Spoken Human Robot Interaction

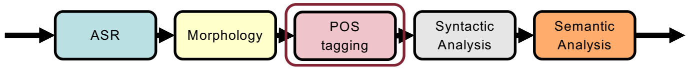
POS Tagging

SAPIENZA
UNIVERSITÀ DI ROMA

Andrea Vanzo, **Daniele Nardi**
lastname@dis.uniroma1.it
Department of Computer, Control, and Management Engineering
"Sapienza" University of Rome, Italy

Artificial Intelligence
AY 2018/19

Our Command Interpretation pipeline



- POS (Part-Of-Speech) useful because of the large amount of information they give about a word (and its neighbors)
  – Knowing whether a word is a **noun** or a **verb** tells a lot about likely neighboring words (e.g., nouns are usually preceded by determiners and adjectives)
- Largely support the Syntactic Analysis
  – Gives information about the syntactic structure around the word (e.g., nouns are generally part of noun phrases)

Outline

- POS tagging
  – What is a POS
  – What is the POS tagging task
  – Approaches
    - Rule-based
    - Statistical approaches
- Evaluation Metrics
  – Gold standard
  – Most Frequent Class Baseline
  – Accuracy

What is a POS tag?

A POS tag is the **morpho-syntactic category** a word belongs to (Dionysius Thrax of Alexandria 100 B.C.)

He defined 8 POS that have lasted until modern linguistics

Main categories:
- Open: noun, verb, adjective, adverb, …
- Closed: determiner, pronoun, …

Main Open POS tag classes

- **Nouns**
  - proper,
  - Common (count) *dog, book, …*
  - Common (mass) *snow, salt, …*
- **Verbs** (Inflections)
  - 3rd pers (*s*), progressive (*ing*), past part. (*ed*)
- **Adjectives** (not in all languages e.g. Korean)
- **Adverbs** (modifiers, but not only of verbs!!)

  "Actually I ran *home extremely quickly yesterday"*
  - Locative, Degree, Manner, Temporal, …

Main Close POS tag classes

- **Prepositions**: spatial, temporal *on, after, with*
- **Particles**: non-compositional *run out, go on,*
- **Determiners**: articles, …, *the, a, an, this,*
- **Conjunctions**: coordinating, subordinating

  and, but *that, after*
- **Pronouns**: personal, possessive, wh-

  I, me, *my, their* *what,whom*
- **Auxiliaries** *be, have, can, may*
- **Interjections** *oh, not, please, there*

Treebanks (Penn 45 classes)

Tag	Description	Example	Tag	Description	Example	Tag	Description	Example
CC	coordinating conjunction	*and, but, or*	PDT	predeterminer	*all, both*	VBP	verb non-3sg present	*eat*
CD	cardinal number	*one, two*	POS	possessive ending	*'s*	VBZ	verb 3sg pres	*eats*
DT	determiner	*a, the*	PRP	personal pronoun	*I, you, he*	WDT	wh-determ.	*which, that*
EX	existential 'there'	*there*	PRP$	possess. pronoun	*your, one's*	WP	wh-pronoun	*what, who*
FW	foreign word	*mea culpa*	RB	adverb	*quickly*	WP$	wh-possess.	*whose*
IN	preposition/ subordin-conj	*of, in, by*	RBR	comparative adverb	*faster*	WRB	wh-adverb	*how, where*
JJ	adjective	*yellow*	RBS	superlatv. adverb	*fastest*	$	dollar sign	*$*
JJR	comparative adj	*bigger*	RP	particle	*up, off*	#	pound sign	*#*
JJS	superlative adj	*wildest*	SYM	symbol	*+,%, &*	"	left quote	' or "
LS	list item marker	*1, 2, One*	TO	"to"	*to*	"	right quote	' or "
MD	modal	*can, should*	UH	interjection	*ah, oops*	(	left paren	[, (, {, <
NN	sing or mass noun	*llama*	VB	verb base form	*eat*	)	right paren	],), }, >
NNS	noun, plural	*llamas*	VBD	verb past tense	*ate*	,	comma	,
NNP	proper noun, sing.	*IBM*	VBG	verb gerund	*eating*	.	sent-end punc	. ! ?
NNPS	proper noun, plu.	*Carolinas*	VBN	verb past part.	*eaten*	:	sent-mid punc	: ; ... – -

What is POS tagging?

- Task of assigning a POS tag to each word of the sentence (or paragraph, document, corpus)
- Input:
  - Tagset
  - Model
  - Input sentence
- Output
  - Labeled sentence (a single POS tag for each token)

go	to	the	kitchen
↓	↓	↓	↓
V	P	DET	N

POS tagging and morphology

- Morphology: given a word, find all the possible morphological interpretations
  - Might be ambiguous
  - talks → talk+s → talk V 3PS **OR** talk N PL
- POS tagging: given a word, find only one morphological interpretation
  - Might be seen as morphological analysis + disambiguation
  - Uses contextual information (preceding words)

POS tagging: ambiguities

- Without contextual information the task becomes extremely challenging
  - *Content*: noun or adjective? Depends on the pronunciation or context
    - *Còntent* is the noun: “check the content of the e-mail”
    - *Contènt* is the adjective: “I’m content”
- English language is rather ambiguous (WSJ)
  - Unambiguous (1 tag): 44,423 (86%)
  - Ambiguous (2+ tag): 7,025 (14%)
  - But 45-55% of words are ambiguous

POS tagging: approaches

- Rule-based: tagging is performed according to hand-crafted rules
  - Naïve tagger
- Stochastic: tagging is performed according to statistical evidences
  - Hidden Markov Models (HMMs)
  - Maximum Entropy Markov Models (MEMMs)
  - Recurrent Neural Networks

POS tagging: Naïve tagger

1. Assign all the possible POS tag to each word of the sentence, driven by a dictionary
2. Apply hand-crafted disambiguation rules

RULES

1: if (VBN|VBD) AND “<begin>,PRP,VBN”
REMOVE VBN
2: if (VB|NN) AND “DT,VB”
REMOVE VB

POS DICTIONARY

back	VB, JJ, RB, NN
bill	NN, VB
promised	VBN , VBD
she	PRP
the	DT
to	TO

			NN		
			RB		
	VBD		JJ		~~VB~~
PRP	~~VBN~~	TO	VB	DT	NN
she	promised	to	back	the	bill

POS tagging: HMM

Probably the most used approach

Goal:
given a sequence of n words $W = [w_1, \dots, w_n]$, find the most likely sequence of POS tags $T^* = [t_1, \dots, t_n]$

$$T^* = \underset{T}{\operatorname{argmax}}\, P(T|W)$$

- States are mapped into POS
- Output symbols are words

after applying Bayes' rule

$$T^* = \underset{T}{\operatorname{argmax}} \frac{P(W|T)P(T)}{P(W)} = \underset{T}{\operatorname{argmax}}\, P(W|T)P(T)$$

POS tagging: HMM

- Two further assumptions
  - The probability of a word depends only on its own tag

$$P(W|T) = \prod_{i=1}^{n} P(w_i|t_i)$$

  - The probability of a tag depends only on the previous tag (*bi-gram* assumption)

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

POS tagging: HMM

- The final equation becomes

$$T^* = \underset{T}{\operatorname{argmax}}\, P(W|T)P(T) = \prod_{i=1}^{n} P(w_i|t_i)P(t_i|t_{i-1})$$

where:

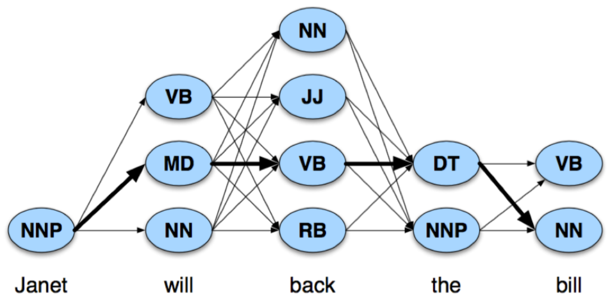
- $P(w_i|t_i)$ is the **emission probability**
  - Probability of emitting the word w_i *given the POS tag* t_i
- $P(t_i|t_{i-1})$ is the **transition probability**
  - Probability of transitioning from POS t_{i-1} to POS t_i

POS tagging: HMM – estimating probabilities

- Probabilities estimated by simply counting occurrences in a large corpus
- Two probability matrices are acquired
  - Emission probability matrix
  - Transition probability matrix

POS tagging: HMM – tagging

- The good old Viterbi algorithm



POS tagging: MEMM

Discriminative sequence model: Sequential adaptation of the multinomial logistic regression classifier

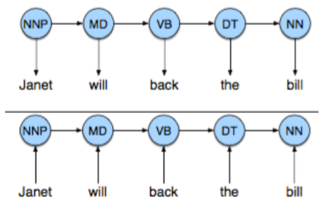
Goal: given a sequence of n words $W = [w_1, \dots, w_n]$, find the most likely sequence of POS tag $T^* = [t_1, \dots, t_n]$

$$T^* = \underset{T}{\operatorname{argmax}} P(T|W)$$

The posterior is computed directly, as a discriminative process across tags, training a a logistic regression classifier

$$T^* = \underset{T}{\operatorname{argmax}} P(T|W) = \underset{T}{\operatorname{argmax}} \prod_i P(t_i|w_i, t_{i-1})$$

POS tagging: MEMM vs HMM



- While HMM computes the likelihood of the observation (word) given the hidden state (POS), MEMM computes the posterior of each state (POS), given the observation (word) and the previous state (POS)

POS tagging: MEMM - features

- $P(t_i|w_i, t_{i-1})$ computed training a logistic regression classifier
- However, w_i and t_{i-1} are not enough to properly characterize the current “example”
- A better feature vector is required and discriminative classifiers allow to incorporate a vast plethora of features
- Through feature engineering we can represent better the example
  - E.g. targeted word, neighboring words, previous tags, capitalization, …

POS tagging: MEMM – training and tagging

- Training: just like any other discriminative classifier
  - Labeled training set → feature extraction → parameter estimation
- Tagging: two possibilities
  - Greedy algorithm: each decision is atomic and does not take into account the “future” choices
    - Makes hard decision on each word before moving on to the next word
  - Viterbi algorithm: chooses the best path that is optimal for the whole sequence
    - The best path is known at the end of the tagging process

Evaluating POS tagging

- Many of the ambiguous tokens are easy to disambiguate
  - Different tags associated with a word are not equally likely
  - E.g. *a* might be either a determiner or a letter, but determiner is more likely
- Two baseline for evaluating POS tagging
  - Gold standard: human annotated datasets
  - *Most Frequent Class Baseline*: assigning each token to the class it occurred in most often in the training set

Evaluating POS tagging: Accuracy

- A standard way to measure the performance of part of-speech taggers is **Accuracy**:
  - percentage of tags correctly labeled
- Procedure:
  - Evaluate the POS tags using the POS tagger
  - Compare the results against either the gold standard, or the Most Frequent Class Baseline
- Gold standard accuracy is supposed to be 100%
- Most Frequent Class Baseline achieves 92.34% over the WSJ dataset
- State-of-the-art taggers achieve around 97%
  - Problem is considered solved

References

- Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. https://web.stanford.edu/~jurafsky/slp3/

Chapter 8 (excluding 8.6)