LING 520: Computational Analysis of English

Semester: FALL '16

Instructor: Sowmya Vajjala

Iowa State University, USA

29 September 2016

Class outline

- Assignment 2 Discussion
- POS Tagging - Background
- Rule based and Probabilistic POS tagging
- Assignment 3 Description
- Practice exercises

Assignment 2 Discussion

Volunteers needed.

POS tagging Background-1

- Task: tag every word in a sentence with its part of speech.
- Used as a pre-processing task for a number of other NLP tasks. Also useful for doing corpus linguistic analysis
- Problem 1: disambiguating which tag to use for a word in given context
- Problem 2: how to adapt the tagger to different types and genres of data (news text, research articles speech, tweets etc.)

POS tagging Background-2

- Broadly 8 parts of speech described in Grammar classes - Noun, verb, adjective, adverb, pronoun, preposition, conjunction, interjection.
- In NLP, they are slightly more fine-grained.
- POS taggers for a language are developed based on the tagsets that are standardized for that language.
- Tagsets are standardized by projects involving groups of linguists and computer scientists, usually in the early days of NLP research for a given language.
- While all tagsets have some common features, there are also some language specific tags.

Some POS Tagsets: English

PTB tagset

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 ”
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	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*			

Other English Tagsets

- Though PTB is the most commonly used tagset for English, there are some others too.
- e.g., NUPOS tagset: http://morphadorner.northwestern.edu/documentation/nupos/
- CLAWS7 tagset: http://ucrel.lancs.ac.uk/claws/

Some POS Tagsets: German

2.3 Tag-Tabelle

POS =	Beschreibung	Beispiele
ADJA	attributives Adjektiv	*[das] große [Haus]*
ADJD	adverbiales oder prädikatives Adjektiv	*[er fährt] schnell* *[er ist] schnell*
ADV	Adverb	*schon, bald, doch*
APPR	Präposition; Zirkumposition links	*in [der Stadt], ohne [mich]*
APPRART	Präposition mit Artikel	*im [Haus], zur [Sache]*
APPO	Postposition	*[ihm] zufolge, [der Sache] wegen*
APZR	Zirkumposition rechts	*[von jetzt] an*
ART	bestimmter oder unbestimmter Artikel	*der, die, das, ein, eine*
CARD	Kardinalzahl	*zwei [Männer], [im Jahre] 1994*
FM	Fremdsprachliches Material	*[Er hat das mit "] A big fish [" übersetzt]*
ITJ	Interjektion	*mhm, ach, tja*
KOUI	unterordnende Konjunktion mit "zu" und Infinitiv	*um [zu leben], anstatt [zu fragen]*
KOUS	unterordnende Konjunktion mit Satz	*weil, daß, damit, wenn, ob*
KON	nebenordnende Konjunktion	*und, oder, aber*
KOKOM	Vergleichspartikel, ohne Satz	*als, wie*
NN	Appellativa	*Tisch, Herr, [das] Reisen*
NE	Eigennamen	*Hans, Hamburg, HSV*
PDS	substituierendes Demonstrativ–pronomen	*dieser, jener*
PDAT	attribuierendes Demonstrativ–pronomen	*jener [Mensch]*
PIS	substituierendes Indefinit–pronomen	*keiner, viele, man, niemand*
PIAT	attribuierendes Indefinit–pronomen ohne Determiner	*kein [Mensch], irgendein [Glas]*
PIDAT	attribuierendes Indefinit–pronomen mit Determiner	*[ein] wenig [Wasser], [die] beiden [Brüder]*
PPER	irreflexives Personalpronomen	*ich, er, ihm, mich, dir*
PPOSS	substituierendes Possessiv–pronomen	*meins, deiner*
PPOSAT	attribuierendes Possessivpronomen	*mein [Buch], deine [Mutter]*
PRELS	substituierendes Relativpronomen	*[der Hund,] der*

POS =	Beschreibung	Beispiele
PRELAT	attribuierendes Relativpronomen Relativpronomen	*[der Mann ,] dessen [Hund]*
PRF	reflexives Personalpronomen	*sich, einander, dich, mir*
PWS	substituierendes Interrogativpronomen	*wer, was*
PWAT	attribuierendes Interrogativpronomen	*welche [Farbe], wessen [Hut]*
PWAV	adverbiales Interrogativ– oder Relativpronomen	*warum, wo, wann, worüber, wobei*
PAV	Pronominaladverb	*dafür, dabei, deswegen, trotzdem*
PTKZU	"zu" vor Infinitiv	*zu [gehen]*
PTKNEG	Negationspartikel	*nicht*
PTKVZ	abgetrennter Verbzusatz	*[er kommt] an, [er fährt] rad*
PTKANT	Antwortpartikel	*ja, nein, danke, bitte*
PTKA	Partikel bei Adjektiv oder Adverb	*am [schönsten], zu [schnell]*
TRUNC	Kompositions–Erstglied	*An– [und Abreise]*
VVFIN	finites Verb, voll	*[du] gehst, [wir] kommen [an]*
VVIMP	Imperativ, voll	*komm [!]*
VVINF	Infinitiv, voll	*gehen, ankommen*
VVIZU	Infinitiv mit "zu", voll	*anzukommen, loszulassen*
VVPP	Partizip Perfekt, voll	*gegangen, angekommen*
VAFIN	finites Verb, aux	*[du] bist, [wir] werden*
VAIMP	Imperativ, aux	*sei [ruhig !]*
VAINF	Infinitiv, aux	*werden, sein*
VAPP	Partizip Perfekt, aux	*gewesen*
VMFIN	finites Verb, modal	*dürfen*
VMINF	Infinitiv, modal	*wollen*
VMPP	Partizip Perfekt, modal	*[er hat] gekonnt*
XY	Nichtwort, Sonderzeichen enthaltend	*D2XW3*
$,	Komma	,
$.	Satzbeendende Interpunktion	. ? ! ; :
$(	sonstige Satzzeichen; satzintern	– [ ] ( )

Some POS Tagsets: Hindi

13.1. POS Tag Set for Indian Languages (Nov 2006, IIIT Hyderabad)

Sl No.	Category	Tag name	Example
1.1	Noun	NN	
1.2	NLoc	NST	
2.	Proper Noun	NNP	
3.1	Pronoun	PRP	
3.2	Demonstrative	DEM	
4	Verb-finite	VM	
5	Verb Aux	VAUX	
6	Adjective	JJ	
7	Adverb	RB	*Only manner adverb
8	Post position	PSP	
9	Particles	RP	bhI, to, hI, jI, hA.N, na,
10	Conjuncts	CC	bole (Bangla)
11	Question Words	WQ	
12.1	Quantifiers	QF	bahut, tho.DA, kam (Hindi)
12.2	Cardinal	QC	
12.3	Ordinal	QO	
12.4	Classifier	CL	
13	Intensifier	INTF	
14	Interjection	INJ	
15	Negation	NEG	
16	Quotative	UT	ani (Telugu), endru (Tamil), bole/mAne (Bangla), mhaNaje (Marathi), mAne (Hindi)
17	Sym	SYM	
18	Compounds	*C	
19	Reduplicative	RDP	
20	Echo	ECH	
21	Unknown	UNK	

It was decided that for foreign/unknown words that the POS tagger may give a tag "UNK"

Tagged Reference Corpus

- When they create tagsets, researchers also create a reference corpus of sentences which are manually annotated with these tags.
- Annotation projects are usually long, time and money intensive, and lots of linguists work together.
- Lots of guidelines, and manuals are prepared to make annotations unambiguous, and with good agreement between humans.
- These tagged corpora are then used for developing automatic taggers.
- So, important thing to keep in mind: there is no guarantee that the tagger will do well on unseen, out of domain data.

POS Tagging: How?

All approaches to tagging fall into one of the two categories:

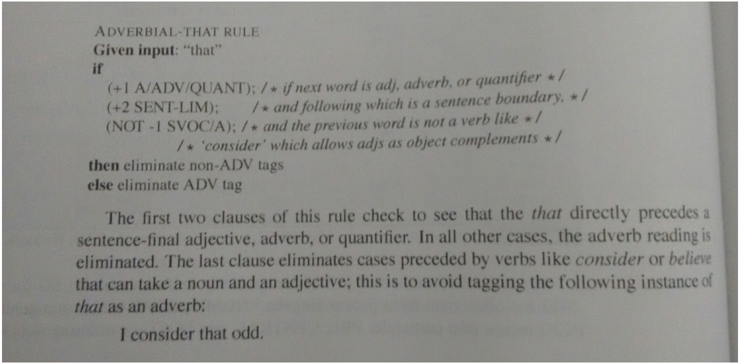
1. Rule based tagging
2. Probabilistic tagging

There is something called "Transformation based Learning" which has features of both these methods.

Rule based tagging

This is usually a two stage process:

1. Stage 1: Assign all possible POS tags for a given word based on language dictionary.
2. Stage 2: Write linguistic disambiguation rules to choose one POS tag for that word in that context.



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

The first two clauses of this rule check to see that the *that* directly precedes a sentence-final adjective, adverb, or quantifier. In all other cases, the adverb reading is eliminated. The last clause eliminates cases preceded by verbs like *consider* or *believe* that can take a noun and an adjective; this is to avoid tagging the following instance of *that* as an adverb:

I consider that odd.

Probabilistic Tagging - Background

- POS tagging can be viewed in two ways:
  1. Text classification: Given some word, predict its most likely tag - classifying some text (word) into some pre-defined category (tag)
  2. As sequence classification/labeling: given a sequence of words, predict the most likely sequence of tags.
- POS tagging is generally treated and modeled as sequence classification.
- Aim in modeling POS tagging as sequence labeling: given a sequence of n words, out of all possible sequences of n tags, choose the one that is most likely for that word sequence

Sequence Labeling - Mathematical Notation

- If w_1^n is our sequence of n words,
- ... and t_1^n is the collection of all sequences of ntags
- We should pick a tag sequence, so that $\mathrm{P}(\mathrm{t}_1^n|\mathrm{w}_1^n)$ is the highest.
- $\hat{t}_1^n = \mathrm{argmax}_{t_1^n}\ \mathrm{P}(\mathrm{t}_1^n|\mathrm{w}_1^n)$
- If we use this notation: $\mathrm{argmax}_x\ \mathrm{f}(\mathrm{x})$, it means "value of x such that f(x) is maximized".
- So how do we find out $\mathrm{P}(\mathrm{t}_1^n|\mathrm{w}_1^n)$ first, to get its argmax value?

When in trouble, ask Mr Bayes

- $\mathrm{P}(\mathrm{t}_1^n|\mathrm{w}_1^n) = \mathrm{P}(\mathrm{w}_1^n|\mathrm{t}_1^n)*\mathrm{P}(\mathrm{t}_1^n)/\mathrm{P}(\mathrm{w}_1^n)$
- So, $\hat{t}_1^n = \mathrm{argmax}_{t_1^n}(\mathrm{P}(\mathrm{w}_1^n|\mathrm{t}_1^n)*\mathrm{P}(\mathrm{t}_1^n)/\mathrm{P}(\mathrm{w}_1^n))$
- which is $\hat{t}_1^n = \mathrm{argmax}_{t_1^n}(\mathrm{P}(\mathrm{w}_1^n|\mathrm{t}_1^n)*\mathrm{P}(\mathrm{t}_1^n))$ - how? why is the denominator dropped?

When in trouble, ask Mr Bayes

- $P(t_1^n|w_1^n) = P(w_1^n|t_1^n)*P(t_1^n)/P(w_1^n)$
- So, $\hat{t}_1^n = \text{argmax}_{t_1^n}(P(w_1^n|t_1^n)*P(t_1^n)/P(w_1^n))$
- which is $\hat{t}_1^n = \text{argmax}_{t_1^n}(P(w_1^n|t_1^n)*P(t_1^n))$ - how? why is the denominator dropped?
- Answer: We are doing argmax over tag sequence. That does not affect the word sequence probability. So, that will remain same in all comparisons for argmax.
- Here, $P(w_1^n|t_1^n)$ is the likelihood of the word string given a tag string and $P(t_1^n)$ is the prior probability of a tag sequence.

Assumptions we make in calculations

- This equation $\hat{t}_1^n = \text{argmax}_{t_1^n}(\mathrm{P}(\mathrm{w}_1^n|\mathrm{t}_1^n)*\mathrm{P}(\mathrm{t}_1^n))$ suffers from the same problems as language models. So we make some assumptions to simplify our calculations again.
- First assumption: probability of a word depends only on its POS tag, not on previous words or tags.
  $\Rightarrow \mathrm{P}(\mathrm{w}_1^n|\mathrm{t}_1^n) \approx \pi_{i=1}^n\ \mathrm{P}(\mathrm{w}_i|\mathrm{t}_i)$
- Second assumption: probability of a tag appearing is only dependent on previous tag instead of entire history (for Bigram taggers!)
  $\Rightarrow \mathrm{P}(\mathrm{t}_1^n) \approx \pi_{i=1}^n\ \mathrm{P}(\mathrm{t}_i|\mathrm{t}_{i-1})$
- Considering these assumptions, $\hat{t}_1^n$ becomes:
  $\text{argmax}_{t_1^n}\pi_{i=1}^n(\mathrm{P}(\mathrm{w}_i|\mathrm{t}_i)*\mathrm{P}(\mathrm{t}_i|\mathrm{t}_{i-1}))$

What does all this mean in normal language?

- This is the equation we have:
  $\hat{t}_1^n = \text{argmax}_{t_1^n} \pi_{i=1}^n (\text{P}(\text{w}_i|\text{t}_i) * \text{P}(\text{t}_i|\text{t}_{i-1}))$
- $(\text{P}(\text{w}_i|\text{t}_i) = \text{Count}(\text{t}_i, w_i)/\text{Count}(\text{t}_i)$
  $\Rightarrow \text{P}(\text{good}|\text{ADJ}) = \text{C}(\text{ADJ,good})/\text{C}(\text{ADJ})$
- $\text{P}(\text{t}_i|\text{t}_{i-1}) = \text{Count}(\text{t}_{i-1}, t_i)/\text{Count}(\text{t}_{i-1})$ (Same as in language models)
  $\Rightarrow \text{P}(\text{NN}|\text{DT}) = \text{C}(\text{DT,NN})/\text{C}(\text{DT})$
- This is HMM tagging. The "Hidden" here refers to the tag sequence, because that is something that we cannot observe in the input which is only a sequence of words.
- If we use HMM for speech recognition, "observed" state is speech sequence, hidden state is word sequence.

Note: There is a lot of stuff about HMM I am not discussing here. Interested people - read Chapter 5 (and perhaps 6 too) in J&M. It may need advance mathematical background.

Assignment 3 Description

- clarification regarding Q1: I am not expecting you to implement a HMM tagger (you can, if you want to). I am fine with seeing an implementation that is based on trigrams of the kind P(Tag2|(Tag1Word2))

Practice Exercise -1

Find one tagging error in each of the following sentences tagged with the PTB tagset. (Q 5.1 in J&M)

1. I/PRP need/VBP a/DT flight/NN from/IN Atlanta/NN
2. Does/VBZ this/DT flight/NN serve/VB dinner/NNS
3. I/PRP have/VB a/DT friend/NN living/VBG in/IN Denver/NNP
4. Can/VBP you/PRP list/VB the/DT nonstop/JJ afternoon/NN flights/NNS

note: PTB tags: https://goo.gl/JVKbYM

Practice Exercise -2

Chapter 5 of NLTK book is about POS tagging and how to use NLTK for that. Go to RegEx tagger description there, and try to develop a small regex tagger on your own, which identifies adjectives and verbs. Test it with some sentences and note your observations. Work in groups of 3. On tuesday, we will start with some presentations of this.

Practice Exercise - 3

Modify Katrin Erk's code sample to incorporate Laplace smoothing and return Log probabilities instead of normal probability.
Note: This is going to be useful in doing Question 1 of Assignment 3. So, try to do this.

Next Class

- Conclusion of POS tagging discussion: Transformation Based Learning, Evaluation of Taggers
- ToDo: listen to Lectures 7.2-7.5 in Radev's coursera course.
- Read and try to work through examples in Chapter 5 of NLTK (This will help you solve one question in A3)