CS 4650/7650, Lecture 9
Part of Speech Tagging

Jacob Eisenstein

September 17, 2013

1 Review of Finite state transducers

FSAs and WFSAs apply to single strings. FSTs and WFSTs apply to pairs of string.

	acceptor	transducer
unweighted	FSA: $\Sigma^* \rightarrow \{0, 1\}$	WFSA: $\Sigma^* \rightarrow \mathbb{R}$
weighted	FST: $\Sigma^* \rightarrow \Sigma^*$	WFST: $\Sigma^* \rightarrow \langle \Sigma^*, \mathbb{R} \rangle$

We saw how finite-state composition can create powerful NLP systems out of simple, modular components.

- For example, we can compose a translation model (one state!) and a bigram language model (V states) to create a finite-state translation machine.
- If we compose the translation machine with a **chain FSA** representing the "input", we get a WFSA whose shortest path is the best translation of the input.
- We didn't talk about algorithms for composition, but the formal definition is

$$(T_1 \circ T_2)(x, y) = \bigoplus_z T_1(x, z) \otimes T_2(z, y). \quad (1)$$

  In other words, to compute the score of $\langle x, y \rangle$ in the composed machine, we take the semiring addition over all strings z, for which we compute the extension of the score of $\langle x, z \rangle$ in T_1 with the score of $\langle z, y \rangle$ in T_2.

- For example, the language model only has edges for $\langle s, s \rangle$, with score $\log P(s_i|s_{i-1})$. The translation model has edges for all $\langle s, t \rangle$, with score $\log P(t|s)$. So the composed machine must have edges for all $\langle s, t \rangle$, with score $\log P(t|s) \otimes \log P(s_i|s_{i-1})$.
- In the chain FSA, each edge takes exactly one string t_i, with score $\overline{1}$. So the composed machine is a WFSA, with edges for each possible s_i, each having score $\log P(t_i|s_i) \otimes \log P(s_i|s_{i-1}) \otimes \overline{1}$. This structure is known as a **trellis**.

Let's keep this in mind and talk about some linguistics. We'll come back to it.

2 Parts of speech

Words can be grouped into rough classes based on syntax.

- Why is *colorless green ideas sleep furiously* more acceptable than *ideas colorless furiously green sleep*?
- Why is *teacher strikes idle children* ambiguous?

In both examples, word classes can provide an explanation.

- Word classes have strong ordering constraints:
  - J J N V R is likely
  - N J R J V is unlikely (why?)
- Ambiguity about word class leads to very different interpretations:
  - N N V N
  - N V J N (ouch!)

So clearly we have intuitions about a few parts-of-speech already: noun, verb, adjective, adverb. The J&M optional text describes these as the four major **open** word classes, although apparently not all languages have all of them.

What other word classes are there?

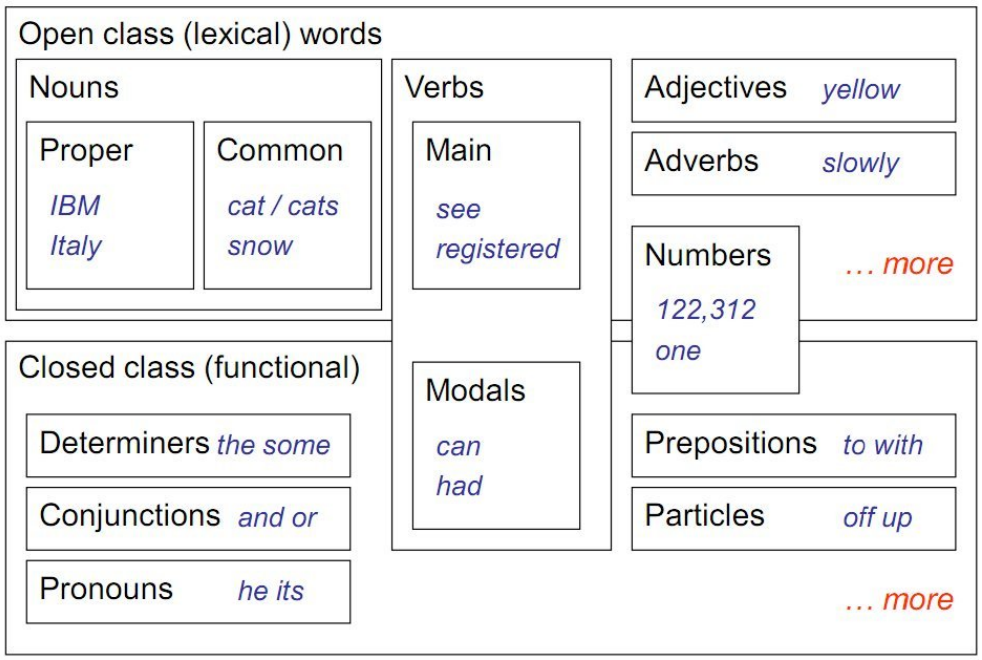
- The Penn Treebank defined a set of 45 POS tags.
- The Brown corpus defined a set of 87 POS tags.
- Petrov, Das, and McDonald (at Google) defined a universal set of 12 tags!

Which is right? Let's look at some data.

> I met a traveller from an antique land
> Who said: Two vast and trunkless legs of stone
> Stand in the desert. Near them, on the sand,
> Half sunk, a shattered visage lies, whose frown,
> And wrinkled lip, and sneer of cold command,
> Tell that its sculptor well those passions read
> Which yet survive, stamped on these lifeless things,
> The hand that mocked them and the heart that fed:
> And on the pedestal these words appear:
> "My name is Ozymandias, king of kings:
> Look on my works, ye Mighty, and despair!"
> Nothing beside remains. Round the decay
> Of that colossal wreck, boundless and bare
> The lone and level sands stretch far away.

- *I*: noun, pronoun, personal pronoun (PRP)
- *met*: verb, past tense verb (VBD)
- *a*: determiner, indefinite article (DET)
- *traveller*: noun (NN)
- *from*: preposition (IN)
- *an*: determiner (DET)
- *antique*: adjective (JJ)
- *land*: noun (NN)
- *who*: noun, pronoun, wh-pronoun
- *said*: verb, past tense verb (VBD)
- *two*: cardinal number (CD)

- *vast*: adjective (JJ)
- *and*: coordinating conjunction (CC)
- *trunkless*: adjective (JJ)
- *legs*: noun, plural (NNS)
- *of*: preposition (IN)
- *stone*: stone (NN)
- *stand*: verb, non-3SP
- *in*: preposition (IN)
- *the*: determiner, definite article (DET)
- *desert*: noun (NN)
- *near*: adverb (RB)
- *them*: personal pronoun (PRP)
- ...
- *half*: adverb (RB)
- *sunk*: verb, past-participle (VBN)
- ...
- *tell*: verb, non-3SP
- *that*: preposition (IN)
- *its*: possessive pronoun (PRP$)
- *sculptor*: noun (NN)
- *well*: adverb (RB)
- *those*: determiner, plural, proximal (DT)
- *passions*: noun (NNS)
- *read*: verb, past tense (VBD)
- ...



Details about parts-of-speech

- **Nouns** describe entities and concepts
  - **Proper nouns** name specific people and entities: *Georgia Tech*, *Janet*, *Buddhism*. In English, they're usually capitalized. PTB tags: NNP (singular), NNPS (plural)
  - **Common nouns** cover everything else. In English, they're often preceded by articles, e.g. *the book*, *the university*. Common nouns decompose into
    - **Count nouns** have a plural and need an article in the singular, *dogs*, *the dog*.
    - **Mass nouns** don't have a plural and don't an article in the singular, *snow is cold, gas is expensive*
  - **Pronouns** refer to specific noun phrases or entities or events.
    - **Personal pronouns** refer to people or entities: *you, she, I, it, me*. PTB tag: PRP
    - **Possessive pronouns** are pronouns that indicate possession: *your, her, my, its, one's, our*. PTB tag: PRP$

    - **Wh-pronouns** are used in question forms (*Where are you going?*, WP) and as relative pronouns in forms like (*The girl who played with fire*)

  Unlike other nouns, the set of possible pronouns cannot be expanded! It is a **closed class**.

- **Verbs** describe activities, processes, and events, e.g. *eat, write, sleep*
  - The Penn Treebank differentiates verbs by morphology: VB (infinitive), VBD (past), VBG (present participle), VBN (past participle), VBP (present, non 3rd person-singular), VBZ (present 3rd person singular)
  - **modals** are a closed subclasses of verbs, such as (*should, can, will, must*). They get PTB tag MD
  - **copula** is *be* with a predicate, e.g. *she is hungry.* The Brown Tagset distinguishes copula, but PTB doesn't.
  - **auxiliary** verbs include *be, have, will* to form complex tenses, e.g. *we will have done it twice.*
    - Also includes *do* in questions and negation, e.g. *did you eat yet?*. Apparently this is from Welsh, which was spoken in England before the Anglo-Saxons invaded; *do* doesn't function this way in German.
    - The Brown corpus has special tags for HAVE and DO, but the PTB doesn't.

- **Adjectives** describe properties of entities: *antique, vast, trunkless*
  - **Attributive use**: *an antique land*
  - **Predicative use**: *the land was antique*
  - **Gradable adjectives** (*big*) have **comparative form** (*bigger*) and **superlative form** (*biggest*)
  - Can you think of an adjective that is not gradable?
  - With *big*, we can move to comparative form by adding the suffix *-est*. This is an example of agglutinative morphology. Can you think of an adjective in English where the relationship is not agglutinative? (rather, it's fusional). How about *good*, *better*, *best*?

  - PTB tags: JJ, JJR, JJS

- **Adverbs** describe properties of events.
  - Manner: *slowly, slower, fast, hesitantly*
  - Degree: *extremely, very, highly*
  - Directional and locative: *here, downstairs, near*
  - Temporal: *yesterday, Monday*
  - Besides verbs, adverbs may also modify sentences, adjectives, or other adverbs. Apparently, the very ill man walks extremely slowly
  - Adverbs may also be gradable. Tags: RB, RBR, RBS

- **Prepositions** are a closed class of words that can come before noun phrases, forming a prepositional phrase that relates the noun phrase to something else in the sentence.
  - *I eat sushi WITH soy sauce*
  - *I eat sushi WITH chopsticks*
  - *To* gets its own tag TO, because it forms the infinitive with bare form verbs (VB), e.g. *I want to eat*
  - Everything else is tagged IN in the PTB.

- **Coordinating conjunctions** join two elements,
  - *vast and trunkless legs*
  - *She eats burgers but she drinks soda*
  - PTB tag: CC

- **Subordinating conjunctions** introduce a subordinate clause, e.g. *She thinks THAT Chomsky is wrong.* PTB tag: annotIN

- **Particles** are oddball words that come with verbs and can change their meaning to a new **phrasal verb**, e.g., *come ON*, *he brushed himself OFF*, *let's check OUT that new restaurant.* They are a closed class, PTB tag RP.

- **Determiners** are a closed class of words that precede noun phrases.

  - Articles: *the, an, a*
  - Demonstratives: *this, these, that*
  - Quantifiers: *some, every, few*
  - Wh-determiners, *WHICH burger should I choose?*
  - PTB tag: DT
- **Oddballs**
  - **Existential there**, e.g. *There is no way out of here*, gets its own tag, EX.
  - So does the possessive ending *'s*, POS
  - So do numbers (CD), list items (LS), commas, and other non-alphabetic symbols.

3 Part of speech tagging

Part of speech tags relate to a number of other linguistic phenomena:

- Lexical semantics: *can*/V vs *can*/N, *teacher strikes children*, etc
- Pronunciation: *inSULT* vs *INsult*, *conTENT* vs *CONtent*
- Translation: *park*/v → *garer*, *park*/N → *parque*
- NP chunking: `grep {JJ | NN}* {NN | NNS}`

POS tagging is a useful preprocessing step for downstream applications.
So how can we build an automatic POS tagger?

- Observation 1: it's "easy."
  - 60% of word types have only one possible POS tag (in English).
  - If you choose the majority POS tag for each token, you get 90% right.
- Observation 2: it's not easy: a few words have a lot of possible POS tags
  - *We're taking it **back**/RB*

- *The shirt off my **back**/NN*
- *Go **back**/RP where you belong*
- *If you challenge him, I'll **back**/VBP you*
- *The **back**/JJ roads are safer*

- Observation 3: 90% is not actually very good. $0.9^{10} \approx .3$, so you will only get 30% of ten-word sentences correct. Sentences have exponentially many possible POS sequences:

VBD		VB			
VBN	VBZ	VBP	VBZ		
NNP	NNS	NN	NNS	CD	NN
fed	*raises*	*interest*	*rates*	*0.5*	*percent*

Anyway, let's look at a tougher poem, Jabberwocky:

'Twas brillig, and the slithy toves
Did gyre and gimble in the wabe:
All mimsy were the borogoves,
And the mome raths outgrabe

Forget *twas*. What about *slithy*? Can you guess the POS? What about *toves*? You don't know these words. What information are you using to guess?

- Word identity: you do know that *and* is CC and *the* is DET
- **Context**
  - JJ NN is likely
  - NN JJ is unlikely
- **Morphology**
  - *-s* → noun or verb
  - *-able* → adjective (98% of the time!)
  - *-ly* → adverb
  - *un-* → adjective or verb

– (not rules, just hints)

Let's put morphology on hold for a minute.

- Suppose we have an annotated corpus, with tagged sentences, $\langle \boldsymbol{w}_{1:N_t}, \boldsymbol{y}_{1:N_t} \rangle_{1:T}$.
- Then we could estimate the likelihood of a word given a tag,

$$P(w|y) = \frac{\text{count}(w, y)}{\text{count}(y)} \tag{2}$$

  As always, smoothing is possible...

- Given this same annotated corpus, we could also compute $P(y_n|y_{n-1})$, a sort of language model over tags.

$$P(y_n|y_{n-1}) = \frac{\text{count}(y_{n-1}, y_n)}{\text{count}(y_{n-1})} \tag{3}$$

- Let's combine these ideas via a **generative story**
  - For word n, draw tag $y_n \sim \text{Categorical}(\theta_{y_{n-1}})$
  - Then draw word $w_n \sim \text{Categorical}(\phi_{y_n})$

  We've built a generative model that explains our observations $\boldsymbol{w}$ through a bigram generative model over the tags.

- Under this model, we can compute

$$\begin{aligned} P(\boldsymbol{y}|\boldsymbol{w}) \propto & P(\boldsymbol{w}, \boldsymbol{y}) && (4) \\ & P(\boldsymbol{w}|\boldsymbol{y})P(\boldsymbol{y}) && (5) \\ & \prod_n^N P(w_n|y_n)P(y_n|y_{n-1}) && (6) \end{aligned}$$

- This is a **hidden Markov model**. It's Markov because the probability of y_n depends only on y_{n-1} and not any of the previous history. It's hidden because y_n is unknown when we decode.
- We can treat this as a special case of finite state transduction. Can you see how?