Tagging (POS, NER)

CS 485, Fall 2023
Applications of Natural Language Processing
https://people.cs.umass.edu/~brenocon/cs485_f23/

Brendan O'Connor
College of Information and Computer Sciences
University of Massachusetts Amherst
• HW2 - how's it going? Phase 1 tomorrow, Phase 2 next week (Fri 10/20)!

• Project proposals: due **Wed 10/25**
  https://people.cs.umass.edu/~brenocon/cs485_f23/project.html

• after that, HW3, syntax

• after that, Midterm: either 11/7 or 11/9. Will know soon. Practice questions will be available.
Topics overview
Part of speech tags

• Syntax = how words compose to form larger meaning-bearing units

• POS = syntactic categories for words
  • You could substitute words within a class and have a syntactically valid sentence.
  • Give information how words can combine.

• I saw the dog
• I saw the cat
• I saw the {table, sky, dream, school, anger, ...}

Schoolhouse Rock: Conjunction Junction
https://www.youtube.com/watch?v=ODGA7ssL-6g&index=1&list=PL6795522EAD6CE2F7
Demo

• https://corenlp.run/
Part of speech tagging

- I saw the fire today

- Fire!
### Open vs closed classes

<table>
<thead>
<tr>
<th>Open class (lexical) words</th>
<th>Closed class (functional)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nouns</strong></td>
<td><strong>Determiners</strong></td>
</tr>
<tr>
<td>Proper</td>
<td>the some</td>
</tr>
<tr>
<td>IBM</td>
<td></td>
</tr>
<tr>
<td>Italy</td>
<td></td>
</tr>
<tr>
<td>Common</td>
<td>and or</td>
</tr>
<tr>
<td>cat / cats</td>
<td></td>
</tr>
<tr>
<td>snow</td>
<td></td>
</tr>
<tr>
<td><strong>Verbs</strong></td>
<td><strong>Conjunctions</strong></td>
</tr>
<tr>
<td>Main</td>
<td>he its</td>
</tr>
<tr>
<td>see</td>
<td></td>
</tr>
<tr>
<td>registered</td>
<td></td>
</tr>
<tr>
<td><strong>Adjectives</strong></td>
<td><strong>Pronouns</strong></td>
</tr>
<tr>
<td>old</td>
<td>he</td>
</tr>
<tr>
<td>older</td>
<td>its</td>
</tr>
<tr>
<td>oldest</td>
<td></td>
</tr>
<tr>
<td><strong>Adverbs</strong></td>
<td><strong>Interjections</strong></td>
</tr>
<tr>
<td>slowly</td>
<td>Ow</td>
</tr>
<tr>
<td><strong>Numbers</strong></td>
<td><strong>Particles</strong></td>
</tr>
<tr>
<td>122,312</td>
<td>off</td>
</tr>
<tr>
<td>one</td>
<td></td>
</tr>
<tr>
<td><strong>Prepositions</strong></td>
<td><strong>Interjections</strong></td>
</tr>
<tr>
<td>to</td>
<td>Ow</td>
</tr>
<tr>
<td>with</td>
<td></td>
</tr>
</tbody>
</table>

---

*slide credit: Chris Manning*
Why do we want POS?

• Useful for many syntactic and other NLP tasks.
  • Phrase identification (“chunking”)
  • Named entity recognition (proper nouns are often names)
  • Syntactic/semantic dependency parsing
  • Sentiment
• Either as features or heuristic filtering
• Esp. useful when not much training data
• Limitations
  • Coarse approximation of grammatical features
  • Sometimes cases are hard and ambiguous
POS patterns: simple noun phrases
POS patterns: simple noun phrases

- Quick and dirty noun phrase identification (Justeson and Katz 1995, Handler et al. 2016)
- BaseNP = (Adj | Noun)* Noun
- PP = Prep Det* BaseNP
- NP = BaseNP PP*

Grammatical structure: Candidate strings are those multi-word noun phrases that are specified by the regular expression $((A \mid N)^+ \mid ((A \mid N)^*(NP)^?) (A \mid N)^*)N$.

<table>
<thead>
<tr>
<th>Tag Pattern</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>A N</td>
<td>linear function</td>
</tr>
<tr>
<td>N N</td>
<td>regression coefficients</td>
</tr>
<tr>
<td>A A N</td>
<td>Gaussian random variable</td>
</tr>
<tr>
<td>A N N</td>
<td>cumulative distribution function</td>
</tr>
<tr>
<td>N A N</td>
<td>mean squared error</td>
</tr>
<tr>
<td>N N N N</td>
<td>class probability function</td>
</tr>
<tr>
<td>N P N</td>
<td>degrees of freedom</td>
</tr>
</tbody>
</table>

Table 5.2 Part of speech tag patterns for collocation filtering. These patterns were used by Justeson and Katz to identify likely collocations among frequently occurring word sequences.
## Congressional bills

(Top terms, ranked by relative log-odds z-scores)

<table>
<thead>
<tr>
<th>Uni. Dem.</th>
<th>and, deleted, health, mental, domestic, inserting, grant, programs, prevention, violence, program, striking, education, forensic, standards, juvenile, grants, partner, science, research</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uni. Rep.</td>
<td>any, offense, property, imprisoned, whoever, person, more, alien, knowingly, officer, not, united, intent, commerce, communication, forfeiture, immigration, official, interstate, subchapter</td>
</tr>
<tr>
<td>NPs Dem.</td>
<td></td>
</tr>
<tr>
<td>NPs Rep.</td>
<td></td>
</tr>
</tbody>
</table>
POS patterns: sentiment


<table>
<thead>
<tr>
<th>Table 1. Patterns of tags for extracting two-word phrases from reviews.</th>
</tr>
</thead>
<tbody>
<tr>
<td>First Word</td>
</tr>
<tr>
<td>1. JJ</td>
</tr>
<tr>
<td>2. RB, RBR, or RBS</td>
</tr>
<tr>
<td>3. JJ</td>
</tr>
<tr>
<td>4. NN or NNS</td>
</tr>
<tr>
<td>5. RB, RBR, or VBS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. An example of the processing of a review that the author has classified as recommended.¹</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extracted Phrase</td>
</tr>
<tr>
<td>online experience</td>
</tr>
<tr>
<td>low fees</td>
</tr>
<tr>
<td>local branch</td>
</tr>
<tr>
<td>small part</td>
</tr>
<tr>
<td>online service</td>
</tr>
<tr>
<td>printable version</td>
</tr>
<tr>
<td>direct deposit</td>
</tr>
<tr>
<td>well other</td>
</tr>
<tr>
<td>inconveniently</td>
</tr>
<tr>
<td>located</td>
</tr>
<tr>
<td>other bank</td>
</tr>
<tr>
<td>true service</td>
</tr>
</tbody>
</table>

(plus co-occurrence information)
POS Taggers

- How do you predict POS tags?
- Off-the-shelf models widely available, at least for mainstream varieties of major world languages
  - e.g. Spacy, Stanza, CoreNLP, etc.
- Typically use logistic regression-like models
  - Each token instance is a classification problem
  - Labeled datasets: e.g. https://universaldependencies.org/
Useful features for a tagger

• Key sources of information:
  • 1. The word itself

• 2. Word-internal characters

• 3. Nearby words in a context window
  • Context window features are used for ALL tagging tasks!
  • Necessary to deal with *lexical ambiguity*
POS Tagging: lexical ambiguity

Can we just use a tag dictionary (one tag per word type)?

<table>
<thead>
<tr>
<th>Types:</th>
<th>WSJ</th>
<th>Brown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unambiguous (1 tag)</td>
<td>44,432 (86%)</td>
<td>45,799 (85%)</td>
</tr>
<tr>
<td>Ambiguous (2+ tags)</td>
<td>7,025 (14%)</td>
<td>8,050 (15%)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tokens:</th>
<th>WSJ</th>
<th>Brown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unambiguous (1 tag)</td>
<td>577,421 (45%)</td>
<td>384,349 (33%)</td>
</tr>
<tr>
<td>Ambiguous (2+ tags)</td>
<td>711,780 (55%)</td>
<td>786,646 (67%)</td>
</tr>
</tbody>
</table>

Most words types are unambiguous ...

But not so for tokens!

- Ambiguous wordtypes tend to be the common ones.
  - I know that he is honest = IN (relativizer)
  - Yes, that play was nice = DT (determiner)
  - You can’t go that far = RB (adverb)
POS Tagging: baseline

- Baseline: most frequent tag. 92.7% accuracy
  - Simple baselines are very important to run!

- Is this actually that high?
  - I get 0.918 accuracy for token tagging
  - ...but, 0.186 whole-sentence accuracy (!)
• Next: many other NLP tasks can be cast as tagging
  • Named entities
  • Word sense disambiguation
Named entity recognition

SOCCER - [PER BLINKER] BAN LIFTED .
[LOC LONDON] 1996-12-06 [MISC Dutch] forward
[PER Reggie Blinker] had his indefinite suspension
lifted by [ORG FIFA] on Friday and was set to make
his [ORG Sheffield Wednesday] comeback against
[ORG Liverpool] on Saturday . [PER Blinker] missed
his club’s last two games after [ORG FIFA] slapped a
worldwide ban on him for appearing to sign contracts for
both [ORG Wednesday] and [ORG Udinese] while he was
playing for [ORG Feyenoord].

Figure 1: Example illustrating challenges in NER.

- Goal: for a fixed entity type inventory (e.g. PERSON, LOCATION, ORGANIZATION), identify all spans from a document
- Name structure typically defined as flat (is this good?)

[Ratinov and Roth 2009]
BIO tagging

• Can we map identify phrases (spans) identification to token-level tagging?
**BIO tagging**

*Goal: represent two spans*

**NAME vs O**

doesn't work

Barack Obama Michelle Obama were ...

<table>
<thead>
<tr>
<th>N</th>
<th>N</th>
<th>N</th>
<th>N</th>
<th>N</th>
<th>O</th>
</tr>
</thead>
</table>

**BIO**

B-N I-N B-N I-N O

make cross-product of "B"egin and "I"nside against each class type:

O, B-PER, I-PER, B-LOC, I-LOC, ...

... then spans can easily be extracted from tagger output.
Features for tagging

• Word-based features
  • Word itself
  • Word shape ("Aa" "aa")
  • Contextual (word window) variants: versions of these at position t-1, t-2, t-3 … t+1, t+2, t+3 …

• External lexical knowledge
  • Gazetteer features: Does word/phrase occur in a list of known names?
  • Other hand-built lexicons

• Neural network embedding representations (later in course)
Intuition from Warren Weaver (1955):

“If one examines the words in a book, one at a time as through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of the words...

But if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say N words on either side, then if N is large enough one can unambiguously decide the meaning of the central word...

The practical question is : ``What minimum value of N will, at least in a tolerable fraction of cases, lead to the correct choice of meaning for the central word?"
Gazetteers example

1) **People**: people, births, deaths. Extracts 494,699 Wikipedia titles and 382,336 redirect links.

2) **Organizations**: cooperatives, federations, teams, clubs, departments, organizations, organisations, banks, legislatures, record labels, constructors, manufacturers, ministries, ministers, military units, military formations, universities, radio stations, newspapers, broadcasters, political parties, television networks, companies, businesses, agencies. Extracts 124,403 titles and 130,588 redirects.

3) **Locations**: airports, districts, regions, countries, areas, lakes, seas, oceans, towns, villages, parks, bays, bases, cities, landmarks, rivers, valleys, deserts, locations, places, neighborhoods. Extracts 211,872 titles and 194,049 redirects.

4) **Named Objects**: aircraft, spacecraft, tanks, rifles, weapons, ships, firearms, automobiles, computers, boats. Extracts 28,739 titles and 31,389 redirects.

5) **Art Work**: novels, books, paintings, operas, plays. Extracts 39,800 titles and 34037 redirects.

6) **Films**: films, telenovelas, shows, musicals. Extracts 50,454 titles and 49,252 redirects.

7) **Songs**: songs, singles, albums. Extracts 109,645 titles and 67,473 redirects.

8) **Events**: playoffs, championships, races, competitions, battles. Extracts 40,176 titles and 15,182 redirects.

[Ratinov and Roth 2009]
Word sense disambiguation

- Task: Choose a word’s sense in context
- Given KB and text:
  Want to tag spans in text with concept IDs
- Disambiguation problem
  - “I saw the bank” => bank#1 or bank#2?
  - “Michael Jordan was here” => ?

- Many terms for this: concept tagging, entity linking, “wikification”, WSD
Word sense disambiguation

- Supervised setting: need ground-truth concept IDs for words in text
- Main approach: use contextual information to disambiguate.
Intuition from Warren Weaver (1955):

“If one examines the words in a book, one at a time as through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of the words...

But if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say N words on either side, then if N is large enough one can unambiguously decide the meaning of the central word...

The practical question is: `What minimum value of N will, at least in a tolerable fraction of cases, lead to the correct choice of meaning for the central word?`
Two kinds of features in the vectors

- **Collocational** features and **bag-of-words** features
  - **Collocational**
    - Features about words at **specific** positions near target word
      - Often limited to just word identity and POS
  - **Bag-of-words**
    - Features about words that occur anywhere in the window (regardless of position)
      - Typically limited to frequency counts
Examples

• Example text (WSJ):
  An electric guitar and **bass** player stand off to one side not really part of the scene
• Assume a window of +/- 2 from the target
Examples

• Example text (WSJ)

  An electric guitar and bass player stand off to one side not really part of the scene,

• Assume a window of +/- 2 from the target
Collocational features

- Position-specific information about the words and collocations in window

\[ [w_{i-2}, \text{POS}_{i-2}, w_{i-1}, \text{POS}_{i-1}, w_{i+1}, \text{POS}_{i+1}, w_{i+2}, \text{POS}_{i+2}, w_{i-1}^i, w_{i+1}^i] \]

[guitar, NN, and, CC, player, NN, stand, VB, and guitar, player stand]

- word 1,2,3 grams in window of ±3 is common

[slide: SLP3]
Bag-of-words features

- “an unordered set of words” – position ignored
- Counts of words occur within the window.
- First choose a vocabulary
- Then count how often each of those terms occurs in a given window
  - sometimes just a binary “indicator” 1 or 0
Word sense disambiguation

• Supervised setting: need ground-truth concept IDs for words in text
• Contextual features
  • Word immediately to left ... to right ...
  • Word within 10 word window  (20 word window? entire document?)
• Features from matching a concept description, if your KB has one
  • Michael Jeffrey Jordan (born February 17, 1963), also known by his initials, MJ,[1] is an American former professional basketball player. He is also a businessman, and principal owner and chairman of the Charlotte Hornets. Jordan played 15 seasons in the National Basketball Association (NBA) for the Chicago Bulls and Washington Wizards.

• Overall (prior) sense frequency
  • For WN, hard to beat Most Frequent Sense baseline (?!)
  • For major real-world named entities: consider "Obama", "Trump"
  • This task is also called "Entity Linking"