Introduction to Natural Language Processing

Chapter 8

Outline

Homework due; discussion schedule and next homework update

New Topic

Parts of Speech

English Parts of Speech

Tagging

Tagging with Rules

Transformation-Based Tagging

Statistical Tagging

Alternative Language Models

N-gram language models are an extremely impoverished attempt to capture what we know about likely sequences of words.

Clearly, syntactic, semantic, and discourse expectations play a role.

New Topic: Syntax

Up until now we've been dealing with individual words and simple-minded (though useful) notions of what sequences of words are likely.

Now we will turn to the study of how words

- are clustered into classes
- group with their neighbors to form phrases and sentences
- depend on other words

We'll be interested in *word order, constituency, grammatical relations.*

But we'll start with syntactic word classes.

Parts of Speech

Most tagsets implicitly encode fine-grained specializations of eight 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, this is straightforward (at least in a given language), in other cases it's not.

The Distribution of Tags

Tags follow all the usual frequency-based distributional behavior.

- most word types have only one part of speech
- of the rest, most have two, etc.
- the most frequently occurring word types tend to have multiple tags (and as we'll see later, they also tend to have more meanings)

Therefore, while it's easy to determine the correct tag for most wordtypes, it isn't necessarily so easy to tag most texts.

Notes on Tagsets

There are various tagsets for formally coding POS.

The choice of tagset pretty much depends on the nature of the application.

Since accurate tagging can be performed with relatively large tagsets it makes sense to use one of the larger standard sets. If it makes distinctions you don't need, you can merge the finer grained tags.

Why Tag POS?

Language Modeling

Pronunciation

Stemming

Parsing

Word Sense Disambiguation

Information Extraction

English Word Classes

Open (lexical) class types

- new words are coined/borrowed
- nouns, verbs, adjectives, adverbs

Closed (functional) class types

- relatively fixed membership
- small class, but frequently occur
- pronouns, prepositions, conjunctions, articles ...

Nouns

Take possessives, occur in plural form, occur with determiners . . .

Proper nouns (*Diane*)

Common nouns

- count nouns (*professor, student, computer*)
- mass nouns (*snow, air*)

Vary in

- number (singular, plural)
- gender (masculine, feminine, neuter)
- case (nominative, genitive, accusative, dative)
- Latin example: filius (m), filia (f), filium (object)

Verbs

Refer to actions, activities, processes, states (*throw, walk, have*)

Tenses: present, past, future, ...

Other inflection: number, person

Voice: active, passive

Standard morphological forms, as previously discussed.

- stem or non-3rd-person-sg (*eat*)
- -s form or 3rd-person-sg (*eats*)
- -ing participle or progressive (*eating*)
- past participle (*eaten*)

Irregular verbs

Auxiliary subclass of verbs are closed class, however.

Adjectives and Adverbs

Adjectives (semantically) describe properties or qualities.

- color (*blue*)
- age (*old*)
- value (*good*)

Adverbs also modify something, often verbs but also other adverbs and verb phrases.

- directional or locative (*downhill*)
- degree (*extremely*)
- manner (*slowly*)
- temporal (*yesterday*)

Pronouns

Roughly, a shorthand for referring to a noun phrase or entity.

- personal (*you, I, me*)
- possessive (*my, yours, mine, ours*)
- wh- (*what, who, whom, whoever, where, when*)

Vary in

- person
- gender
- number
- case (in English, nominative, accusative, possessive, 2nd possessive, reflexive)

Pronouns (continued)

English pronouns from CELEX on-line dictionary, with frequency counts from a 16 million word corpus.

it	199,920	how	13,137	yourself	2,437	no one	106
I	198,139	another	12,551	why	2,220	wherein	58
he	158,366	where	11,857	little	2,089	double	39
you	128,688	same	11,841	none	1,992	thine	30
his	99,820	something	11,754	nobody	1,684	summat	22
they	88,416	each	11,320	further	1,666	suchlike	18
this	84,927	both	10,930	everybody	1,474	fewest	15
that	82,603	last	10,816	ourselves	1,428	thyself	14
she	73,966	every	9,788	mine	1,426	whomever	11
her	69,004	himself	9,113	somebody	1,322	whosoever	10
we	64,846	nothing	9,026	former	1,177	whomsoever	8
all	61,767	when	8,336	past	984	wherefore	6
which	61,399	one	7,423	plenty	940	whereat	5
their	51,922	much	7,237	either	848	whatsoever	4
what	50,116	anything	6,937	yours	826	whereon	2
my	46,791	next	6,047	neither	618	whoso	2
him	45,024	themselves	5,990	fewer	536	aught	1
me	43,071	most	5,115	hers	482	howsoever	1
who	42,881	itself	5,032	ours	458	thrice	1
them	42,099	myself	4,819	whoever	391	wheresoever	1
no	33,458	everything	4,662	least	386	you-all	1
some	32,863	several	4,306	twice	382	additional	0
other	29,391	less	4,278	theirs	303	anybody	0
your	28,923	herself	4,016	wherever	289	each other	0
its	27,783	whose	4,005	oneself	239	once	0
our	23,029	someone	3,755	thou	229	one another	0
these	22,697	certain	3,345	'un	227	overmuch	0
any	22,666	anyone	3,318	ye	192	such and such	0
more	21,873	whom	3,229	thy	191	whate'er	0
many	17,343	enough	3,197	whereby	176	whenever	0
such	16,880	half	3,065	thee	166	whereof	0
those	15,819	few	2,933	yourselves	148	whereto	0
own	15,741	everyone	2,812	latter	142	whereunto	0
us	15,724	whatever	2,571	whichever	121	whichsoever	0

Prepositions

Prepositions occur before noun phrases.

Semantically, they are relational.

- spatial (*on*)
- temporal (*after*)
- . . .

of	540,085	through	14,964	worth	1,563	pace	12
in	331,235	after	13,670	toward	1,390	nigh	9
for	142,421	between	13,275	plus	750	re	4
to	125,691	under	9,525	till	686	mid	3
with	124,965	per	6,515	amongst	525	o'er	2
on	109,129	among	5,090	via	351	but	0
at	100,169	within	5,030	amid	222	ere	0
by	77,794	towards	4,700	underneath	164	less	0
from	74,843	above	3,056	versus	113	midst	0
about	38,428	near	2,026	amidst	67	o'	0
than	20,210	off	1,695	sans	20	thru	0
over	18,071	past	1,575	circa	14	vice	0

Conjunctions

Conjunctions join two things.

- coordinating (*and, or, but*): things are of equal stature
- subordinating (*that, if, because, although*): one element has an embedded status
- ...

and	514,946	yet	5,040	considering	174	forasmuch as	0
that	134,773	since	4,843	lest	131	however	0
but	96,889	where	3,952	albeit	104	immediately	0
or	76,563	nor	3,078	providing	96	in as far as	0
as	54,608	once	2,826	whereupon	85	in so far as	0
if	53,917	unless	2,205	seeing	63	inasmuch as	0
when	37,975	why	1,333	directly	26	insomuch as	0
because	23,626	now	1,290	ere	12	insomuch that	0
so	12,933	neither	1,120	notwithstanding	3	like	0
before	10,720	whenever	913	according as	0	neither nor	0
though	10,329	whereas	867	as if	0	now that	0
than	9,511	except	864	as long as	0	only	0
while	8,144	till	686	as though	0	provided that	0
after	7,042	provided	594	both and	0	providing that	0
whether	5,978	whilst	351	but that	0	seeing as	0
for	5,935	suppose	281	but then	0	seeing as how	0
although	5,424	cos	188	but then again	0	seeing that	0
until	5,072	supposing	185	either or	0	without	0

Articles

Articles (determiners) begin noun phrases, and thus help identify them.

- articles: *the* (definite), *a*, *an* (indefinite)
- demonstratives: *this*, *that*

A particularly small class, but also very frequent.

- *the* 1,071,676
- *a* 413,887
- *an* 59,359
- sometimes others

Other Parts of Speech

Prepositions vs. Particles

- *She ran up a hill/bill.*

Phrasal Verbs

- *The plane took off./Take it off.*

Interjections

- *Ouch!*

Penn Treebank Tagset Formalization

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	“to”	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential ‘there’	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	*$*
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	*#*
PDT	Predeterminer	*all, both*	“	Left quote	(‘ or “)
POS	Possessive ending	*’s*	”	Right quote	(’ or ”)
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one’s*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

How are nouns, verbs, adjectives, adverbs, pronouns, prepositions, conjunctions, and articles formalized?

How would you tag the following sentence from the Brown Corpus?

- The grand jury commented on a number of other topics.

Penn Treebank (continued)

Tag	Description	Example	Tag	Description	Example
CC	Coordin. Conjunction	*and, but, or*	SYM	Symbol	+,%, &
CD	Cardinal number	*one, two, three*	TO	“to”	*to*
DT	Determiner	*a, the*	UH	Interjection	*ah, oops*
EX	Existential ‘there’	*there*	VB	Verb, base form	*eat*
FW	Foreign word	*mea culpa*	VBD	Verb, past tense	*ate*
IN	Preposition/sub-conj	*of, in, by*	VBG	Verb, gerund	*eating*
JJ	Adjective	*yellow*	VBN	Verb, past participle	*eaten*
JJR	Adj., comparative	*bigger*	VBP	Verb, non-3sg pres	*eat*
JJS	Adj., superlative	*wildest*	VBZ	Verb, 3sg pres	*eats*
LS	List item marker	*1, 2, One*	WDT	Wh-determiner	*which, that*
MD	Modal	*can, should*	WP	Wh-pronoun	*what, who*
NN	Noun, sing. or mass	*llama*	WP$	Possessive wh-	*whose*
NNS	Noun, plural	*llamas*	WRB	Wh-adverb	*how, where*
NNP	Proper noun, singular	*IBM*	$	Dollar sign	$
NNPS	Proper noun, plural	*Carolinas*	#	Pound sign	#
PDT	Predeterminer	*all, both*	“	Left quote	(‘ or “)
POS	Possessive ending	*’s*	”	Right quote	(’ or ”)
PP	Personal pronoun	*I, you, he*	(	Left parenthesis	([, (, {, <)
PP$	Possessive pronoun	*your, one’s*	)	Right parenthesis	( ],), }, >)
RB	Adverb	*quickly, never*	,	Comma	,
RBR	Adverb, comparative	*faster*	.	Sentence-final punc	(. ! ?)
RBS	Adverb, superlative	*fastest*	:	Mid-sentence punc	(: ; ... – -)
RP	Particle	*up, off*			

Tagged sentence from Brown Corpus

- The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

UCREL C5 Tagset Fragment

Tag	Description	Example
PNX	reflexive pronoun	*itself, ourselves*
POS	possessive *'s* or *'*	
PRF	the preposition *of*	
PRP	preposition (except *of*)	*for, above, to*
PUL	punctuation – left bracket	(or [
PUN	punctuation – general mark	. ! , : ; - ? ...
PUQ	punctuation – quotation mark	‘ ’ ”
PUR	punctuation – right bracket	) or]
TO0	infinitive marker *to*	
UNC	unclassified items (not English)	
VBB	base forms of *be* (except infinitive)	*am, are*
VBD	past form of *be*	*was, were*
VBG	-ing form of *be*	*being*
VBI	infinitive of *be*	
VBN	past participle of *be*	*been*
VBZ	-s form of *be*	*is, 's*
VDB	base form of *do*(except infinitive)	*does*
VDD	past form of *do*	*did*
VDG	-ing form of *do*	*doing*
VDI	infinitive of *do*	*to do*
VDN	past participle of *do*	*done*
VDZ	-s form of *do*	*does*
VHB	base form of *have* (except infinitive)	*have*
VHD	past tense form of *have*	*had, 'd*
VHG	-ing form of *have*	*having*
VHI	infinitive of *have*	
VHN	past participle of *have*	*had*
VHZ	-s form of *have*	*has, 's*
VM0	modal auxiliary verb	*can, could, will, 'll*
VVB	base form of lexical verb (except infin.)	*take, live*
VVD	past tense form of lexical verb	*took, lived*
VVG	-ing form of lexical verb	*taking, living*
VVI	infinitive of lexical verb	*take, live*
VVN	past participle form of lex. verb	*taken, lived*
VVZ	-s form of lexical verb	*takes, lives*
XX0	the negative *not* or *n't*	
ZZ0	alphabetical symbol	*A, B, c, d*

Part of Speech Tagging

Part of speech tagging is simply assigning the correct part of speech for each word in a corpus.

Input

- a set of tags (a tagset)
- a dictionary that tells you the possible tags for each word (including all morphological variants)
- a text to be tagged (string of words)

Output

- single best tag for each word
- e.g., Book/VB that/DT flight/NN

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 DT or complementizer
  - *My travel agent said that there would be a meal on this flight.*

Outline

Last Time

- Parts of Speech
- English Parts of Speech

Today: Tagging

- Tagging with Rules
- Transformation-Based Tagging
- Statistical Tagging

How Hard is the Tagging Problem?

Recall that part of speech tagging assigns the correct part of speech for each word in a corpus.

11.5% of English words in the Brown corpus are ambiguous

But 40% of tokens in the Brown corpus are ambiguous

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")

The Brown Corpus

The Brown Corpus of Standard American English was the first of the modern, computer readable general corpora. Compiled by W. N. Francis and H. Kucera, Brown University, Providence, RI.

Corpus consists of one million words of American English texts 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 . . . ?

There is also a tagged version of the Brown Corpus

`http://www.hit.uib.no/icame/brown/bcm.html`

The Brown Corpus (cont.)

The corpus consists of 500 texts, each consisting of just over 2000 words, compiled from 15 different text categories

1. press (reportage) (44 texts)
2. press (editorial) (27 texts)
3. press (reviews) (17 texts)
4. religion (17 texts)
5. skills and hobbies (36 texts)
6. popular lore (48 texts)
7. belles-lettres (75 texts)
8. miscellaneous: govt. and house organs (30 texts)
9. learned (80 texts)
10. fiction (general) (29 texts)
11. fiction (mystery) (24 texts)
12. fiction (science) (6 texts)
13. fiction (adventure) (29 texts)
14. fiction (romance) (29 texts)
15. humor (9 texts)

Tagging with Rules

The rule-based approach uses *handcrafted* sets of rules to tag input sentences.

- e.g., if input follows a determiner, tag it as a noun.

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

A two-stage architecture

- use lexicon FST (dictionary) to tag each word with all possible POS
- apply rules to eliminate tags

Thus, these rules eliminate tags that are inconsistent with context, and should reduce the list of POS tags to a single POS per word.

This approach does work and produces accurate results.

What are the drawbacks?

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 is to 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 (a supervised method).

Brill Tagging (continued)

Example

1. is expected to race tomorrow
2. the race for outer space

Tag all uses of "race" as nouns (the most likely tag in the tagged Brown corpus).

1. is expected to race/NN tomorrow
2. the race/NN for outer space

Then use a transformation rule to replace that tag with one for verb for all "race" uses that are preceded by the tag TO.

1. is expected to race/VB tomorrow (correct)
2. the race/NN for outer space (correct)

What about

- drawing the district line according to race

Brill Tagging (continued)

Assume some tagged training corpus.

1. Tag the corpus with the most likely tag for each word (unigram model).
2. Choose a transformation that deterministically replaces an existing tag with a new tag such that the resulting tagged training corpus has the lowest error rate out of all transformations.
3. Apply that transformation to the training set.
4. Iterate.
5. Return as your tagger one that first tags using unigrams and then applies the learned transformations in order.

Brill Tagging (continued)

Templates (abstracted transformations) are used to keep the number of transformations finite.

- each begins with "Change tag $\mathbf{a}$ to tag $\mathbf{b}$ when ..." where variables range over the POS tags

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**.

```
function TBL(corpus) returns transforms-queue
 INTIALIZE-WITH-MOST-LIKELY-TAGS(corpus)
 until end condition is met do
   templates ← GENERATE-POTENTIAL-RELEVANT-TEMPLATES
   best-transform ← GET-BEST-TRANSFORM(corpus, templates)
   APPLY-TRANSFORM(best-transform, corpus)
   ENQUEUE(best-transform-rule, transforms-queue)
 end
 return(transforms-queue)
```

```
function GET-BEST-TRANSFORM(corpus, templates) returns transform
  for each template in templates
    (instance, score) ← GET-BEST-INSTANCE(corpus, template)
    if (score > best-transform.score) then best-transform ← (instance, score)
  return(best-transform)
```

```
function GET-BEST-INSTANCE(corpus, template) returns transform
  for from-tag ← from tag−1 to tag−n do
    for to-tag ← from tag−1 to tag−n do
      for pos ← from 1 to corpus-size do
        if (correct-tag(pos) == to-tag && current-tag(pos) == from-tag)
              num-good-transforms(current-tag(pos−1))++
        elseif (correct-tag(pos)==from-tag && current-tag(pos)==from-tag)
              num-bad-transforms(current-tag(pos−1))++
      end
      best-Z ← ARGMAX_t(num-good-transforms(t) - num-bad-transforms(t))
      if(num-good-transforms(best-Z) - num-bad-transforms(best-Z)
                 > best-instance.Z) then
        best-instance ← "Change tag from from-tag to to-tag
                               if previous tag is best-Z"
 return(best-instance)
```

```
procedure APPLY-TRANSFORM(transform, corpus)
for pos ← from 1 to corpus-size do
  if (current-tag(pos)==best-rule-from)
        && (current-tag(pos−1)==best-rule-prev))
    current-tag(pos) = best-rule-to
```

Example Learned Transformations

#	Change tags From	To	Condition	Example
1	NN	VB	Previous tag is TO	to/TO race/NN → VB
2	VBP	VB	One of the previous 3 tags is MD	might/MD vanish/VBP → VB
3	NN	VB	One of the previous 2 tags is MD	might/MD not reply/NN → VB
4	VB	NN	One of the previous 2 tags is DT	
5	VBD	VBN	One of the previous 3 tags is VBZ	

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 taq sequence that maximizes

$$P(word \mid tag) * P(tag \mid previousntags)$$

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

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

Statistical Tagging (cont.)

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

$$t_i = argmax_j P(t_j \mid t_{i-1}) P(w_i \mid t_j)$$

- the function $argmax_x f(x)$ means "the x such that f(x) is maximized"
- the first P is the tag sequence probability, the second the word likelihood

Most of the better statistical models report around 95% accuracy on standard datasets (often the Brown corpus).

But note that you get 91% accuracy just by picking the most likely tag (more on this later).

A Simplifying Example

Examples 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 some other mechanism has already tagged the surrounding words, and we're trying to tag the untagged word *race*.

Bigram tagger assumes we are given just the subsequences

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

Example (cont.)

Goal: choose between NN and VB for the sequence *to race*

Plug these into our bigram HMM tagging equation $P(tag \mid previoustag) * P(word \mid tag)$

- $P(VB \mid TO) * P(race \mid VB)$
- $P(NN \mid TO) * P(race \mid NN)$

So, we need to compute these tag sequence probabilities and the word likelihoods

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:

- $P(VB \mid TO) = .340$
- $P(NN \mid TO) = .021$

Word Likelihood

We must compute the likelihood of the noun $race$ given each tag. ie., $P(race \mid VB)$ and $P(race \mid NN)$

Note, we are not asking which is the most likely tag for this 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:

- $P(race \mid VB) = .00003$
- $P(race \mid NN) = .00041$

And the Winner is . . .

Multiplying tag sequence probabilities by word likelihoods gives

- $P(VB \mid TO) * P(race \mid VB) = .000010$
- $P(NN \mid TO) * P(race \mid NN) = .000007$

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

Review

We looked at a simple case of Statistical Tagging, the bigram HMM tagger:

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

What we really want is to find the best sequence of tags for an entire sentence (sequence of words).

Hidden Markov Model (HMM) Taggers compute the most probable sequence of tags T given a sequence of words W:

$$argmax P(T \mid W)$$

Statistical Tagging (cont.)

To do this requires new machinery

- weighted automata or probabilistic FSA
- in particular, Markov Chain special case and Hidden Markov Models
- details in Chapters 5 and 7 of J&M

Unsupervised HMM Training

What if you have the following, but not a tagged training corpus?

- the tags
- the set of possible words
- the correct set of tags for each word

Use an *unsupervised* approach.

Nobody really does this for tagging anymore.

- it does work
- but, even a tiny amount of smoothed hand-tagged data results in better performance than a purely unsupervised HMM approach

Other Issues

Multiple Tags

Multiple and Split Words

Unknown Words

Class-based N-grams

- can use POS results to augment our previous N-grams

Evaluation

How do you know how well you've done?

- use a tagged test corpus

How do you know if you're doing well?

Doing ok?

Doing great?

Making stupid claims?

A Common Evaluation Metric

Percent Correct

- the percentage of all tags in the test set where the tagger and a human labeled "gold standard" agree
- typically 96-97% for POS tagging

Evaluation Issues

Lower Bound/Baseline

- your goodness metric has to take into account some baseline that any dumb approach could achieve
- in the case of POS tagging, this is the most frequent tag heuristic (unigrams)
- can get you 90% for POS!
- thus, a result of 99% for POS tagging is less impressive than for say speech act tagging, where the baseline is much lower

Upper Bound/Ceiling

- this is all dependent on how good the human taggers did their job
- if there's a 3% error in your training corpus, then reporting 99% accuracy will make you look silly

Another Evaluation Metric

Agreement via Kappa

- adjusts for a baseline and thus normalizes for task difficulty
- the ratio of the proportion of times that two classifiers agree (corrected for chance agreement) to the maximum proportion of times that the classifiers could agree (corrected for chance agreement)
- $$\frac{P(A) - P(E)}{1 - P(E)}$$
- P(A) = percent correct
- P(E) = expected agreement (by chance)

Kappa is also useful for evaluating human labelers without a gold standard.

Error Analysis

Once you have some results, you can improve your tagging scheme by doing error analysis.

Contingency Tables/Confusion Matrices are a way of visualizing what went wrong, where

- usually a table of raw counts
- rows indicate the correct tag
- columns indicate the output of the tagger
- $cell(row_i, column_j)$ contains the number of times that an item with the correct class x was classified as class y

Example

```
- Tagger Output
- A B C
A 0 0 5
B 0 1 4
C 0 7 1
```

- A was mistagged as C 5 times (i.e., 5 times when the tagger proposed C, it should have been a A)
- B ...
- C ...
- error analysis suggests that you might want to
  - merge tags B and C,
  - rewrite the coding instructions

Kappa and Confusion Matrices

Consider an example confusion matrix for a tagger

-	A	B	C
A	25	0	0
B	0	10	4
C	0	7	5

Recall that P(A) is the proportion of times the tagging hypothesis agrees with the standard (percent correct)

Computing P(A) is easy

- $P(A) = \frac{25+10+5}{25+10+5+4+7} = .78$

Computing P(E)

Recall that P(E) is the proportion of times the tagging hypothesis and the standard would be expected to agree by chance

This is computed as the sum of the probabilities of getting each individual tag right

- P(E) $= \Sigma_T P(E_T)$

The expected probability for a tag T, $P(E_T)$, is the product of the proportion of the time the tagger selected that tag, and the proportion that the tag occurs in the standard.

- $P(E_A) = \frac{25}{51} * \frac{25}{51} = .24$
- $P(E_B) = \frac{10+7}{51} * \frac{10+4}{51} = .092$
- $P(E_C) =? = .041$

So $P(E) = P(E_A) + P(E_B) + P(E_C)$

Also see exercise 8.4 in text.

Confusion Matrices (cont.)

The textbook uses a slightly different representation, where the values of the cels represent the overall tagging error.

For example,

```
- A B C
A 25 0 0
B 0 10 4
C 0 7 5
```

becomes

```
- A B C
A 0 0 0
B 0 0 36.4
C 0 63.6 0
```

So, 36.4% of the tagging errors involved mistagging a B as a C.

This is how the program in our homework works as well.

For Next Time

Discussion Paper

Homework 2

Chapter 9

Extra Notes on Statistical Tagging

More Probability: Bayes

Product rule

$$P(A \wedge B) = P(A \mid B)P(B) = P(B \mid A)P(A)$$

gives *Bayes Rule*

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

Why is this useful?

We substitute computing something we can't calculate for things that are easier to calculate

$$P(cause \mid effect) = \frac{P(effect \mid cause)P(cause)}{P(effect)}$$

For example, meningitis could be a cause, and stiff neck an effect.

The key point of Bayes Rule is that we can move from $P(A \mid B)$ to $P(B \mid A)$ and back given appropriate information.

Bayes Rule and the Noisy Channel

For our NLP noisy channel applications, we want to compute the most probable source given an observed signal

$$argmax_{source} P(source \mid signal)$$

Unfortunately we don't usually know how to compute this.

Rewriting as

$$P(source \mid signal) = \frac{P(signal \mid source)P(source)}{P(signal)}$$

gives

$$argmax_{source} \frac{P(signal \mid source)P(source)}{P(signal)}$$

How does this help if all we have is the signal?

We *do* know the set of all possible sources. So, we can plug each into the equation one by one and compute their probabilities using this equation. The source hypothesis with the highest probability wins.

Returning to Tagging

Hidden Markov Model (HMM) Tagger

$$argmax P(TagSequence \mid WordSequence)$$

- again, argmax means the Tag Sequence such that P is maximized

Rewriting this using *Bayes Rule* yields

$$\frac{P(WordSequence \mid TagSequence) P(TagSequence)}{P(WordSequence)}$$

Statistical Tagging

If we assume a tagged corpus and a trigram language model, then $P(T)$ can be approximated as

$$P(t_1)P(t_2 \mid t_1)\Pi_{i=3}^{n}P(t_i \mid t_{i-2}t_{i-1})$$

using counting and smoothing.

If we make the simplifying assumption that each word in the sequence depends only on its tag, then $P(W \mid T)$ can be approximated as

$$\Pi_{i=1}^{n}P(w_i \mid t_i)$$