POS Tagging & Sequence Labeling Tasks

CMSC 470
Marine Carpuat
This is a simple sentence.

But it is an instructive one.
Parts of Speech

• “Equivalence class” of linguistic entities
  • “Categories” or “types” of words that occur in similar morphological and syntactic contexts

• Study dates back to the ancient Greeks
  • Dionysius Thrax of Alexandria (c. 100 BC)
  • 8 parts of speech: noun, verb, pronoun, preposition, adverb, conjunction, participle, article
  • Remarkably enduring list!
How can we define POS?

• By meaning?
  • Verbs are actions
  • Adjectives are properties
  • Nouns are things

• By the syntactic environment
  • What occurs nearby?
  • What does it act as?

• By what morphological processes affect it
  • What affixes does it take?

• Typically combination of syntactic+morphology
Parts of Speech

• Open class
  • Impossible to completely enumerate
  • New words continuously being invented, borrowed, etc.

• Closed class
  • Closed, fixed membership
  • Reasonably easy to enumerate
  • Generally, short function words that “structure” sentences
Open Class POS

• Four major open classes in English
  • Nouns
  • Verbs
  • Adjectives
  • Adverbs

• All languages have nouns and verbs... but may not have the other two
Nouns

• Open class
  • New inventions all the time: muggle, webinar, ...

• Semantics:
  • Generally, words for people, places, things
  • But not always (bandwidth, energy, ...)

• Syntactic environment:
  • Occurring with determiners
  • Pluralizable, possessivizable

• Other characteristics:
  • Mass vs. count nouns
Verbs

• Open class
  • New inventions all the time: google, tweet, ...

• Semantics
  • Generally, denote actions, processes, etc.

• Syntactic environment
  • E.g., Intransitive, transitive

• Other characteristics
  • Main vs. auxiliary verbs
  • Gerunds (verbs behaving like nouns)
  • Participles (verbs behaving like adjectives)
Adjectives and Adverbs

• Adjectives
  • Generally modify nouns, e.g., tall building

• Adverbs
  • A semantic and formal hodge-podge...
  • Sometimes modify verbs, e.g., sang beautifully
  • Sometimes modify adjectives, e.g., extremely cold
Closed Class POS

• Prepositions
  • In English, occurring before noun phrases
  • Specifying some type of relation (spatial, temporal, ...)
  • Examples: on the shelf, before noon

• Particles
  • Resembles a preposition, but used with a verb (“phrasal verbs”)
  • Examples: find out, turn over, go on
Particle vs. Prepositions

He came *by* the office in a hurry  
He came *by* his fortune honestly  
(both *by* are particles)

We ran *up* the phone bill  
We ran *up* the small hill  
(both *up* are prepositions)

He lived *down* the block  
He never lived *down* the nicknames  
(both *down* are prepositions)
More Closed Class POS

• Determiners
  • Establish reference for a noun
  • Examples: *a, an, the* (articles), *that, this, many, such, ...*

• Pronouns
  • Refer to person or entities: *he, she, it*
  • Possessive pronouns: *his, her, its*
  • Wh-pronouns: *what, who*
Closed Class POS: Conjunctions

- Coordinating conjunctions
  - Join two elements of “equal status”
  - Examples: cats *and* dogs, salad *or* soup

- Subordinating conjunctions
  - Join two elements of “unequal status”
  - Examples: We’ll leave *after* you finish eating. *While* I was waiting in line, I saw my friend.
  - Complementizers are a special case: I think *that* you should finish your assignment
Beyond English...

Chinese
No verb/adjective distinction!

漂亮: beautiful/to be beautiful

Riau Indonesian/Malay
No Articles
No Tense Marking
3rd person pronouns neutral to both gender and number
No features distinguishing verbs from nouns

Ayam (chicken) Makan (eat)
- The chicken is eating
- The chicken ate
- The chicken will eat
- The chicken is being eaten
- Where the chicken is eating
- How the chicken is eating
- Somebody is eating the chicken
- The chicken that is eating
POS TAGGING
POS Tagging: What’s the task?

• Process of assigning part-of-speech tags to words

• But what tags are we going to assign?
  • Coarse grained: noun, verb, adjective, adverb, ...
  • Fine grained: {proper, common} noun
  • Even finer-grained: {proper, common} noun ± animate

• Important issues to remember
  • Choice of tags encodes certain distinctions/non-distinctions
  • Tagsets will differ across languages!

• For English, Penn Treebank is the most common tagset
Penn Treebank Tagset: 45 Tags

<table>
<thead>
<tr>
<th>Tag</th>
<th>Description</th>
<th>Example</th>
<th>Tag</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>coordin. conjunction</td>
<td><em>and, but, or</em></td>
<td>SYM</td>
<td>symbol</td>
<td>+, %, &amp;</td>
</tr>
<tr>
<td>CD</td>
<td>cardinal number</td>
<td><em>one, two, three</em></td>
<td>TO</td>
<td>“to”</td>
<td>to</td>
</tr>
<tr>
<td>DT</td>
<td>determiner</td>
<td><em>a, the</em></td>
<td>UH</td>
<td>interjection</td>
<td><em>ah, oops</em></td>
</tr>
<tr>
<td>EX</td>
<td>existential ‘there’</td>
<td><em>there</em></td>
<td>VB</td>
<td>verb, base form</td>
<td><em>eat</em></td>
</tr>
<tr>
<td>FW</td>
<td>foreign word</td>
<td><em>mea culpa</em></td>
<td>VBD</td>
<td>verb, past tense</td>
<td><em>ate</em></td>
</tr>
<tr>
<td>IN</td>
<td>preposition/sub-conj</td>
<td><em>of, in, by</em></td>
<td>VBG</td>
<td>verb, gerund</td>
<td><em>eating</em></td>
</tr>
<tr>
<td>JJ</td>
<td>adjective</td>
<td><em>yellow</em></td>
<td>VBN</td>
<td>verb, past participle</td>
<td><em>eaten</em></td>
</tr>
<tr>
<td>JJR</td>
<td>adj., comparative</td>
<td><em>bigger</em></td>
<td>VBP</td>
<td>verb, non-3sg pres</td>
<td><em>eat</em></td>
</tr>
<tr>
<td>JJS</td>
<td>adj., superlative</td>
<td><em>wildest</em></td>
<td>VBZ</td>
<td>verb, 3sg pres</td>
<td><em>eats</em></td>
</tr>
<tr>
<td>LS</td>
<td>list item marker</td>
<td><em>1, 2, One</em></td>
<td>WDT</td>
<td>wh-determiner</td>
<td><em>which, that</em></td>
</tr>
<tr>
<td>MD</td>
<td>modal</td>
<td><em>can, should</em></td>
<td>WP</td>
<td>wh-pronoun</td>
<td><em>what, who</em></td>
</tr>
<tr>
<td>NN</td>
<td>noun, sing. or mass</td>
<td><em>llama</em></td>
<td>WP$</td>
<td>possessive wh-</td>
<td><em>whose</em></td>
</tr>
<tr>
<td>NNS</td>
<td>noun, plural</td>
<td><em>llamas</em></td>
<td>WRB</td>
<td>wh-adverb</td>
<td><em>how, where</em></td>
</tr>
<tr>
<td>NNP</td>
<td>proper noun, singular</td>
<td><em>IBM</em></td>
<td>$</td>
<td>dollar sign</td>
<td>$</td>
</tr>
<tr>
<td>NNPS</td>
<td>proper noun, plural</td>
<td><em>Carolinas</em></td>
<td>#</td>
<td>pound sign</td>
<td>#</td>
</tr>
<tr>
<td>PDT</td>
<td>preadeterminer</td>
<td><em>all, both</em></td>
<td>“</td>
<td>left quote</td>
<td>‘ or “</td>
</tr>
<tr>
<td>POS</td>
<td>possessive ending</td>
<td><em>‘s</em></td>
<td>”</td>
<td>right quote</td>
<td>“ or ”</td>
</tr>
<tr>
<td>PRP</td>
<td>personal pronoun</td>
<td><em>I, you, he</em></td>
<td>(</td>
<td>left parenthesis</td>
<td>[, (, {, &lt;</td>
</tr>
<tr>
<td>PRPS</td>
<td>possessive pronoun</td>
<td><em>your, one’s</em></td>
<td>)</td>
<td>right parenthesis</td>
<td>)., }, &gt;</td>
</tr>
<tr>
<td>RB</td>
<td>adverb</td>
<td><em>quickly, never</em></td>
<td>,</td>
<td>comma</td>
<td></td>
</tr>
<tr>
<td>RBR</td>
<td>adverb, comparative</td>
<td><em>faster</em></td>
<td>.</td>
<td>sentence-final punctuation</td>
<td>! ?</td>
</tr>
<tr>
<td>RBS</td>
<td>adverb, superlative</td>
<td><em>fastest</em></td>
<td>:</td>
<td>mid-sentence punctuation</td>
<td>; ; ... -</td>
</tr>
<tr>
<td>RP</td>
<td>particle</td>
<td><em>up, off</em></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Penn Treebank Tagset: Choices

• Example:
  • The/DT grand/JJ jury/NN commmented/VBD on/IN a/DT number/NN of/IN other/JJ topics/NNS ./.

• Distinctions and non-distinctions
  • Prepositions and subordinating conjunctions are tagged “IN” (“Although/IN I/PRP.”)
  • Except the preposition/complementizer “to” is tagged “TO”
Why do POS tagging?

• One of the most basic NLP tasks
  • Nicely illustrates principles of data-driven NLP

• Useful for higher-level analysis
  • Needed for syntactic analysis
  • Needed for semantic analysis
Try your hand at tagging...

• The back door
• On my back
• Win the voters back
• Promised to back the bill
Try your hand at tagging...

• I hope that she wins
• That day was nice
• You can go that far
Why is POS tagging hard?

• Ambiguity!

• Ambiguity in English
  • 11.5% of word types ambiguous in Brown corpus
  • 40% of word tokens ambiguous in Brown corpus
  • Annotator disagreement in Penn Treebank: 3.5%
POS tagging: how to do it?

• Given Penn Treebank, how would you build a system that can POS tag new text?

• Baseline: pick most frequent tag for each word type
  • 90% accuracy if train+test sets are drawn from Penn Treebank

• How can we do better?
We can view POS tagging as classification and use the perceptron again!

\[ \hat{y} = \arg\max_{\hat{y} \in \mathcal{Y}(x)} w \cdot \phi(x, \hat{y}) \]

Algorithm 40 \textit{StructuredPerceptronTrain}(D, MaxIter)

1: \( w \leftarrow 0 \) \hspace{1cm} // initialize weights
2: \textbf{for} iter = 1 \ldots MaxIter \textbf{do}
3: \hspace{1cm} \textbf{for} all \((x, y) \in D\) \textbf{do}
4: \hspace{2cm} \hat{y} \leftarrow \arg\max_{\hat{y} \in \mathcal{Y}(x)} w \cdot \phi(x, \hat{y}) \hspace{1cm} // compute prediction
5: \hspace{2cm} \textbf{if} \ \hat{y} \neq y \textbf{ then}
6: \hspace{3cm} w \leftarrow w + \phi(x, y) - \phi(x, \hat{y}) \hspace{1cm} // update weights
7: \hspace{2cm} \textbf{end if}
8: \hspace{1cm} \textbf{end for}
9: \textbf{end for}
10: \textbf{return} w \hspace{1cm} // return learned weights
Sequence labeling problem

• Input:
  • sequence of tokens $x = [x_1 \ldots x_L]$
  • Variable length $L$

• Output (aka label):
  • sequence of tags $y = [y_1 \ldots y_L]$
  • # tags = $K$
  • Size of output space?

Structured Perceptron

• Perceptron algorithm can be used for sequence labeling

• But there are challenges
  • How to compute argmax efficiently?
  • What are appropriate features?

• Approach: leverage structure of output space
Feature functions for sequence labeling

\[ x = \text{“monsters eat tasty bunnies”} \]
\[ y = \text{noun verb adj noun} \]

- Example features?
  - Number of times “monsters” is tagged as noun
  - Number of times noun is followed by verb
  - Number of times tasty as tagged as verb
  - Number of times two verbs are adjacent
  - ...
Feature functions for sequence labeling

- Standard features of POS tagging
  - **Unary features**: capture relationship between input $x$ and a **single label** in the output sequence $y$
    - e.g., “# times word $w$ has been labeled with tag $l$ for all words $w$ and all tags $l$”
  - **Markov features**: capture relationship between **adjacent labels** in the output sequence $y$
    - e.g., “# times tag $l$ is adjacent to tag $l'$ in output for all tags $l$ and $l'$”

- Given these feature types, the size of the feature vector is constant with respect to input length

$x = "\text{monsters eat tasty bunnies}"$

$y = \text{noun verb adj noun}$

Example from CIML chapter 17
We can view POS tagging as classification and use the perceptron again!

\[
\hat{y} = \arg\max_{\hat{y} \in \mathcal{Y}(x)} w \cdot \phi(x, \hat{y})
\]

---

**Algorithm 40** \texttt{StructuredPerceptronTrain}(D, MaxIter)

1. \( w \leftarrow 0 \)  // initialize weights
2. \textbf{for} iter = 1 \ldots MaxIter \textbf{do}
3. \hspace{1em} \textbf{for all} \ (x, y) \in D \textbf{do}
4. \hspace{2em} \hat{y} \leftarrow \arg\max_{\hat{y} \in \mathcal{Y}(x)} w \cdot \phi(x, \hat{y}) \hspace{1em} // compute prediction
5. \hspace{2em} \textbf{if} \ \hat{y} \neq y \textbf{ then}
6. \hspace{3em} \hspace{1em} w \leftarrow w + \phi(x, y) - \phi(x, \hat{y}) \hspace{1em} // update weights
7. \hspace{2em} \textbf{end if}
8. \hspace{1em} \textbf{end for}
9. \textbf{end for}
10. \textbf{return} \ w \hspace{1em} // return learned weights

Algorithm from CIML chapter 17
Solving the argmax problem for sequences

- Trellis sequence labeling
  - Any path represents a labeling of input sentence
  - Gold standard path in red

- Each edge receives a weight such that adding weights along the path corresponds to score for input/output configuration

- Any max-weight path algorithm can find the argmax
  - e.g. Viterbi algorithm $O(LK^2)$
Solving the argmax problem for sequences with dynamic programming

\[ x = "monsters eat tasty bunnies" \]
\[ y = \text{noun verb adj noun} \]

- Efficient algorithms possible if the feature function decomposes over the input
- This holds for unary and markov features used for POS tagging
POS tagging

• An example of sequence labeling tasks
• Requires a predefined set of POS tags
  • Penn Treebank commonly used for English
  • Encodes some distinctions and not others
• Given annotated examples, we can address sequence labeling with multiclass perceptron
  • but computing the argmax naively is expensive
  • constraints on the feature definition make efficient algorithms possible