CS395-T
Topics in Natural Language Processing

LECTURE 2

Sanda Harabagiu
Department of Computer Sciences
University of Texas at Austin
Tu, Th 11AM-12:20 PM
sanda@cs.utexas.edu
http://www.cs.utexas.edu/users/sanda/cs395T.html
http://www.engr.smu.edu/~sanda/cs395T.html

Part-of-Speech Tagging

- What is PoS tagging?
- Associating with any word a lexical tag
  => word classes
- 45 classes for the Penn Treebank
- 87 classes for the Brown Corpus
- 146 classes for the C7 tagset
  (the CLAWS system)

English Word Classes

Parts of speech:

Closed classes:

- prepositions: on, under, over, at, from, etc.
- determiners: a, an, the
- pronouns: she, who, I, others
- conjunctions: and, but, or
- auxiliary verbs: can, may, should, are
- particles: up, down, on, off, in, out
- numerals: one, two, first, second

Open classes:

- nouns
- verbs
- adjectives
- adverbs

3

Prepositions from CELEX

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

Frequency counts are from the COBUILD 16 million word corpus

4

Particles

- words that resemble a preposition or an adverb, often combined with a verb to form a larger unit called a phrasal verb.

aboard	aside	besides	forward(s)	opposite	through
about	astray	between	home	out	throughout
above	away	beyond	in	outside	together
across	back	by	inside	over	under
ahead	before	close	instead	overhead	underneath
alongside	behind	down	near	past	up
apart	below	east, etc	off	round	within
around	beneath	eastward(s), etc	on	since	without

5

Conjunctions from CELEX

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	none 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

Frequency counts are from the COBUILD 16 million word corpus

6

Pronouns from CELEX

it	199,920	me	43,071	how	13,137	most	5,115
I	198,139	who	42,881	another	12,551	itself	5,032
he	158,366	them	42,099	where	11,857	myself	4,819
you	128,688	no	33,458	same	11,841	everything	4,662
his	99,820	some	32,863	something	11,754	several	4,306
they	88,416	other	29,391	each	11,320	less	4,278
this	84,927	your	28,923	both	10,930	herself	4,016
that	82,603	its	27,783	last	10,816	whose	4,005
she	73,966	our	23,029	every	9,788	someone	3,755
her	69,004	these	22,697	himself	9,113	certain	3,345
we	64,846	any	22,666	nothing	9,026	anyone	3,318
all	61,767	more	21,873	when	8,336	whom	3,229
which	61,399	many	17,343	one	7,423	enough	3,197
their	51,922	such	16,880	much	7,237	half	3,065
what	50,116	those	15,819	anything	6,937	few	2,933
my	46,791	own	15,741	next	6,047	everyone	2,812
him	45,024	us	15,724	themselves	5,990	whatever	2,571

Frequency counts are from the COBUILD 16 million word corpus

7

Pronouns from CELEX

yourself	2,437	hers	482	no one	106	howsoever	1
why	2,220	ours	458	wherein	58	thrice	1
little	2,089	whoever	391	double	39	wheresoever	1
none	1,992	least	386	thine	30	you-all	1
nobody	1,684	twice	382	summat	22	additional	0
further	1,666	theirs	303	suchlike	18	anybody	0
everybody	1,474	wherever	289	fewest	15	each other	0
ourselves	1,428	oneself	239	thyself	14	once	0
mine	1,426	thou	229	whomever	11	one another	0
somebody	1,322	'un	227	whosoever	10	overmuch	0
former	1,177	ye	192	whomsoever	8	such and such	0
past	984	thy	191	wherefore	6	whate'er	0
plenty	940	whereby	176	whereat	5	whenever	0
either	848	thee	166	whatsoever	4	whereof	0
yours	826	yourselves	148	whereon	2	whereto	0
neither	618	latter	142	whoso	2	whereunto	0
fewer	536	whichever	121	aught	1	whichsoever	0

Frequency counts are from the COBUILD 16 million word corpus

8

Modal Verbs from CELEX

can	70,930	might	5,580	shouldn't	858
will	69,206	couldn't	4,265	mustn't	332
may	25,802	shall	4,118	'll	175
would	18,448	wouldn't	3,548	needn't	148
should	17,760	won't	3,100	mightn't	68
must	16,520	'd	2,299	oughtn't	44
need	9,950	ought	1,845	mayn't	3
can't	6,375	will	862	dare	??
have	???				

Frequency counts are from the COBUILD 16 million word corpus

9

Tagsets for English

Tag	**Description**	**Example**	**Tag**	**Description**	**Example**
CC	Coordin. Conjunction	*and, but, or*	NNS	Noun, plural	*llamas*
CD	Cardinal number	*one, two, three*	NNP	Proper noun, singular	*IBM*
DT	Determiner	*a, the*	NNPS	Proper noun, plural	*Carolinas*
EX	Existential 'there'	*there*	PDT	Predeterminer	*all, both*
FW	Foreign word	*mea culpa*	POS	Possesive ending	*'s*
IN	Preposition/sub-conj	*of, in, by*	PP	Personal pronoun	*I, you, he*
JJ	Adjective	*yellow*	PP$	Possesive pronoun	*your, one's*
JJR	Adjective, comparative	*bigger*	RB	Adverb	*quickly, never*
JJS	Adjective, superlative	*wildest*	RBR	Adverb, comparative	*faster*
LS	List item marker	*1, 2, One*	RBS	Adverb, superlative	*fastest*
MD	Modal	*can, should*	RP	Particle	*up, off*
NN	Noun, singular or mass	*llama*	SYM	Symbol	*+, %, &*

10

Tagsets for English

Tag	Description	Example	Tag	Description	Example
TO	"to"	*to*	WRB	Wh-adverb	*how, where*
UH	Interjection	*ah, oops*	$	Dollar sign	*$*
VB	Verb, base form	*eat*	#	Pound sign	*#*
VBD	Verb, past tense	*ate*	"	Left quote	*' or "*
VBG	Verb,gerund	*eating*	"	Right quote	*' or "*
VBN	Verb, past participle	*eaten*	(	Left parenthesis	*[, (, {, <*
VBP	Verb, non-3sg. present	*eat*	)	Right parenthesis	*],), }, >*
VBZ	Verb, 3sg. present	*eats*	,	Comma	*,*
WDT	Wh-determiner	*which, that*	.	Sentence-final punc	*. ! ?*
WP	Wh-pronoun	*what, who*	:	Mid-sentence punc	*: ; … -- -*
WP$	Possesive wh-pronoun	*whose*			

11

PoS-Tagging

We used:

- A set of tags
- A dictionary that indicates all possible tags for each word.
- Input text
- General purpose vs. special purpose

VB	*DT*	*NN*	.		
Book	*that*	*flight*	.		
VBZ	*DT*	*NN*	*VB*	*NN*	*?*
Does	*that*	*flight*	*serve*	*dinner*	*?*

Methods:

- Rule-based tagging
- Stochastic tagging
- Transformation-based tagging

12

How hard is PoS ?

- Most words are unambiguous
- Many of the most common English words are ambiguous
  - 11.5% of the words in the Brown corpus are ambiguous
  - 40% of Brown tokens 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	
7tags	1	("still")

The number of word types in Brown corpus by degree of ambiguity

13

Rule-based Tagging

A two stage architecture

(Harris '62, Klein & Simmons '63, Greene & Rubin '71)

a) Use a dictionary to assign to each word **a list** of potential POS

b) Use **large lists** of hand-written disambiguation rules

Example:

The ENGTWOL tagger (Vontilainen, 1995)

56000 of English word stems.

14

Sample Lexical Entries from ENGTWOL

Word	POS	Additional POS features
smaller	ADJ	COMPARATIVE
entire	ADJ	ABSOLUTE ATTRIBUTIVE
fast	ADV	SUPERLATIVE
that	DET	CENTRAL DEMONSTRATIVE SG
all	DET	PREDETERMINER SG/PL QUANTIFIER
dog's	N	GENITIVE SG
furniture	N	NOMINATIVE SG NONINDEFDETERMINER
one-third	NUM	SG
she	PRON	PERSONAL FEMININE NOMINATIVE SG3
show	V	IMPERATIVE VFIN
show	V	PRESENT -SG3 VFIN
show	N	NOMINATIVE SG
shown	PCP2	SVOO SVO SV
occurred	PCP2	SV
occurred	V	PAST VFIN SV

15

Rule-based Tagging

- Step 1 – assign to each word a list of potential parts-of-speech.
- Step 2 – use if-then rules to pinpoint the correct tag for each word.

Example:

Pavlov had shown that salivation

Step 1:

Pavlov	**PAVLOV N NOM SG PROPER**
had	**HAVE V PAST VFIN SVO**
	HAVE PCP2 SVO
shown	**SHOW PCP2 SVOO SVO SV**
that	ADV
	PRON DEM SG
	DET CENTRAL DEM SG
	CS
salivation	**N NOM SG**

Step 2:

```
ADVERBIAL-THAT RULE
Given input: "that"
if
   (+1 A/ADV/QUANT); /* if next word is adj, adverb, or quantifier */
   (+2 SENT-LIM);    /* and following which is a sentence boundary, *
   (NOT -1 SVOC/A); /* 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
```

- Advantage: high precision (99%)
- Disadvantage: needs lots of rules

Stochastic POS Tagging

- Old idea of using probabilities in tags
  - (Stolz '65)
  - Bahl and Mercer '76 - complete probabilistic tagger with Viterbi decoding
  - In the 80s: Marshall '83, Garside '87, Church '88, De Rose '88
- Hidden Markov Model

17

Hidden Markov Model

- For a given sentence or word sequence HMM taggers choose the tag sequence that maximizes:

  P(word|tag)*P(tag|previous n tags)
- Given a word w_i, a bigram-HMM tagger chooses the tag t_i:

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

- This can be restated as:

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

18

Statistical POS Tagging

Example:

Suppose w_i = race, that is a verb (VB) or a noun (NN)

Assume that other mechanism has already don the best tagging to the surrounding words, leaving only **race** untagged

1) Secretariat/NNP is/VBZ expected/VBN to/TO race/VB tomorrow/NN
2) People/NNS continue/VBP to/TO inquire/VB the/DT reason/NN for/IN the/DT race/NN for/IN outer/JJ space/NN

Simplifying the problem:
to/To race/???
The/DT race/???

$$t_i = \underset{j}{\text{argmax}}\ P(t_j|t_{i-1})P(w_i|t_j)$$

choose between

$P(VB|TO)\quad P(race|VB) =$
and
$P(NN|TO)\quad P(race|NN) =$

19

Where is the DATA ????

Look at the Brown and Switchboard corpora:

$P(NN|TO) = 0.021$
$P(VB|TO) = 0.34$

What about P(race|VB)
P(race|NN)

"If we are expecting a verb, how likely is it that the verb would be race ?"

$P(race|NN) = 0.00041$
$P(race|VB) = 0.00003$

Finally:

$P(VB|TO)\ P(race|VB) = 0.00001$
$P(NN|TO)\ P(race|NN) = 0.000007$

20

The Actual Algorithm for HMM tagging

The HMM-tagger isn't just choosing the best tag for a single word, but the best sequence of tags for an entire sentence !

Viterbi approximation: chose the most probable sequence for each sentence

Given the sentence S = w_1 w_2 … w_n

Compute the most probable sequence of tags

T = t_1, t_2, … ,t_n

$$\hat{T} = \text{argmax} \; P(T)P(W|T)$$

21

Stochastic POS Tagging

Bayes Law: $P(T|W) = \frac{P(T)P(w|T)}{P(w)}$

The probability P(w) will be the same for each tag sequence => ignore it !

$$\hat{T} = \underset{T \in \tau}{\text{argmax}} \; P(T)P(W|T)$$

From the chain rule of probability:

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

$$P(t_i | t_{i-2} t_{i-1}) = \frac{C(t_{i-2} t_{i-1} t_i)}{C(t_{i-2} t_{i-1})} \qquad P(\text{Tag Sequence}) = P(t_1)P(t_2|t_1)\prod_{i=3}^{n} P(t_i | t_{i-2} t_{i-1})$$

22

HMM Model Continued

- Simplifying assumption:
  - the probability of a word depends only on its tag

$$P(w_i \mid w_1 t_1 ... w_{i-1} t_{i-1} t_i) = p(w_i \mid t_i)$$

- Second assumption:
  - the tag history can be approximated by the most recent two tags:

$$P(w_i \mid w_1 t_1 ... w_{i-1} t_{i-1}) = p(t_i \mid t_{i-2} t_{i-1})$$

23

What Is Needed ?

- Choose the tag sequence that maximizes:

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

How ?

Use maximum likelihood estimation from relative frequencies to estimate these probabilities:

$$P(t_i \mid t_{i-2} t_{i-1}) = \frac{C(t_{i-2} t_{i-1} t_i)}{C(t_{i-2} t_{i-1})}$$

$$P(w_i \mid t_i) = \frac{C(w_i, t_i)}{C(t_i)}$$

24

Additional Work

- The model can be smoothed (e.g. backoff or deleted interpolation algorithms) to avoid zero probabilities
- The most probable tag sequence ?
  - Viterbi algorithm, Accuracy = 96% (Weischedel '93, De Rose '88)

25

Transformation-based Tagging

- The pure rule-based approach is too expensive, slow, tedious.
- Brill's Transformation-Based Learning (TLB)
- Basic idea is to do a poor job first, and then use learned rules to improve things.

Example: tag "race"
Step 1

$$P(race \mid NN) = 0.98$$
$$P(race \mid VB) = 0.02$$

Tag all uses of race as nouns
is/VBZ expected/VBN to/TO race/NN tomorrow/NN

the/DT race/NN for/IN outer/JJ space/NN

26

Brill Tagging

Step 2
Rule: Change NN to VB when the previous tag is TO.
Assume some tagged training corpus.
Make the assumption that the word depends only on its tag.

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.

Transformation-Based Tagging

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**.

Brill's (1995) templates. Each begins with "Change tag a to tag b when: …". The variables a, b, z, and w range over parts of speech.

28

The First 5 non-lexicalized 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	

29

The TBL Algorithm

- 2 functions: GET_Best_Transform
  GET_Best_Instance
- GET_Best_Transform
  - is called with a list of potential templates
  - it calls GET_Best_Instance
- GET_Best_Instance iteratively tests every possible instantiation of each template by filling specific values for the tag variables a,b,z and w

30

The Code

```
function TBL(corpus) returns transforms-queue
  INITIALIZE-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, transform-queue)
  end
return(transforms-queue)

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

31

More Code

```
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)
```

32

More Code

```
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
```

33