Tagging, Part 2

CS 490A, Fall 2021

https://people.cs.umass.edu/~brenocon/cs490a_f21/

Laure Thompson and Brendan O'Connor

College of Information and Computer Sciences

University of Massachusetts Amherst

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)?

Types:		**WSJ**		**Brown**	
Unambiguous	(1 tag)	44,432	(**86%**)	45,799	(**85%**)
Ambiguous	(2+ tags)	7,025	(**14%**)	8,050	(**15%**)

Tokens:					
Unambiguous	(1 tag)	577,421	(**45%**)	384,349	(**33%**)
Ambiguous	(2+ tags)	711,780	(**55%**)	786,646	(**67%**)

Most words types are unambiguous ...

But not so for *tokens*!

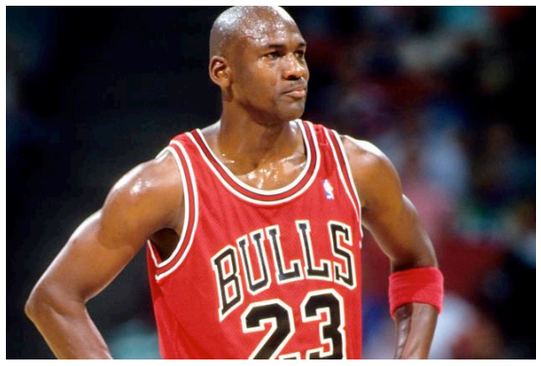
- Ambiguous wordtypes tend to be the common ones.
  - I know **that** he is honest = IN (relativizer)
  - Yes, **that** play was nice = DT (determiner)
  - You can't go **that** far = RB (adverb)

POS Tagging: baseline

- Baseline: most frequent tag. 92.7% accuracy
  - Simple baselines are very important to run!
- Why so high?
  - Many ambiguous words have a skewed distribution of tags
  - Credit for easy things like punctuation, “the”, “a”, etc.
- Is this actually that high?
  - I get 0.918 accuracy for token tagging
  - ...but, 0.186 whole-sentence accuracy (!)

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?"

[slide: SLP3]

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

[slide: SLP3]

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

[slide: SLP3]

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

[slide: SLP3]

Collocational features

- Position-specific information about the words and collocations in window
- guitar and bass player stand

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

```
[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

[slide: SLP3]

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 theChicago 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"

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 span identification to token-level tagging?

BIO tagging

Goal: represent two spans

Barack Obama Michelle Obama were ...

NAME vs O doesn't work

	N	N	N	N	O
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.

[slide made after lecture]

Features for NER/POS

- Word-based features
  - Word itself
  - Word shape
  - Contextual 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 (in ~2 weeks)

Gazetteers example

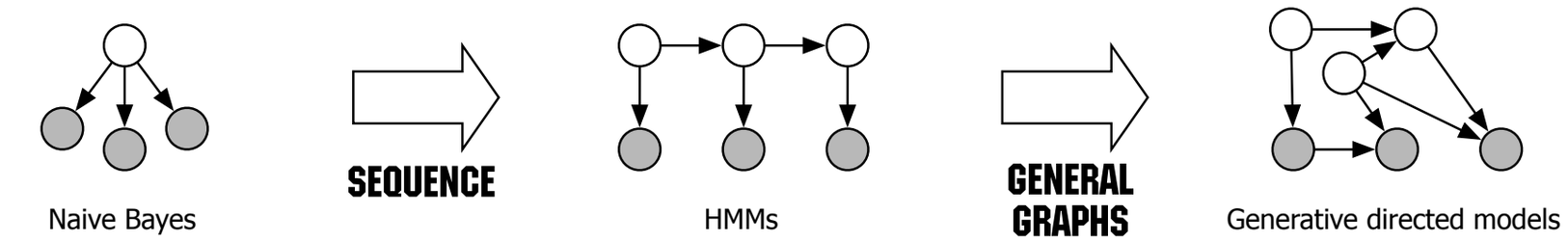
1)**People**: *people, births, deaths*. Extracts 494,699 Wikipedia titles and 382,336 redirect links. 2)**Organizations**: *cooperatives, federations, teams, clubs, departments, organizations, organisations, banks, legislatures, record labels, constructors, manufacturers, ministries, ministers, military 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 20,176 titles and 15,182 redirects.

[Ratinov and Roth 2009]

Feature-based sequence modeling

- Independent logistic regression
- Conditional random fields
- You should know what they are, but we'll only talk about a few details

Log-linear models (NB, LogReg, HMM, CRF...)



[Diagram: Sutton and McCallum 2005]

Figure 1.2 Diagram of the relationship between naive Bayes, logistic regression, HMMs, linear-chain CRFs, generative models, and general CRFs.

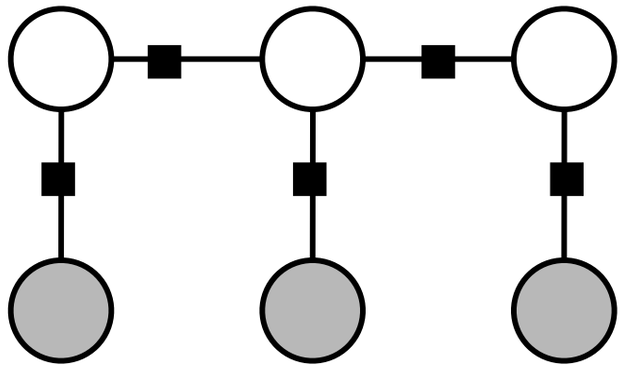
- x: Text Data
- y: Proposed class or sequence
- θ: Feature weights (model parameters)
- f(x,y): Feature extractor, produces feature vector

$$p(y|x) = \frac{1}{Z} \exp\left(\underbrace{\theta^\mathsf{T} f(x,y)}_{G(y)}\right)$$

Decision rule:

$$\arg \max_{y^* \in outputs(x)} G(y^*)$$

Conditional Random Fields



Linear-chain CRFs

$$p(\vec{y} \mid \vec{x}) = \frac{1}{Z} \prod_{t=1}^{N} \exp(\alpha_{y_t, y_{t+1}}) \ \exp(\beta' \vec{f}(y_t, x_t))$$

- Whole-sequence features decompose into
  - LOCAL (in LogReg): For each token, features for what tag to predict
    - Can use features from neighboring words
  - STRUCTURAL (new in CRF): features for *pairs* of tags
    - e.g. Adj-Noun is a likely pair. Det-Det is unlikely.
- If structural features are only about neighboring tags (Markov property), fast algorithms exist to make predictions (Viterbi) and help do learning

Conditional Random Fields

- We'll skip over the modeling details in this class
- If you want to use, there are easy-to-use software frameworks for them (e.g. CRFSuite):
  - You provide a feature vector per token
  - CRFSuite handles features for tag bigrams or trigrams
  - Typically a CRF can improve accuracy a few percentage points, compared to independent logistic regression

- CRFs have rich theory and are strongly related to Hidden Markov Models
  - Covered more in Ling 492B (comp ling) and CS 688 (probabilistic graphical models)