Natural Language Processing

Assist. Prof. Dr. Tuğba YILDIZ

İSTANBUL BİLGİ UNIVERSITY
Department of Computer Engineering

March 31, 2017

Word Classes and Part of Speech Tagging

- words are traditionally grouped into equivalence classes called
  - parts of speech (POS)
  - word classes
  - morphological classes
  - lexical tags

Word Classes and Part of Speech Tagging

- in traditional grammars, there were generally only a few parts of speech (noun, verb, adjective, preposition, adverb, conjunction, etc.).
- more recent models have much larger numbers of word classes:
  - 45 for the Penn Treebank (Marcus et al., 1993)
  - 87 for the Brown corpus (Francis, 1979; Francis and Kucera, 1982)
  - 146 for the C7 tagset (Garside et al.,1997)

Word Classes and Part of Speech Tagging

- the POS for a word gives a significant amount of information about the word and its neighbors.
- for example: possessive pronouns (my, your, his, her, its) and personal pronouns (I, you, he, me).
- knowing whether a word is a possessive pronoun or a personal pronoun can tell us what words are likely to occur in its vicinity
- possessive pronouns are likely to be followed by a noun, personal pronouns by a verb
- this can be useful in a language model for speech recognition

Word Classes and Part of Speech Tagging

- A word's part-of-speech can tell us something about how the word is pronounced.
- OBject (noun) and obJECT (verb)
- DIScount (noun) and disCOUNT (verb)
- CONtent (noun) and the conTENT (adjective)

Word Classes and Part of Speech Tagging

- parts of speech can be divided into two broad supercategories: closed class types and open class types
- prepositions are a closed class because there is a fixed set of them in English; new prepositions are rarely coined.
- by contrast nouns and verbs are open classes because new nouns and verbs are continually coined or borrowed from other languages
- closed class words are generally also function words
- function words are grammatical words like **of, it, and** which tend to be very short, occur frequently, and play an important role in grammar.

Word Classes and Part of Speech Tagging

- there are four major open classes that occur in the languages of the world: nouns, verbs, adjectives, and adverbs.
- noun is the name given to the lexical class in which the words for most people, places, or things occur
- nouns are traditionally grouped into proper nouns and common nouns
- proper nouns, like Regina, Colorado, and IBM, are names of specific persons or entities

Word Classes and Part of Speech Tagging

- common nouns are divided into count nouns and mass nouns.
- count nouns are those that allow grammatical enumeration (one goat, two goats)
- mass nouns are used when something is conceptualized as a homogeneous group (snow, salt, and communism)

Word Classes and Part of Speech Tagging

- the verb class includes most of the words referring to actions and processes (draw, provide, differ, and go)
- the third open class English form is adjectives
- semantically this class includes many terms that describe properties or qualities
- the final open class form is adverbs
  - directional adverbs or locative adverbs (here, downhill) specify the direction or location of some action
  - degree adverbs (extremely, very, somewhat) specify the extent of some action, process, or property
  - manner adverbs (slowly, slinkily, delicately) describe the manner of some action or process
  - temporal adverbs describe the time that some action or event took place (yesterday, Monday)

Word Classes and Part of Speech Tagging

- Closed class:
- prepositions: on, under, over, near, by, at, from, to, with
- determiners: a, an, the
- pronouns: she, who, I, others
- conjunctions: and, but, or, as, if, when
- auxiliary verbs: can, may, should, are
- particles: up, down, on, off, in, out, at, by,
- numerals: one, two, three, first, second, third

Word Classes and Part of Speech Tagging

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

Prepositions (and particles) of English from the CELEX on-line dictionary. Frequency counts are from the COBUILD 16 million word corpus

Word Classes and Part of Speech Tagging

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

English single-word particles from Quirk et al. (1985a)

Word Classes and Part of Speech Tagging

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

Coordinating and subordinating conjunctions of English from the CELEX on-line dictionary. Frequency counts are from the COBUILD 16 million word corpus.

Word Classes and Part of Speech Tagging

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

Pronouns of English from the CELEX on-line dictionary. Frequency counts are from the COBUILD 16 million word corpus.

Word Classes and Part of Speech Tagging

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	9,955	ought	1,845	mayn't	3
can't	6,375	will	862	dare	??
have	???				

English modal verbs from the CELEX on-line dictionary. Frequency counts are from the COBUILD 16 million word corpus.

Tagsets fo English

- There are a small number of popular tagsets for English
  - 87-tag tagset used for the Brown corpus (Francis, 1979; Francis and Kučera, 1982).
  - the most commonly used are the small 45-tag Penn Treebank tagset (Marcus et al., 1993)
  - the medium-sized 61 tag C5 tagset
  - the larger 146-tag C7 tagset (Leech et al., 1994)

Tagsets fo English

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*			

Penn Treebank for POS Tagging

Tagsets fo English

- The Penn Treebank tagset has been applied to the Brown corpus and a number of other corpora.
    - The/DT grand/JJ jury/NN commented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

Part of Speech Tagging

- Part-of-speech tagging (or just tagging for short) is the process of assigning a part-of-speech or other lexical class marker to each word in a corpus.
- The input to a tagging algorithm is a string of words and a specified tagset
- The output is a single best tag for each word.

VB DT NN .
Book that flight .

VBZ DT NN VB NN ?
Does that flight serve dinner ?

Tagged output

Part of Speech Tagging

- Automatically assigning a tag to each word is not trivial.
- For example, book is ambiguous.
- That is, it has more than one possible usage and part of speech.
- It can be a verb (as in book that flight or to book the suspect)
- or a noun (as in hand me that book, or a book of matches).
- For example, that is ambiguous.
- Similarly that can be a determiner (as in Does that flight serve dinner), or a
- complementizer (as in I thought that your flight was earlier).

Part of Speech Tagging

- The problem of POS-tagging is to resolve these ambiguities, choosing the proper tag for the context.
- Part-of-speech tagging is thus one of the many disambiguation tasks

Part of Speech Tagging

- How hard is the tagging problem?
- Most words in English are unambiguous; i.e. they have only a single tag.
- But many of the most common words of English are ambiguous
- for example can can be an auxiliary ('to be able'), a noun ('a metal container'), or a verb ('to put something in such a metal container')). Brown tokens are ambiguous.

Part of Speech Tagging

- Words often have more than one POS: **back**
- The **back** door = JJ
- On my **back** = NN
- Win the voters **back** = RB
- Promised to **back** the bill = VB
- The POS tagging problem is to determine the POS tag for a particular instance of a word.

Tagsets fo English

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 number of word types in Brown corpus by degree of ambiguity

Methods for POS Tagging

- Three methods:
  - Rule-based tagging
    - ENGTWOL
  - Stochastic (=Probabilistic) tagging
    - HMM (Hidden Markov Model) tagging
  - Transformation-based tagging
    - Brill tagger

Rule-based Tagger

- Rule-based Taggers:
  - The earliest algorithms for automatically assigning part-of-speech were based on a two-stage architecture
  - The first stage used a dictionary to assign each word a list of potential parts of speech.
  - The second stage used large lists of hand-written disambiguation rules to extract a single part-of-speech for each word.

Rule-based Tagger

- Rule-based Taggers:
  - she: PRP
  - promised: VBN,VBD
  - to: TO
  - back: VB, JJ, RB, NN
  - the: DT
  - bill: NN, VB
- Use the dictionary to assign every possible tag
- Write rules to eliminate tags
  - Eliminate VBN if VBD is an option when VBN|VBD follows "<start> PRP"

Rule-based Tagger

- Rule-based Taggers:
  - The ENGTWOL tagger (Voutilainen, 1995) is based on the same two-stage architecture
  - The ENGTWOL lexicon is based on the two-level morphology
  - has about 56,000 entries for English word stems
  - counting a word with multiple parts of speech (e.g. nominal and verbal senses of hit)
  - Each entry is annotated with a set of morphological and syntactic features

Tagsets fo English

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 NOINDEFDETERMINER
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

Sample lexical entries from the ENGTWOL lexicon described in Voutilainen (1995) and Heikkilä (1995).

Rule-based Tagger

- Rule-based Taggers:
  - SG for singular
  - SG3 for other than third-person-singular
  - ABSOLUTE means non-comparative and non-superlative for an adjective
  - NOMINATIVE just means non-genitive
  - PCP2 means past participle

Rule-based Tagger

- Rule-based Taggers:
  - In the fist stage of the tagger, each word is run through the two-level lexicon transducer
  - the entries for all possible parts of speech are returned
  - "Pavlov had shown that salivation"

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**

List for Pavlov has shown that salivation"

Rule-based Tagger

- Rule-based Taggers:
  - a set of about 1,100 constraints are then applied to the input sentence to rule out incorrect parts of speech

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

adverbial rules for that

Probability

- Probability
  - What's the probability of a random word (from a random dictionary page) being a verb?

$$P(\textit{drawing a verb}) = \frac{\textit{number of ways to get a verb}}{\textit{all words}}$$

Probability

- Probability
  - How to compute each of these?
  - All words = just count all the words in the dictionary
  - number of ways to get a verb: number of words which are verbs!
  - If a dictionary has 50000 entries and 10000 are verbs
  - P(V) is $10000/50000 = 1/5 = 0.20$

Probability

- Probability
  - What's the probability of picking two verbs randomly from the dictionary
  - Events are independent, so multiply probs
  - P(w1=V,w2=V) = P(V) * P(V)
  - = 1/5 * 1/5
  - = 0.04
  - What if events are not independent?

Probability

- Conditional probability
  - Written P(A|B)
  - Let's say A is "it's raining"
  - Let's say B is "it was sunny ten minutes ago"
  - P(A|B) means "what is the probability of it raining now if it was sunny 10 minutes ago"
  - P(A|B) is probably way less than P(A)

Probability

- Conditional probability
  - P(Verb) is the probability of a randomly selected word being a verb.
  - P(Verb|race) is "what's the probability of a word being a verb given that it's the word "race"?
  - Race can be a noun or a verb.
  - It's more likely to be a noun.
  - P(Verb|race) can be estimated by looking at some corpus and saying "out of all the times we saw race", how many were verbs?

Probability

- Conditional probability
  - In Brown corpus, P(Verb|race) = 96/98 = .98

$$P(V|race) = \frac{Count(race\ is\ verb)}{total\ Count(race)}$$

Most Frequent Tagger Algorithm

- Most Frequent Tagger
  - Some ambiguous words have a more frequent tag and a less frequent tag:
  - Consider the word "a" in these 2 sentences:
    - would/MD prohibit/VB a/DT suit/NN for/IN refund/NN
    - of/IN section/NN 381/CD (/(a/NN)/) ./.
  - Which do you think is more frequent?

Most Frequent Tagger Algorithm

- Most Frequent Tagger
  - We could count in a corpus
  - A corpus: an on-line collection of text, often linguistically annotated
  - The Brown Corpus: 1 million words from 1961
  - Part of speech tagged at U Penn

number of a	tag
21830	DT
6	NN
3	FW

Most Frequent Tagger Algorithm

- Most Frequent Tagger
  - The Most Frequent Tag algorithm:
  - For each word
    - Create a dictionary with each possible tag for a word
    - Take a tagged corpus
    - Count the number of times each tag occurs for that word
  - Given a new sentence
    - For each word, pick the most frequent tag for that word from the corpus.

Most Frequent Tagger Algorithm

- Most Frequent Tagger
  - The Most Frequent Tag algorithm:
  - Q: Where does the dictionary come from?
  - A: One option is to use the same corpus that we use for computing the tags

Most Frequent Tagger Algorithm

- Most Frequent Tagger
  - The Most Frequent Tag algorithm:
  - Using a corpus to build a dictionary
    - The/DT City/NNP Purchasing/NNP Department/NNP ,/, the/DT jury/NN said/VBD,/, is/VBZ lacking/VBG in/IN experienced/VBN clerical/JJ personnel/NNS . . .
  - From this sentence, dictionary is:
  - clerical
  - department
  - experienced
  - in
  - is
  - jury
  - ...

Most Frequent Tagger Algorithm

- Most Frequent Tagger
  - Evaluating the performance:
  - How do we know how well a tagger does?
  - Say we had a test sentence, or a set of test sentences, that were already tagged by a human (a “Gold Standard”)
  - We could run a tagger on this set of test sentences
  - See how many of the tags we got right.
  - This is called “Tag accuracy” or “Tag percent correct”

Most Frequent Tagger Algorithm

- Most Frequent Tagger
    - Evaluating the performance: Test Set
    - We take a set of test sentences
    - Hand-label them for part of speech
    - The result is a "Gold Standard" test set
    - Who does this?
    - Brown corpus: done by U Penn
    - Grad students in linguistics

Most Frequent Tagger Algorithm

- Most Frequent Tagger
  - Evaluating the performance: Test Set and Train Set
  - But we can't train our frequencies on the test set sentences. (Why?)
  - So for testing the Most-Frequent-Tag algorithm (or any other stochastic algorithm), we need 2 things:
    - A hand-labeled training set: the data that we compute frequencies from, etc
    - A hand-labeled test set: The data that we use to compute our % correct.

Most Frequent Tagger Algorithm

- Most Frequent Tagger
  - Of all the words in the test set
  - For what percent of them did the tag chosen by the tagger equal the human-selected tag.

$$correct = \frac{\{\#\,\text{of}\ words\ \text{tagged}\ correctly\ in\ test\ set}{total\ \#\ of\ words\ in\ test\ set}$$

- Often they come from the same labeled corpus!
- generally, use 90% of the corpus for training and save out 10% for testing!

Most Frequent Tagger Algorithm

- Most Frequent Tagger
- Does the same evaluation metric work for rule-based taggers?
- Yes!
- Rule-based taggers don't need the training set.
- But they still need a test set to see how well the rules are working.

Stochastic Tagging

- Stochastic Tagging
- Based on probability of certain tag occurring given various possibilities
- Necessitates a training corpus
- No probabilities for words not in corpus.
- Training corpus may be too different from test corpus.

Stochastic Tagging

- HMM Tagger
- Intuition: Pick the most likely tag for this word.
- HMM Taggers choose tag sequence that maximizes this formula:
- P(word|tag) x P(tag|previous n tags)
- Let T = t1,t2,...,tn
- Let W = w1,w2,...,wn
- Find POS tags that generate a sequence of words, i.e., look for most probable sequence of tags T underlying the observed words W.

Stochastic Tagging

- HMM Tagger
- argmaxT P(T|W)
- argmaxT P(W|T)P(T) Bayes Rule
- argmaxT P(w1...wn|t1...tn)P(t1...tn)
- Remember, we are trying to find the sequence T that will maximize P(T|W) so this equation is calculated over the whole sentence.
- Assume word is dependent only on its own POS tag: it is independent of the others around it
- argmaxT [P(w1|t1)P(w2|t2)...P(wn|tn)][P(t1)P(t2|t1)...P(tn|tn-1)]

Stochastic Tagging

- Bigram HMM Tagger
- Also assume that probability is dependent only on previous tag
- For each word and possible tag, need to calculate:
- P(ti) = P(wi|ti)P(ti|ti-1)
- then multiply this over each possible tag and each word over the sequence of tags

Stochastic Tagging

- Bigram HMM Tagger
- How do we compute P(ti|ti-1)?
- c(ti-1ti)/c(ti-1)
- How do we compute P(wi|ti)?
- c(wi,ti)/c(ti)
- How do we compute the most probable tag sequence?
- Viterbi algorithm

Stochastic Tagging

- HMM Tagger
- using an HMM tagger to assign the proper tag to the single word race in the following examples
  - 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
- In the first example race is a verb (VB), in the second a noun (NN).

Stochastic Tagging

- HMM Tagger
  - 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
- to/TO race/???
- the/DT race/???
- we choose the tag that has the greater of these two probabilities:
- $P(\text{race} \mid \text{VB})P(\text{VB} \mid \text{TO})$
- $P(\text{race} \mid \text{NN})P(\text{NN} \mid \text{TO})$

Stochastic Tagging

- HMM Tagger
  - ti = argmaxj P(wi|tj)P(tj|t_{j-1})
  - i = num of word in sequence, j = num among possible tags
  - max[P(VB|TO)P(race|VB) , P(NN|TO)P(race|NN)]
  - Brown:
  - P(race|NN) = .00041 X P(NN|TO) = .021 = .000007
  - P(race|VB) = .00003 X P(VB|TO) = .34 = .00001

Stochastic Tagging

- HMM Tagger
  - Generally, we make the Viterbi approximation and choose the most probable tag sequence for each sentence.
  - This approach thus assumes that we are trying to compute for each sentence the most probable sequence of tags
  - T = t1, t2,...tn given the sequence of words in the sentence (W):

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

Stochastic Tagging

- HMM Tagger
  - By Bayes Law, P (T | W) can be expressed as:

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

  - Thus we are attempting to choose the sequence of tags that maximizes

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

Stochastic Tagging

- HMM Tagger
  - Since we are looking for the most likely tag sequence for a sentence given a particular word sequence
  - the probability of the word sequence P (W) will be the same for each tag sequence and we can ignore it.

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

  - From the chain rule of probability:

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

Stochastic Tagging

- HMM Tagger
  - We make the N-gram assumption again for modeling the probability of word sequences.
  - The trigram model is most often used, so let's define that one.
  - First, we make the simplifying assumption that the probability of a word is dependent only its tag:

$$P(w_i|w_1t_1 \ldots w_{i-1}t_{i-1}t_i) = p(w_i|t_i)$$

  - Next, we make the assumption that the tag history can be approximated by the most recent two tags:

$$P(t_i|w_1t_1 \ldots w_{i-1}t_{i-1}) = P(t_i|t_{i-2}t_{i-1})$$

Stochastic Tagging

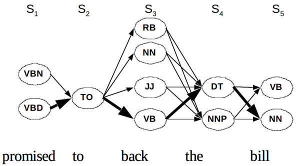
- HMM Tagger
  - As usual, we can use maximum likelihood estimation from relative frequencies to estimate these probabilities.

$$P(t_i|t_{i-2}t_{i-1}) = \frac{c(t_{i-2}t_{i-1}t_i)}{c(t_{i-2}t_{i-1})}$$

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

Stochastic Tagging

- Viterbi



Transformation-based Tagging

- Brill Tagger
  - Transformation-Based Tagging, sometimes called Brill tagging,
  - draws inspiration from both the rule-based and stochastic taggers.
  - Like the rule-based taggers, TBL is based on rules that specify what tags should be assigned to what words.
  - But like the stochastic taggers, TBL is a machine learning technique, in which rules are automatically induced from the data.
  - Like some but not all of the HMM taggers, TBL is a supervised learning technique; it assumes a pre-tagged training corpus.

Transformation-based Tagging

- Brill Tagger
  - Input:
    - tagged corpus
    - dictionary (with most frequent tags)
  - Basic Idea:
    - Set the most probable tag for each word as a start value
    - Change tags according to rules of type "if word-1 is a determiner and word is a verb then change the tag to noun" in a specific order
  - Training is done on tagged corpus:
    - Write a set of rule templates
    - Among the set of rules, find one with highest score
    - Continue from 2 until lowest score threshold is passed
    - Keep the ordered set of rules
  - Rules make errors that are corrected by later rules

Transformation-based Tagging

- Brill Tagger
  - Before the rules apply, the tagger labels every word with its most-likely tag.
  - We get these most-likely tags from a tagged corpus.
  - For example, in the Brown corpus, race is most likely to be a noun:
  - $P(NN \mid race) = .98$
  - $P(VB \mid race) = .02$
  - This means that the two examples of race that we saw above will both be coded as NN.

Transformation-based Tagging

- Brill Tagger
  - In the first case, this is a mistake, as NN is the incorrect tag
  - is/VBZ expected/VBN to/TO race/NN tomorrow/NN
  - In the second case this race is correctly tagged as an NN:
  - the/DT race/NN for/IN outer/JJ space/NN
  - After selecting the most-likely tag, Brill's tagger applies its transformation rules.
  - As it happens, Brill's tagger learned a rule that applies exactly to this mistagging of race:
  - Change NN to VB when the previous tag is TO
  - This rule would change race/NN to race/VB in exactly the following situation, since it is preceded by to/TO:
  - expected/VBN to/TO race/NN $--$ > expected/VBN to/TO race/VB

Transformation-based Tagging

- Brill Tagger
  - Brill's TBL algorithm has three major stages.
  - It first labels every word with its most-likely tag.
  - It then examines every possible transformation, and selects the one that results in the most improved tagging.
  - Finally, it then re-tags the data according to this rule.
  - These three stages are repeated until some stopping criterion is reached, such as insufficient improvement over the previous pass.

Transformation-based Tagging

- Brill Tagger
  - Note that stage two requires that TBL knows the correct tag of each word; i.e., TBL is a supervised learning algorithm.
  - The output of the TBL process is an ordered list of transformations

Transformation-based Tagging

- TBL-Problems
  - In principle the set of possible transformations is infinite
  - since we could imagine transformations such as "transform NN to VB if the previous word was 'IBM"'
  - But TBL needs to consider every possible transformation, in order to pick the best one on each pass through the algorithm.
  - Thus the algorithm needs a way to limit the set of transformations.
  - This is done by designing a small set of templates, abstracted transformations.
  - Rules are learned in ordered sequence
  - Rules are compact and can be inspected by humans

Stochastic Tagging

- Viterbi

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:'.

Transformation-based Tagging

- TBL-Problems
  - First 100 rules achieve 96.8% accuracy
  - First 200 rules achieve 97.0% accuracy
  - Execution Speed: TBL tagger is slower than HMM approach
  - Learning Speed: Brill's implementation over a day (600k tokens)
  - BUT ...
    1. Learns small number of simple, non-stochastic rules
    2. Can be made to work faster with FST
    3. Best performing algorithm on unknown words

Transformation-based Tagging

- TBL-Problems
  - New words added to (newspaper) language 20+ per month
  - Plus many proper names . . .
  - Increases error rates by 1-2%
  - Method 1: assume they are nouns
  - Method 2: assume the unknown words have a probability distribution similar to words only occurring once in the training set.
  - Method 3: Use morphological information, e.g., words ending with -ed tend to be tagged VBN.

Transformation-based Tagging

- TBL- evaluation
  - The result is compared with a manually coded “Gold Standard”
  - Typically accuracy reaches 96-97%
  - This may be compared with result for a baseline tagger
  - Important: 100% is impossible even for human annotators.

References

- Speech and Language Processing (3rd ed. draft) by D. Jurafsky & J. H. Martin (web.stanford.edu)