Constraint-based Part-of-Speech Tagging

Neng-Fa Zhou

CUNY Brooklyn College & Graduate Center

Part-of-Speech Tagging

- Identify the lexical type, called part-of-speech, of each word in a sentence.

  John (NN) saw (VB) the (DT) saw (NN) and (CC)
  decided (VB) to (TO) take (VB) it (PRP) to (IN) the (DT) table (NN).

- POS-tagging approaches
  - rule-based, probabilistic models, NN models.
- The SOTA POS taggers achieve over 97% word accuracy, but only 60% sentence accuracy.

  Naomi Osaka places (NN) top priority on consistency.

  What you listen to sounds (NN) amazing. **NLTK/spaCy/Standford/SyntaxNet**

Constraint-based POS Tagging

- Treats POS tagging as a CSP
  - Treats each word as a variable
    - The domain is a set of all possible POS tags determined by a lexicon
      - D_{can} = {NN, MD, VB}
  - Context constraints encode linguistic knowledge
    - e.g., *a noun phrase cannot begin with a base-form verb*
  - Utilizes statistical models to order domain values

Constraint-based POS Tagging

- Advantages
  - Works well with a backtracking syntax parser
    - Allows multiple assignments
    - Chooses the next viable assignment upon backtracking
  - Harnesses knowledge and statistics
- Disadvantages
  - Requires a lexicon
    - Fortunately, one can be built from online resources
  - Only nouns can be open, and all others are closed

Lexicon

verbs

see	see
sees	see
saw	see
seeing	see
seen	see
.	.
.	.
.	.

nouns

seed	seed
seeds	seed
can	can
cans	can
flies	fly
.	.
.	.
.	.

adjectives

good	good
better	good
best	good
sunny	sunny
sunnier	sunny
.	.
.	.
.	.

modal verbs

adverbs

prepositions

pronouns

determiners

conjunctions

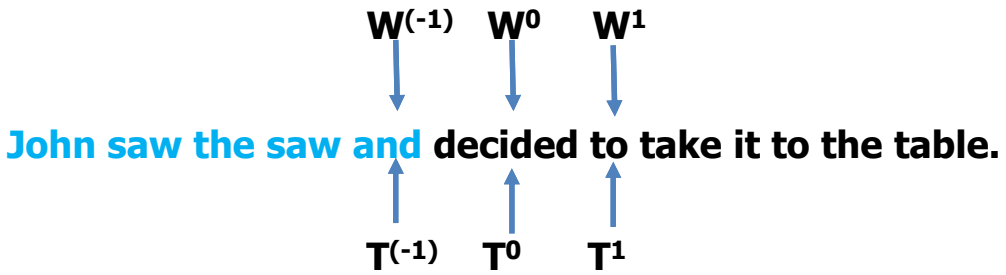
POS Tags (An Abstraction of Penn Treebank POS Tags)

CC	Conjunction (e.g., *if*, *and*, *or*)
DT	Determiner (e.g., *a*, *the*, *this*)
IN	Preposition (e.g., *for*, *on*)
JJ	Adjective
MD	Modal (e.g., *can*, *must*)
NJ	Noun or adjective
NN	Noun
PR	Pronoun (e.g., *we*, *its*)
PS	Possessive (e.g., *'s* in *John's book*)
RB	Adverb
SYM	Symbol
THERE	The word *there*
THAT	The word *that*
TO	The word *to*
VB	Verb

Predicates Used in Context Constraints

- `article(W)` is true if W is one of the articles: *a*, *an*, and *the*.
- `base(W)` is true if W is a base-form noun, verb, or adjective. For example, `base(can)`, `base(good)` and `base(see)` are true, but `base(sounds)` and `base(reduced)` are false.
- `be(W)` is true if W is a be-verb, meaning that W is one of the following: *am*, *are*, *be*, *been*, *being*, *is*, *was*, *were*.
- `bp(W)` is true if W is a base-form word or a plural noun: `bp(W)` $\leftrightarrow$ `base(W)` $\vee$ `plural(W)`.
- `phrasal_verb_of(W)` is true if W and the word *of* constitute a phrasal verb. For example, *conceive of* and *think of* are phrasal verbs.
- `plural(W)` is true if W is a plural form of a noun for which `base(W)` is false. For example, `plural(cans)` and `plural(leaves)` are true, but `plural(sheep)` is false because *sheep*'s plural form is identical to its base form.
- `ppn(W)` is true if W is a possessive pronoun. For example, `ppn(its)` and `ppn(her)` are true.
- `complementizer([`W_1`,`W_2`, ..., `W_n`])` is true if the word sequence W_1,W_2, ..., W_n forms a complementizer [4]. For example, `complementizer([as,long,as])` and `complementizer([assuming, that])` are true.

Context Constraints



Context Constraints on Nouns

CN-1: $T^{-1} = \texttt{JJ} \wedge \texttt{bp}(W^0) \rightarrow T^0 \neq \texttt{VB}$ Ex: *industrial conglomerate*

CN-2: $T^{-1} = \texttt{PS} \wedge \texttt{bp}(W^0) \rightarrow T^0 \neq \texttt{VB}$ Ex: *company's jump*

CN-3: $\texttt{ppn}(W^{-1}) \wedge \texttt{bp}(W^0) \rightarrow T^0 \neq \texttt{VB}$ Ex: *her company*

CN-4: $\texttt{article}(W^{-1}) \wedge \texttt{bp}(W^0) \rightarrow T^0 \neq \texttt{VB}$ Ex: *a company*

CN-5: $\texttt{complementizer}([W^{-k}, \ldots, W^{-1}]) \wedge \texttt{bp}(W^0) \rightarrow T^0 \neq \texttt{VB}$
Ex: *as long as company*

CN-6: $T^{-1} = \texttt{IN} \wedge \texttt{bp}(W^0) \rightarrow T^0 \neq \texttt{VB}$ Ex: *on leave*

CN-7: $\texttt{be}(W^{-1}) \wedge \texttt{plural}(W^0) \rightarrow T^0 = \texttt{NN}$ Ex: *are concerns*

CN-8: $\texttt{bp}(W^0) \wedge W^1 = \texttt{of} \wedge \neg\, \texttt{phrasal\_verb\_of}(W^0) \rightarrow T^0 = \texttt{NN}$
Ex: *yields of*

Context Constraints on Nouns

CV-1: $T^{-2} = \texttt{NN} \wedge T^{-1} = \texttt{MD} \rightarrow T^{0} \neq \texttt{NN}$

Ex: *stocks will jump*

CV-2: $\texttt{pre-infinitive}(W^{-2}) \wedge W^{-1} = \texttt{to} \rightarrow T^{0} \neq \texttt{NN}$

Ex: *he has to leave*
he want to leave
he attempt to leave

Ordering Domain Values

- Unigram model

$$P(T^0|W^0),$$

D_{can} = [MD, VB, NN]

- Trigram model

$$P(T^0|T^{-2}, T^{-1})$$

Experimental Results

Table 1: CPOST's performance on identifying BP verbs

Datasets	CPOST0		CPOST	
	Recall	Precision	Recall	Precision
TreeBank	89.5%	90.5%	92.2%	95.3%
CONLL2000	89.5%	86.7%	90.0%	92.6%

Conclusion

- Constraint-based POS tagging
  - Allows multiple assignments
  - Chooses the next viable assignment upon backtracking
  - Harnesses knowledge and statistics
- Related work
  - Rule-based POS tagging
  - Constraint grammar
- Future work
  - A comprehensive set of constraints
  - NN for ordering domain values
  - How parsing can improve POS tagging