Part of Speech Tagging (Chapter 5)

Lecture #6

September 2009

New Topic: Syntax

- Up until now we have been dealing with individual words and simple-minded (though useful) notions of what sequence of words are likely.
- Now we turn to the study of how words
  - Are clustered into classes
  - Group with their neighbors to form phrases and sentences
  - Depend on other words
- Interesting notions:
  - Word order
  - Consittuency
  - Grammatical relations
- Today: syntactic word classes – part of speech tagging

What is a word class?

- Words that somehow 'behave' alike:
  - Appear in similar contexts
  - Perform similar functions in sentences
  - Undergo similar transformations

Why do we want to identify them?

- Someone say
  - Refuse
  - Project
  - Compact
  - Content
  - Discount
- Why do we want to identify them?
  - Pronunciation (desert/desert)
  - Stemming
  - Semantics
  - More accurate N-grams
  - Simple syntactic information

How many word classes are there?

- A basic set:
  - N, V, Adj, Adv, Prep, Det, Aux, Part, Conj, Num
- A simple division: open/content vs. closed/function
  - Open: N, V, Adj, Adv
  - Closed: Prep, Det, Aux, Part, Conj, Num
- Many subclasses, e.g.
  - eats/V $\Rightarrow$ eat/VB, eat/VBP, eats/VBZ, ate/VBD, eaten/VBN, eating/VBG, ...
  - Reflect morphological form & syntactic function

How do we decide which words go in which classes?

- Nouns denote people, places and things and can be preceded by articles? But…
  My typing is very bad.
  *The Mary loves John.
- Verbs are used to refer to actions and processes
  - But some are closed class and some are open
  I will have emailed everyone by noon.
- Adjectives describe properties or qualities, but
  a cat sitter, a child seat

- Adverbs include locatives (here), degree modifiers (very), manner adverbs (gingerly) and temporals (today)
  - Is Monday a temporal adverb or a noun?
- Closed class items (Prep, Det, Pron, Conj, Aux, Part, Num) are easier, since we can enumerate them….but
  - Part vs. Prep
    - George eats up his dinner/George eats his dinner up.
    - George eats up the street/*George eats the street up.
  - Articles come in 2 flavors: definite (the) and indefinite (a, an)

7

  - Conjunctions also have 2 varieties, coordinate (and, but) and subordinate/complementizers (that, because, unless,…)
  - Pronouns may be personal (I, he,...), possessive (my, his), or wh (who, whom,...)
  - Auxiliary verbs include the copula (be), do, have and their variants plus the modals (can, will, shall,…)
- And more…
  - Interjections/discourse markers
  - Existential there
  - Greetings, politeness terms

8

Tagsets

- What set of parts of speech do we use?
- Most tagsets implicitly encode fine-grained specializations of 8 basic parts of speech (POS, word classes, morphological classes, lexical tags):
  - Noun, verb, pronoun, preposition, adjective, conjunction, article, adverb
- These categories are based on morphological and distributional similarities and not, as you might think, semantics.
- In some cases, tagging is fairly straightforward (at least in a given language), in other cases it is not.

9

Distribution of Tags

- Parts of speech follow the usual frequency-based distributional behavior
  - Most word types have only one part of speech
  - Of the rest, most have two
  - A small number of word types have lots of parts of speech
  - Unfortunately, the word types with lots of parts of speech occur with high frequency (and words that occur most frequently tend to have multiple tags)

10

Distribution of Tags – Brown

- To see the problem:
  - 11.5% of English words in the Brown corpus are ambigjous
  - 40% of tokens in the Brown corpus are ambignous

Unambiguous	(1 tag)	35,340	
Ambiguous	(2-7 tags)	4,100	
	2 tags	3,760	
	3 tags	264	
	4 tags	61	
	5 tags	12	
	6 tags	2	
	7 tags	1	(“still”)

11

The Brown Corpus

- The Brown Corpus of Standard American English was the first of the modern, computer readable general corpora. (Compiled at Brown University)
- Corpus consists of 1 million words of American English text printed in 1961.
- For a long time, Brown and LOB (British) corpora were the only easily available online, so many studies have been done on these corpora.
- Studying the same data allows comparison of findings without having to take into consideration possible variation caused by the use of different data.
- But…?
- Tagged version of Brown is available.

12

Tagsets

- There are various standard tagsets to choose from; some have a lot more tags than others
- The choice of tagset is based on the application
- Accurate tagging can be done with even large tagsets

13

So how do we choose a Tagset?

- http://www.comp.leeds.ac.uk/amalgam/tagsets/tagmenu.html
- Brown Corpus (Francis & Kucera '82), 1M words, 87 tags.
  - http://www.comp.leeds.ac.uk/amalgam/tagsets/brown.html
- Penn Treebank: hand-annotated corpus of *Wall Street Journal*, 1M words, 45-46 tags
  - http://www.comp.leeds.ac.uk/amalgam/tagsets/upenn.html

14

Tagsets

- How do tagsets differ?
  - Degree of granularity
  - Idiosyncratic decisions, e.g. Penn Treebank doesn't distinguish to/Prep from to/Inf, eg.
  - I/PP want/VBP to/TO go/VB to/TO Zanzibar/NNP ./.
  - Don't tag it if you can recover from word (e.g. do forms)

15

What does Tagging do?

1. Collapses distinctions
   - E.g., all personal pronouns tagged as PRP
   - Lexical identity may be completely discarded
2. Introduces distinctions (by reducing ambiguity)
   - E.g., deal tagged with NN or VB

16

Tagging

- Part of speech tagging is the process of assigning parts of speech to each word in a sentence
- Assume we have
  - A tagset
  - A dictionary that gives you the possible set of tags for each entry
  - A text to be tagged
- Output
  - Single best tag for each word
  - E.g., Book/VB that/DT flight/NN

17

Part-of-Speech Tagging

- How do we assign POS tags to words in a sentence?
  - Get/V the/Det bass/N
  - Time flies like an arrow.
  - Time/[V,N] flies/[V,N] like/[V,Prep] an/Det arrow/N
  - Time/N flies/V like/Prep an/Det arrow/N
  - Fruit/N flies/N like/V a/DET banana/N
  - Fruit/N flies/V like/V a/DET banana/N
  - The/Det flies/N like/V a/DET banana/N

18

Just for Fun…

- Using Penn Treebank tags, tag the following sentence from the Brown Corpus:
- The grand jury commented on a number of other topics.

19

Just for Fun…

- Using Penn Treebank tags, tag the following sentence from the Brown Corpus:
- The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

20

Why is Tagging Hard?

- Example
  - Book/VB that/DT flight/NN
  - Does/VBZ that/DT flight/NN serve/VB dinner/NN
- Tagging is a type of disambiguation
  - Book can be NN or VB
  - Can I read a book on this flight?
  - That can be a DT or complementizer
  - My travel agent said that there would be a meal on this flight.

21

Potential Sources of Disambiguation

- Many words have only one POS tag (e.g. is, Mary, very, smallest)
- Others have a single most likely tag (e.g. a, dog)
- But tags also tend to co-occur regularly with other tags (e.g. Det, N)
- In addition to conditional probabilities of words $P(w_1|w_{n-1})$, we can look at POS likelihoods $P(t_1|t_{n-1})$ to disambiguate sentences and to assess sentence likelihoods

22

Approaches to POS Tagging

- Rule-based Approach
  - Uses handcrafted sets of rules to tag input sentences
- Statistical approaches
  - Use training corpus to compute probability of a tag in a context
- Hybrid systems (e.g. Brill's transformation-based learning)

23

ENGTWOL Rule-Based Tagger

A Two-stage architecture

- Use lexicon FST (dictionary) to tag each word with all possible POS
- Apply hand-written rules to eliminate tags.
- The rules eliminate tags that are inconsistent with the context, and should reduce the list of POS tags to a single POS per word.

24

Det-Noun Rule:

- If an ambiguous word follows a determiner, tag it as a noun

25

ENGTWOL Adverbial-that Rule

Given input "that"

- **If** the next word is adj, adverb, or quantifier, and following that is a sentence boundary, and the previous word is not a verb like "consider" which allows adjs as object complements,
- **Then** eliminate non-ADV tags,
- **Else** eliminate ADV tag

- I consider that odd. (that is NOT ADV)
- It isn't that strange. (that is an ADV)

26

Does it work?

- This approach does work and produces accurate results.
- What are the drawbacks?
  - Extremely labor-intensive

27

Statistical Tagging

- Statistical (or stochastic) taggers use a training corpus to compute the probability of a tag in a context.
- For a given word sequence, Hidden Markov Model (HMM) Taggers choose the tag sequence that maximixes

  P(word | tag) * P(tag | previous-n-tags)

A bigram HMM tagger chooses the tag t_i for word w_i that is most probable given the previous tag, t_{i-1}

$$t_i = \text{argmax}_j P(t_j \mid t_{i-1}, w_i)$$

28

Statistical POS Tagging (Allen95)

- Let's step back a minute and remember some probability theory and its use in POS tagging.
- Suppose, with no context, we just want to know given the word "flies" whether it should be tagged as a noun or as a verb.
- We use conditional probability for this: we want to know which is greater

  PROB(N | flies) or PROB(V | flies)
- Note definition of conditional probability

  PROB(a | b) = PROB(a & b) / PROB(b)

  – Where PROB(a & b) is the probability of the two events a dn b occurring simultaneously

29

Calculating POS for "flies"

We need to know which is more

- PROB(N | flies) = PROB(flies & N) / PROB(flies)
- PROB(V | flies) = PROB(flies & V) / PROB(flies)

- Count on a Corpus

30

Corpus to Estimate

1,273,000 words; 1000 uses of flies; 400 flies in N sense; 600 flies in V sense

PROB(flies) ≈ 1000/1,273,000 = .0008

PROB(flies & N) ≈ 400/1,273,000 = .0003

PROB(flies & V) ≈ 600/1,273,000 = .0005

Out best guess is that flies is a V

PROB(V | flies) = PROB(V & flies) / PROB(flies)

= .0005/.0008 = .625

31

Doing Better

- Simple Method: Always choose the tag that appears most frequently in the training set – will work correctly about 91% of the time.
- How to do better: Consider more of the context. Knowing "the flies" gives much higher probability of a Noun
- General Equation: find the sequence of tags that maximizes:

$$PROB(T_1,\ldots T_n \mid w_1,\ldots w_n)$$

32

Estimating Too Hard

$$PROB(T_1,\ldots T_n \mid w_1,\ldots w_n)$$

Estimating the above takes far too much data. Need to do some reasonable approximations.

Bayes Rule:

PROB(A | B) = PROB(B | A) * PROB(A) / PROB(B)

Rewriting:

$PROB(w_1,\ldots w_n \mid T_1,\ldots T_n) * PROB(T_1,\ldots T_n) / PROB(w_1,\ldots w_n)$

33

$PROB(w_1,\ldots w_n \mid T_1,\ldots T_n) * PROB(T_1,\ldots T_n) / PROB(w_1,\ldots w_n)$

Remember we are interested in finding the sequence of tags that maximizes this formula – so we can ignore $PROB(w_1,\ldots w_n)$ since it is always the same

So, we want to find the sequence of tags that maximizes

$$PROB(w_1,\ldots w_n \mid T_1,\ldots T_n) * PROB(T_1,\ldots T_n)$$

This is still too hard to calculate – so we need to make some independence assumptions.

34

Independence Assumptions

So, we want to find the sequence of tags that maximizes

$$PROB(T_1,\ldots T_n) * PROB(w_1,\ldots w_n \mid T_1,\ldots T_n)$$

For Tags – use bigram probabilities

$PROB(T_1,\ldots T_n) \approx \prod_{i=1,n} PROB(T_i \mid T_{i-1})$

PROB(ART N V N) ≈ PROB(ART | Φ) * PROB(N | ART) * PROB(V | N) * PROB(N | V)

For second probability: assume word tag is independent of words around it:

$PROB(w_1,\ldots w_n \mid T_1,\ldots T_n) \approx \prod_{i=1,n} PROB(w_i \mid T_i)$

35

POS Formula

- Find the sequence of tags that maximizes:

$$\prod_{i=1,n} PROB(T_i \mid T_{i-1}) * PROB(w_i \mid T_i)$$

- These probabilities can be estimated from a corpus of text labeled with parts of speech. (See handout which takes us through calculating for whole sequence – to slide 47)

36

Statistical Tagging (cont.)

- Making some simplifying Markov assumptions, the basic HMM equation for a single tag is:

$$t_i = \text{argmax}_j\ P(t_j \mid t_{i-1}) * P(w_i \mid t_j)$$

  - The function $\text{argmax}_x F(x)$ means "the x such that F(x) is maximized"
  - The first P is the tag sequence probability, the second is the word likelihood given the tag.
- Most of the better statistical models report around 95% accuracy on standard datasets
- But, note you get 91% accuracy just by picking the most likely tag!

37

A Simple Example

- From the Brown Corpus
- Secretariat/NNP is/VBZ expected/VBN to/TO *race*/VB tomorrow/NN
- People/NNS continue/VBP to/TO inquire/VB the/DT reason/NN for/IN the/DT *race*/NN for/IN outer/JJ space/NN

Assume previous words have been tagged, and we want to tag the word *race*.

Bigram tagger

- to/TO *race*/?
- the/DT *race*/?

38

Example (cont,.)

- Goal: choose between NN and VB for the sequence *to race*
- Plug these into our bigram HMM tagging equation:

P(word | tag) * P(tag | previous-n-tags)

- P(race | VB) * P(VB | TO)
- P(race | NN) * P(NN | TO)

How do we compute the tag sequence probabilities and the word likelihoods?

39

Word Likelihood

- We must compute the likelihood of the word race given each tag. I.e., P(race | VB) and P(race | NN)
- Note: we are NOT asking which is the most likely tag for the word.
- Instead, we are asking, if we were expecting a verb, how likely is it that this verb would be *race*?
- From the Brown and Switchboard Corpora:
  P(*race* | VB) = .00003
  P(*race* | NN) = .00041

40

Tag Sequence Probabilities

- Computed from the corpus by counting and normalizing.
- We expect VB more likely to follow TO because infinitives (*to race, to eat*) are common in English, but it is possible for NN to follow TO (*walk to school, related to fishing*).
- From the Brown and Switchboard corpora:
  P(VB | TO) = .340
  P(NN | TO) = .021

41

And the Winner is…

Multiplying tag sequence probabilities by word likelihoods gives

- P(*race* | VB) * P(VB | TO) = .000010
- P(*race* | NN) * P(NN | TO) = .000007

So, even a simple bigram version correctly tags race as a VB, despite the fact that it is the less likely sense.

42

Statistical POS Tagging (whole sequence)

- Goal: choose the best sequence of tags T for a sequence of words W in a sentence

$$T' = \arg\max_{T \in \tau} P(T|W)$$

  - By Bayes Rule (giving us something easier to calculate)

$$P(T|W) = \frac{P(T)P(W|T)}{P(W)}$$

  - Since we can ignore P(W), we have

$$T' = \arg\max_{T \in \tau} P(T)P(W|T)$$

43

Statistical POS Tagging: the Prior

$P(T) = P(t_1, t_2, \ldots, t_{n-1}, t_n)$

By the Chain Rule:

$= P(t_n \mid t_1, \ldots, t_{n-1})\, P(t_1, \ldots, t_{n-1})$

$= \prod_{i=1}^{n} P(t_i|t_1^{i-1})$

Making the Markov assumption:

e.g., for bigrams,

$\approx P(t_i|t_{i-N+1}^{i-1})$ $\prod_{i=1}^{n} P(t_i|t_{i-1})$

44

Statistical POS Tagging: the (Lexical) Likelihood

$P(W|T) = P(w_1, w_2, \ldots, w_n \mid t_1, t_2, \ldots, t_n)$

From the Chain Rule:

$= \prod_{i=1}^{n} P(w_i|w_1 t_1 \ldots w_{i-1} t_{i-1} t_i)$

Simplifying assumption: probability of a word depends only on its own tag $P(w_i|t_i)$

So... $\approx \prod_{i=1}^{n} P(w_i|t_i)$

$$T' = \arg\max_{T \in \tau} \prod_{i=1}^{n} P(t_i|t_{i-1}) \prod_{i=1}^{n} P(w_i|t_i)$$

45

Estimate the Tag Priors and the Lexical Likelihoods from Corpus

- Maximum-Likelihood Estimation
  - For bigrams:

$P(t_i|t_{i-1}) = c(t_{i-1}, t_i)/c(t_{i-1})$

$P(w_i|t_i) = \frac{c(wt)}{c(t)}$

46

Performance

- This method has achieved 95-96% correct with reasonably complex English tagsets and reasonable amounts of hand-tagged training data.

47

Transformation-Based (Brill) Tagging

A hybrid approach

- Like rule-based taggers, this tagging is based on rules
- Like (most) stochastic taggers, rules are also automatically induced from hand-tagged data

Basic Idea: do a quick and dirty job first, and then use learned rules to patch things up

Overcomes the pure rule-based approach problems of being too expensive, too slow, too tedious etc…

An instance of Transformation-Based Learning.

48

Transformation-Based Tagging

- Combine rules and statistics…
- Start with a dumb statistical system and patch up the typical mistakes it makes.
- How dumb?
  - Assign the most frequent tag (unigram) to each word in the input

49

Examples

- Race
  - "race" as NN: .98
  - "race" as VB: .02
- So you'll be wrong 2% of the time, which really isn't bad
- Patch the cases where you know it has to be a verb
  - Change NN to VB when previous tag is TO

50

Brill's Tagger 3 Stages

1. Label every word with its most likely tag.
2. Examine every possible transformation, and select the one that results in the most improved tagging.
3. Re-tag the data according to the selected rule.

Go to 2 until stopping criterion is reached.

Stopping:
Insufficient improvement over previous pass.

Output: Ordered list of transformations. These constitute a tagging procedure.

51

Rules

- Where did that transformational rule come from?
- In principle, the set of possible rules is infinite.
  - Use set of rule templates to define possible rules to try in the search.

52

Hypothesis Space

- In Brill tagging it's defined by a set of templates of the form
  - Change tag a to tag b when …

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.
The preceding (following) word is tagged z and the word two before (after) is tagged w.

- a, b, w and z range over the tags

53

How?

- Deciding whether or not to accept a transformation depends on the overall change made by the rule.
- If a given tag change rule makes things better (fixes tags that were wrong) should you always accept it?
  - No. It might break things that were right.

54

Brill Tagging: TBL

- Start with simple (less accurate) rules…learn better ones from tagged corpus
  - Tag each word initially with most likely POS
  - Examine set of transformations to see which improves tagging decisions compared to tagged corpus
  - Re-tag corpus using best transformation
  - Repeat until, e.g., performance doesn't improve
  - Result: tagging procedure (ordered list of transformations) which can be applied to new, untagged text

55

An Example

The horse raced past the barn fell.

The/DT horse/NN raced/VBN past/IN the/DT barn/NN fell/VBD ./.

1) Tag every word with most likely tag and score

The/DT horse/NN raced/VBD past/NN the/DT barn/NN fell/VBD ./.

2) For each template, try every instantiation (e.g. Change VBN to VBD when the preceding word is tagged NN, add rule to ruleset, retag corpus, and score

56

3) Stop when no transformation improves score

4) Result: set of transformation rules which can be applied to new, untagged data (after initializing with most common tag)

….What problems will this process run into?

57

Methodology: Evaluation

- For any NLP problem, we need to know how to evaluate our solutions
- Possible Gold Standards -- ceiling:
  - Annotated naturally occurring corpus
  - Human task performance (96-7%)
    - How well do humans agree?
    - Kappa statistic: avg pairwise agreement corrected for chance agreement
  - Can be hard to obtain for some tasks

58

- Baseline: how well does simple method do?
  - For tagging, most common tag for each word (91%)
  - How much improvement do we get over baseline

59

Methodology: Error Analysis

- Confusion matrix:
  - E.g. which tags did we most often confuse with which other tags?
  - How much of the overall error does each confusion account for?

60

More Complex Issues

- Tag indeterminacy: when 'truth' isn't clear
  Carribean cooking, child seat
- Tagging multipart words
  wouldn't --> would/MD n't/RB
- Unknown words
  - Assume all tags equally likely
  - Assume same tag distribution as all other singletons in corpus
  - Use morphology, word length,….

61

Summary

- We can develop statistical methods for identifying the POS of word sequences which reach close to human performance
- But not completely "solved"
- Next Class: Guest Lecture by Owen Rambow
  - Read Chapter 9
  - Homework 1 due

62