Hasso Plattner Institut

IT Systems Engineering | Universität Potsdam

Natural Language Processing

Part of Speech Tagging and Named Entity Recognition

Potsdam, 3 May 2012

Saeedeh Momtazi
Information Systems Group

based on the slides of the course book

Outline

Hasso Plattner Institut

1. Part of Speech Tagging
2. Named Entity Recognition
3. Sequential Modeling

Outline

1. Part of Speech Tagging
2. Named Entity Recognition
3. Sequential Modeling

Parts Of Speech (POS)

- 8 Parts of speech are traditionally used to summarize the linguistic knowledge
  - Noun, Verb, Preposition, Adverb, Article, Interjection, Pronoun, Conjunction

- The modified list is currently used
  - Noun, Verb, Auxiliary, Preposition, Adjective, Adverb, Number, Determiner, Interjection, Pronoun, Conjunction, Particle

- Known as:
  - Parts of speech
  - Lexical categories
  - Word classes
  - Morphological classes
  - Lexical tags

POS Examples

Noun	book/books, sugar, Germany, Sony
Verb	eat, wrote
Auxiliary	can, should, have
Adjective	new, newer, newest
Adverb	well, urgently
Numbers	872, two, first
Determiner	the, some
Conjunction	and, or
Pronoun	he, my
Preposition	to, in
Particle	off, up
Interjection	Ow, Eh

Open vs. Closed Classes

- Closed (limited number of words, do not grow usually)
  - Determiners: the, some, a, an, ...
  - Pronouns: she, he, I, ...
  - Prepositions: to, in, on, under, over, by, ...
  - Auxiliaries: can, should, have, had, ...
  - Conjunctions: and, or
  - Particles: off, up
  - Interjections: Ow, Eh

- Open (unlimited number of words)
  - Nouns
  - Verbs
  - Adjectives
  - Adverbs

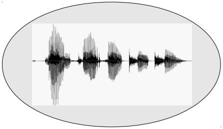
Applications

- Speech Synthesis
- Parsing
- Machine Translation
- Information Extraction

Applications

- Speech Synthesis

How to pronounce *"lead"* ?



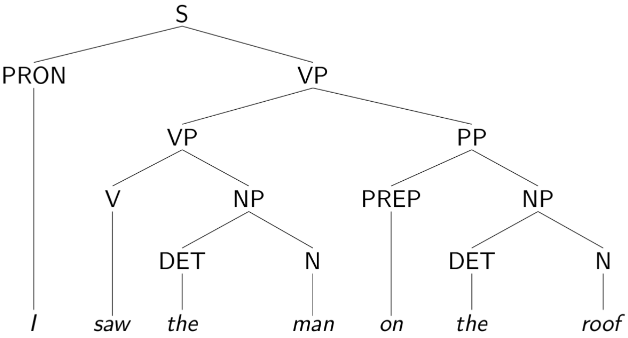
Applications

- Machine Translation

"I like ..."	$\Rightarrow$	*"Ich mag ..."*
		"Ich wie ..."

Applications

- Parsing



POS Tagset

- There are so many parts of speech tagsets we can draw
- Choosing a standard tagset is essential
- Tag types
  - Coarse-grained
    - noun
    - verb
    - adjective
    - ...
  - Fine-grained
    - noun-proper-singular, noun-proper-plural, noun-common-mass, ..
    - verb-past, verb-present-3rd, verb-base, ...
    - adjective-simple, adjective-comparative, ...
    - ...

Penn TreeBank

A large annotated corpus of English
tagset: 45 tags

Penn TreeBank Tagset

HPI Hasso Plattner Institut

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*			

POS Tagging

Hasso Plattner Institut

- Definition
  - The process of assigning a part of speech to each word in a text

- Challenge
  - Words often have more than one POS

On my back$_{[NN]}$

The back$_{[JJ]}$ *door*

Win the voters back$_{[RB]}$

Promised to back$_{[VB]}$ *the bill*

Distribution of Ambiguities

		45-tag Treebank Brown	
Unambiguous (1 tag)		**38,857**	
Ambiguous (2–7 tags)		**8844**	
Details:	2 tags	6,731	
	3 tags	1621	
	4 tags	357	
	5 tags	90	
	6 tags	32	
	7 tags	6	(*well*, *set*, *round*, *open*, *fit*, *down*)
	8 tags	4	(*'s*, *half*, *back*, *a*)
	9 tags	3	(*that*, *more*, *in*)

POS Tagging

Plays well with others

Plays	NNS/VBZ
well	UH/JJ/NN/RB
with	IN
others	NNS

Plays$_{[VBZ]}$ *well*$_{[RB]}$ *with*$_{[IN]}$ *others*$_{[NNS]}$

Performance

- Baseline model
  - Tagging unambiguous words with the correct label
  - Tagging ambiguous words with their most frequent label
  - Tagging unknown words as a noun

Already performs around 90%

Outline

1. Part of Speech Tagging

2. Named Entity Recognition

3. Sequential Modeling

Motivation

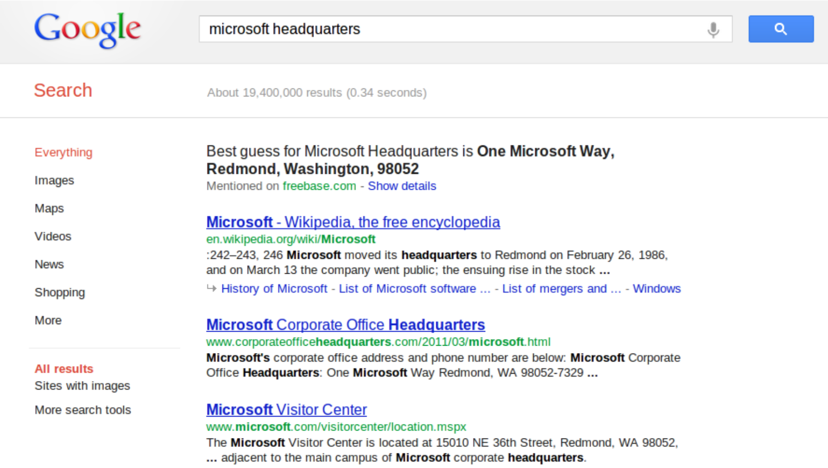
- Factual information and knowledge are normally expressed by named entities
  - Who, Whom, Where, When, ...
- Question answering systems are looking for named entities to answer users' questions
- Named entity recognition is the core of the information extraction systems

Applications

- Finding the important information of an event from an invitation
  - Date, Time, Location, Host, Contact person
- Finding the main information of a company from its reports
  - Founder, Board members, Headquarters, Profits
- Finding medical information from medical literature
  - Drugs, Genes, Interaction products
- Finding the target of sentiments
  - Products, Celebrities

Applications

Hasso Plattner Institut



Named Entity Recognition (NER)

- Finding named entities in a text
- Classifying them to the corresponding classes

"Steven Paul Jobs, co-founder of Apple Inc, was born in California."

" $\underbrace{\text{Steven Paul Jobs}}$, co-founder of $\underbrace{\text{Apple Inc}}$, was born in $\underbrace{\text{California}}$."

" $\underbrace{\text{Steven Paul Jobs}}_{PER}$, co-founder of $\underbrace{\text{Apple Inc}}_{ORG}$, was born in $\underbrace{\text{California}}_{LOC}$."

Named Entity Classes

- Person
  - Person names
- Organization
  - Companies, Government, Organizations, Committees, ..
- Location
  - Cities, Countries, Rivers, ..
- Date and time expression
- Measure
  - Percent, Money, Weight, ...
- Religious
- Book title
- Movie title
- Drug name

NER Task

- Assigning a label to each token of the text

IO

Steven	PER
Paul	PER
Jobs	PER
,	O
co-founder	O
of	O
Apple	ORG
Inc	ORG
,	O
was	O
born	O
in	O
California	LOC
.	O

IOB

Steven	B-PER
Paul	I-PER
Jobs	I-PER
,	O
co-founder	O
of	O
Apple	B-ORG
Inc	I-ORG
,	O
was	O
born	O
in	O
California	B-LOC
.	O

NER Ambiguity

- IO vs. IOB Encoding

John	PER
Shows	O
Mary	PER
Hermann	PER
Hesse	PER
’s	O
book	O
.	O

John	B-PER
Shows	O
Mary	B-PER
Hermann	B-PER
Hesse	I-PER
’s	O
book	O
.	O

NER Ambiguity

- Ambiguity between named entities and common words
  - May

- Ambiguity between named entity types
  - Washington (Location or Person)

Outline

1. Part of Speech Tagging
2. Named Entity Recognition
3. Sequential Modeling

Task

- Similar to a normal classification task
  - Feature Selection
  - Algorithm

POS Tagging

- Features

Word	the: the $\rightarrow$ DT
Prefixes	unbelievable: un- $\rightarrow$ JJ
Suffixes	slowly: -ly $\rightarrow$ RB
Lowercased word	Importantly: importantly $\rightarrow$ RB
Capitalization	Stefan: [CAP] $\rightarrow$ NNP
Word shapes	35-year: d-x $\rightarrow$ JJ

- Model
  - Maximum Entropy
    $P(t|w)$

Data	Performance
Overall	93.7
Unknown	82.6

NER

- Features

Word	Germany: Germany
POS tag	Washington: NNP
Capitalization	Stefan: [CAP]
Punctuation	St.: [PUNC]
Lowercased word	Book: book
Suffixes	Spanish: -ish
Word shapes	1920-2008: dddd-dddd

- List lookup
  - Extensive list of names are available via various resources
  - Gazetteer: a large list of place names

POS Tagging

- More Features?

They[PRP] *left*[VBD] *as*[IN] *soon*[RB] *as*[IN] *he*[PRP] *arrivied*[VBD]

- Better Algorithm
  - Using Sequence Modeling

Sequence Modeling

- Many of the NLP techniques should deal with data represented as sequence of items
  - Characters, Words, Phrases, Lines, ...

警察枪杀了那个逃

B I BI B B B B I

$\textit{I}_{[PRP]}$ $\textit{saw}_{[VBP]}$ $\textit{the}_{[DT]}$ $\textit{man}_{[NN]}$ $\textit{on}_{[IN]}$ $\textit{the}_{[DT]}$ $\textit{roof}_{[NN]}$.

Steven	Paul	Jobs,	co-founder	of	Apple	Inc,	was	born	in	California.
PER	PER	PER	O	O	ORG	ORG	O	O	O	LOC

Sequence Modeling

- Making a decision based on the
  - Current Observation
    - Word (W_0)
    - Prefix
    - Suffix
    - Lowercased word
    - Capitalization
    - Word shape
  - Surrounding observations
    - W_{+1}
    - W_{-1}
  - Previous decisions
    - T_{-1}
    - T_{-2}

Maximum Entropy Markov Model (MEMM)
Conditional Markov Model (CMM)

Context Words

- NER
  - Sherwood Forest
  - Portobello Street
  - Mr Smith
  - Apple Inc
  - John earns 3000 €
  - John joined IBM

Learning Model

Training

$$\begin{array}{l} T_1 \rightarrow C_1 \\ T_2 \rightarrow C_2 \\ \ldots \\ T_n \rightarrow C_n \end{array} \longrightarrow \begin{array}{c} F_1 \\ F_2 \\ \ldots \\ F_n \end{array} \longrightarrow \boxed{\text{Model(F,C)}}$$

Testing

$$T_{n+1} \rightarrow ? \longrightarrow F_{n+1} \longrightarrow C_{n+1}$$

Sequence Modeling

- Greedy inference
  - Starting from the beginning of the sequence
  - Assigning a label to each item using the classifier in that position
  - Using previous decisions as well as the observed data

- Beam inference
  - Keeping the top *k* labels in each position
  - Extending each sequence in each local way
  - Finding the best *k* labels for the next position

Hidden Markov Model (HMM)

- Finding the best sequence of tags ($t_1 \dots t_n$) that corresponds to the sequence of observations ($w_1 \dots w_n$)

- Probabilistic View
  - Considering all possible sequences of tags
  - Choosing the tag sequence from this universe of sequences, which is most probable given the observation sequence

$$\hat{t}_1^n = \text{argmax}_{t_1^n} P(t_1^n | w_1^n)$$

Using Bayes Rule

$$\hat{t}_1^n = \text{argmax}_{t_1^n} P(t_1^n | w_1^n)$$

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

$$P(t_1^n | w_1^n) = \frac{P(w_1^n | t_1^n) \cdot P(t_1^n)}{P(w_1^n)}$$

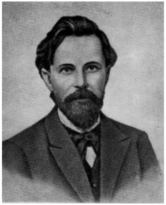
$$\hat{t}_1^n = \text{argmax}_{t_1^n} P(w_1^n | t_1^n) \cdot P(t_1^n)$$

Using Markov Assumption

$$\hat{t}_1^n = \text{argmax}_{t_1^n} P(w_1^n|t_1^n) \cdot P(t_1^n)$$

$$P(w_1^n|t_1^n) \approx \prod_{i=1}^{n} P(w_i|t_i)$$

$$P(t_1^n) \approx \prod_{i=1}^{n} P(t_i|t_{i-1})$$



$$\hat{t}_1^n = \text{argmax}_{t_1^n} \prod_{i=1}^{n} P(w_i|t_i) \cdot P(t_i|t_{i-1})$$

Two Probabilities

- The tag transition probabilities: $P(t_i|t_{i-1})$
  - Finding the likelihood of a tag to proceed by another tag
  - Similar to the normal bigram model

$$P(t_i|t_{i-1}) = \frac{C(t_{i-1}, t_i)}{C(t_{i-1})}$$

Two Probabilities

- The word likelihood probabilities: $P(w_i|t_i)$
  - Finding the likelihood of a word to appear given a tag

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

Two Probabilities

$I_{[PRP]}$ $saw_{[VBP]}$ $the_{[DT]}$ $man_{[NN?]}$ $on_{[]}$ $the_{[]}$ $roof_{[]}$.

$$P([NN]|[DT]) = \frac{C([DT],[NN])}{C([DT])}$$

$$P(man|[NN]) = \frac{C([NN],man)}{C([NN])}$$

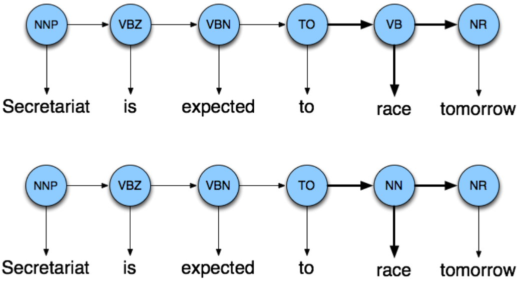
Ambiguity

Secretariat$_{[NNP]}$ *is*$_{[VBZ]}$ *expected*$_{[VBN]}$ *to*$_{[TO]}$ ***race***$_{[VB]}$ *tomorrow*$_{[NR]}$.

People$_{[NNS]}$ *inquire*$_{[VB]}$ *the*$_{[DT]}$ *reason*$_{[NN]}$ *for*$_{[IN]}$ *the*$_{[DT]}$ ***race***$_{[NN]}$.

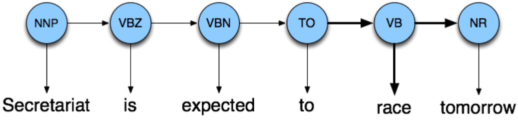
Ambiguity

Secretariat$_{[NNP]}$ *is*$_{[VBZ]}$ *expected*$_{[VBN]}$ *to*$_{[TO]}$ ***race***$_{[VB]}$ *tomorrow*$_{[NR]}$.



Ambiguity

Secretariat$_{[NNP]}$ *is*$_{[VBZ]}$ *expected*$_{[VBN]}$ *to*$_{[TO]}$ ***race***$_{[VB]}$ *tomorrow*$_{[NR]}$.

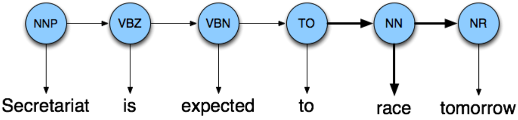


$$P(VB|TO) = 0.83$$
$$P(race|VB) = 0.00012$$
$$P(NR|VB) = 0.0027$$

$$P(VB|TO)P(NR|VB)P(race|VB) = 0.00000027$$

Ambiguity

Secretariat$_{[NNP]}$ *is*$_{[VBZ]}$ *expected*$_{[VBN]}$ *to*$_{[TO]}$ ***race***$_{[VB]}$ *tomorrow*$_{[NR]}$.



$$P(NN|TO) = 0.00047$$

$$P(race|NN) = 0.00057$$

$$P(NR|NN) = 0.0012$$

$$P(NN|TO)P(NR|NN)P(race|NN) = 0.00000000032$$

Hidden Markov Model (HMM)

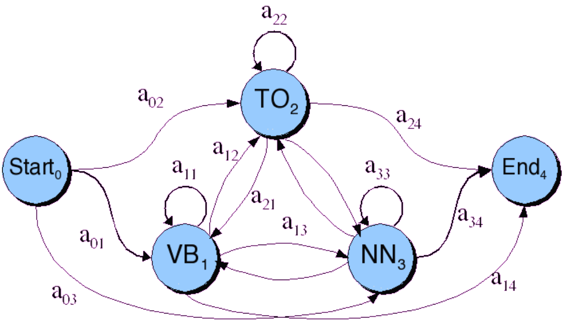
- A weighted finite-state automaton adds probabilities to the arcs
  - The probabilities leaving any arc must sum to one

- An HMM is an extension of a Markov chain in which the input symbols are not the same as the states

- We do not know which state we are in
  - The output symbols are words
  - The hidden states are POS tags

Hidden Markov Model (HMM)

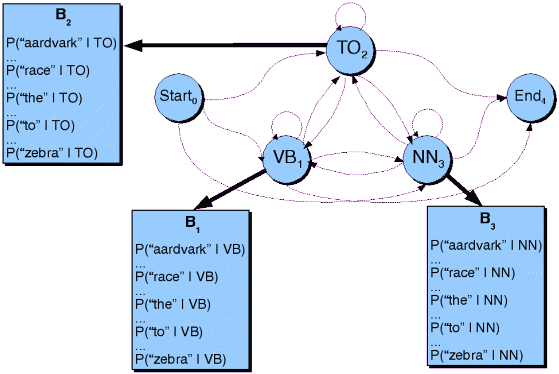
- Transition probabilities



Hidden Markov Model (HMM)

Hasso Plattner Institut

- Word likelihood probabilities



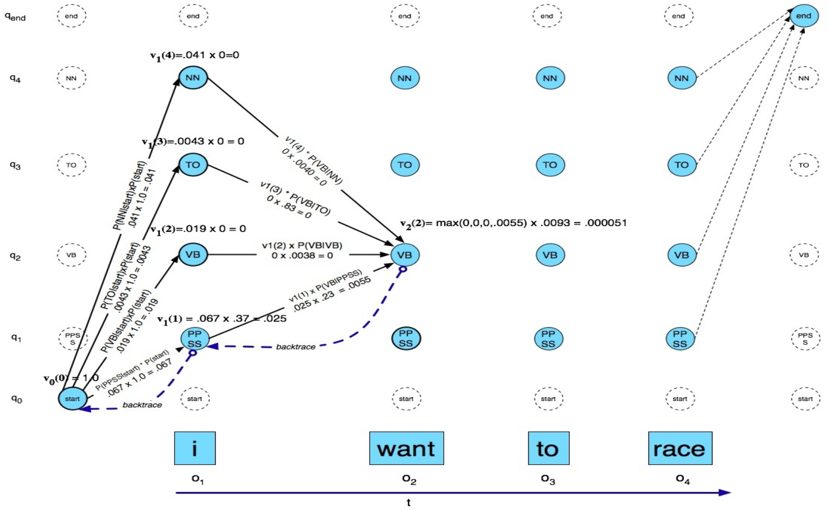
The Viterbi Algorithm

- Creating an array
  - Columns corresponding to inputs
  - Rows corresponding to possible states

- Sweeping through the array in one pass filling the columns left to right using the transition probabilities and observation probabilities

- Storing the max probability path to each cell (not all paths) using dynamic programming

The Viterbi Algorithm



Further Reading

- Speech and Language Processing
  - Chapter 5: POS Tagging
  - Chapter 6: MaxEnt & HMM
  - Chapter 22.1: NER