $T=t_1,t_2,\ldots,t_n$
- By Bayes' theorem
  - P(T|W) = P(T)P(W|T)/P(W)
- Thus we are attempting to choose the sequence of tags that maximizes the right hand side of the equation
  - P(W) can be ignored
  - P(T) is called the prior, P(W|T) is called the likelihood.

HMM Tagging

- Complete formula
  - $P(T)P(W|T) = \prod P(w_i|w_1t_1 \ldots w_{i-1}t_{i-1}t_i)P(t_i|t_1 \ldots t_{i-2}t_{i-1})$
- Simplification 1:
  - $P(W|T) = \prod P(w_i|t_i)$
- Simplification 2:
  - $P(T) = \prod P(t_i|t_{i-1})$
- Bigram approximation
  - $T = \text{argmax } P(T|W) = \text{argmax } \prod P(w_i|t_i)\ P(t_i|t_{i-1})$

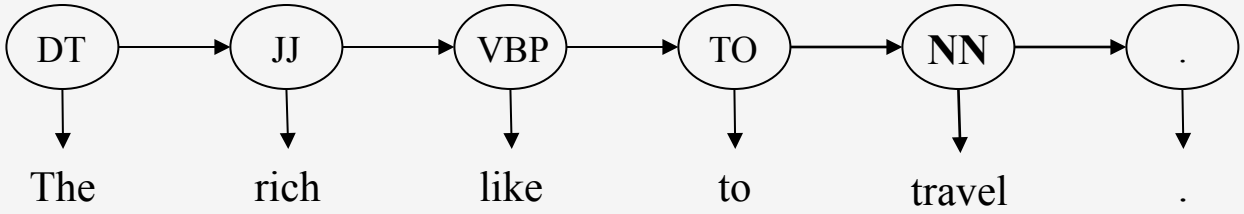
Maximum Likelihood Estimates

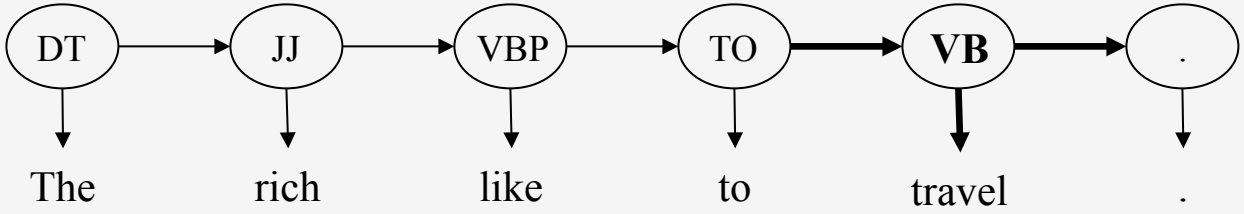
- P(NN|JJ) = C(JJ,NN)/C(JJ)= 22301/89401= .249
- P(this|DT) = C(DT,this)/C(dT)= 7037/103687 = .068

Example

- The/DT rich/JJ like/VBP to/TO travel/VB ./.

Example


DT
JJ
VBP
TO
NN
.
The
rich
like
to
travel
.


DT
JJ
VBP
TO
VB
.
The
rich
like
to
travel
.

Evaluating Taggers

- Data set
  - Training set
  - Development set
  - Test set
- Tagging accuracy
  - how many tags right
- Results
  - Accuracy around 97% on PTB trained on 800,000 words
  - (50–85% on unknown words; 50% for trigrams)
  - Upper bound 98% – noise (e.g., errors and inconsistencies in the data, e.g., NN vs JJ)

Transformation-Based Learning

- [Brill 1995]
- Example
  - P(NN|sleep) = .9
  - P(VB|sleep) = .1
  - Change NN to VB when the previous tag is TO
- Types of rules:
  - The preceding (following) word is tagged z
  - The word two before (after) is tagged z
  - One of the two preceding (following) words is tagged z
  - One of the three preceding (following) words is tagged z
  - The preceding word is tagged z and the following word is tagged w

Transformation Based Tagger

#	Change Tag From	Change Tag To	Condition
1	NN	VB	Previous tag is *TO*
2	VBP	VB	One of the previous three tags is *MD*
3	NN	VB	One of the previous two tags is *MD*
4	VB	NN	One of the previous two tags is *DT*
5	VBD	VBN	One of the previous three tags is *VBZ*
6	VBN	VBD	Previous tag is *PRP*
7	VBN	VBD	Previous tag is *NNP*
8	VBD	VBN	Previous tag is *VBD*
9	VBP	VB	Previous tag is *TO*
10	POS	VBZ	Previous tag is *PRP*
11	VB	VBP	Previous tag is *NNS*
12	VBD	VBN	One of previous three tags is *VBP*
13	IN	WDT	One of next two tags is *VB*
14	VBD	VBN	One of previous two tags is *VB*
15	VB	VBP	Previous tag is *PRP*
16	IN	WDT	Next tag is *VBZ*
17	IN	DT	Next tag is *NN*
18	JJ	NNP	Next tag is *NNP*
19	IN	WDT	Next tag is *VBD*
20	JJR	RBR	Next tag is *JJ*

Figure 4
The first 20 nonlexicalized transformations.

Transformation Based Tagger

Change tag **a** *to tag* **b** *when:*

1. The preceding (following) word is w.
2. The word two before (after) is w.
3. One of the two preceding (following) words is w.
4. The current word is w and the preceding (following) word is x.
5. The current word is w and the preceding (following) word is tagged z.
6. The current word is w.
7. The preceding (following) word is w and the preceding (following) tag is t.
8. The current word is w, the preceding (following) word is w_2 and the preceding (following) tag is t.

where w and x are variables over all words in the training corpus, and z and t are variables over all parts of speech.

Transformation Based Tagger

	Change Tag		
#	From	To	Condition
1	NN	NNS	Has suffix **-s**
2	NN	CD	Has character **.**
3	NN	JJ	Has character **-**
4	NN	VBN	Has suffix **-ed**
5	NN	VBG	Has suffix **-ing**
6	??	RB	Has suffix **-ly**
7	??	JJ	Adding suffix **-ly** results in a word.
8	NN	CD	The word **$** can appear to the left.
9	NN	JJ	Has suffix **-al**
10	NN	VB	The word **would** can appear to the left.
11	NN	CD	Has character **0**
12	NN	JJ	The word **be** can appear to the left.
13	NNS	JJ	Has suffix **-us**
14	NNS	VBZ	The word **it** can appear to the left.
15	NN	JJ	Has suffix **-ble**
16	NN	JJ	Has suffix **-ic**
17	NN	CD	Has character **1**
18	NNS	NN	Has suffix **-ss**
19	??	JJ	Deleting the prefix **un-** results in a word
20	NN	JJ	Has suffix **-ive**

Figure 6
The first 20 transformations for unknown words.

Thoughts About POS Taggers

- New domains
  - Lower performance
- Distributional clustering
  - Combine statistics about semantically related words
  - Example: names of companies
  - Example: days of the week
  - Example: animals

External Links

- Jason Eisner's awesome interactive spreadsheet about learning HMMs
  - http://cs.jhu.edu/~jason/papers/#eisner-2002-tnlp
  - http://cs.jhu.edu/~jason/papers/eisner.hmm.xls

NLP