Why Language is Hard: Structure and Predictions

Introduction to Data Science Algorithms
Jordan Boyd-Graber and Michael Paul
SLIDES ADAPTED FROM RAY MOONEY (NOT ON FINAL!)
POS Tagging: Task Definition

• Annotate each word in a sentence with a part-of-speech marker.
• Lowest level of syntactic analysis.

John saw the saw and decided to take it to the table
NNP VBD DT NN CC VBD TO VB PRP IN DT NN
Tag Examples

- **Noun (person, place or thing)**
  - Singular (NN): dog, fork
  - Plural (NNS): dogs, forks
  - Proper (NNP, NNPS): John, Springfields

- **Personal pronoun (PRP):** I, you, he, she, it

- **Wh-pronoun (WP):** who, what

- **Verb (actions and processes)**
  - Base, infinitive (VB): eat
  - Past tense (VBD): ate
  - Gerund (VBG): eating
  - Past participle (VBN): eaten
  - Non 3rd person singular present tense (VBP): eat
  - 3rd person singular present tense: (VBZ): eats
  - Modal (MD): should, can
  - To (TO): to (to eat)
Ambiguity

“Like” can be a verb or a preposition
- I like/VBP candy.
- Time flies like/IN an arrow.

“Around” can be a preposition, particle, or adverb
- I bought it at the shop around/IN the corner.
- I never got around/RP to getting a car.
- A new Prius costs around/RB $25K.
How hard is it?

• Usually assume a separate initial tokenization process that separates and/or disambiguates punctuation, including detecting sentence boundaries.

• Degree of ambiguity in English (based on Brown corpus)
  ○ 11.5% of word types are ambiguous.
  ○ 40% of word tokens are ambiguous.

• Average POS tagging disagreement amongst expert human judges for the Penn treebank was 3.5%

• Based on correcting the output of an initial automated tagger, which was deemed to be more accurate than tagging from scratch.

• Baseline: Picking the most frequent tag for each specific word type gives about 90% accuracy 93.7% if use model for unknown words for Penn Treebank tagset.
What about classification / feature engineering?

- Just predict the most frequent class: 0.38 accuracy
- Can get to around 60% accuracy by adding in dictionaries, prefix / suffix features
What about classification / feature engineering?

- Just predict the most frequent class: 0.38 accuracy
- Can get to around 60% accuracy by adding in dictionaries, prefix / suffix features
- Can get to 95% accuracy if you take correlated predictions into account
A more fundamental problem …

- If you have a noun, it’s more likely to be preceded by an adjective
- Determiners are followed by either a noun or an adjective
- Determiners don’t follow each other
Parameter Definition

Assume $K$ parts of speech, a lexicon size of $V$, a series of observations \{\(x_1, \ldots, x_N\)\}, and a series of unobserved states \(\{z_1, \ldots, z_N\}\).

- $\pi$ A distribution over start states (vector of length $K$): $\pi_i = p(z_1 = i)$
- $\theta$ Transition matrix (matrix of size $K$ by $K$): $\theta_{i,j} = p(z_n = j | z_{n-1} = i)$
- $\beta$ An emission matrix (matrix of size $K$ by $V$): $\beta_{j,w} = p(x_n = w | z_n = j)$
Assume $K$ parts of speech, a lexicon size of $V$, a series of observations \( \{x_1, \ldots, x_N\} \), and a series of unobserved states \( \{z_1, \ldots, z_N\} \).

\[\pi\] A distribution over start states (vector of length $K$): \( \pi_i = p(z_1 = i) \)

\[\theta\] Transition matrix (matrix of size $K$ by $K$): \( \theta_{i,j} = p(z_n = j | z_{n-1} = i) \)

\[\beta\] An emission matrix (matrix of size $K$ by $V$): \( \beta_{j,w} = p(x_n = w | z_n = j) \)

Two problems: How do we move from data to a model? (Estimation) How do we move from a model and unlabeled data to labeled data? (Inference)
Cartoon

P(PropNoun Verb Det Noun) = 0.4*0.8*0.25*0.95*0.1=0.0076
Reminder: How do we estimate a probability?

- For a multinomial distribution (i.e. a discrete distribution, like over words):

\[ \theta_i = \frac{n_i + \alpha_i}{\sum_k n_k + \alpha_k} \]  

(1)

- \( \alpha_i \) is called a smoothing factor, a pseudocount, etc.
Reminder: How do we estimate a probability?

- For a multinomial distribution (i.e. a discrete distribution, like over words):
  \[ \theta_i = \frac{n_i + \alpha_i}{\sum_k n_k + \alpha_k} \] (1)

- \( \alpha_i \) is called a smoothing factor, a pseudocount, etc.

- When \( \alpha_i = 1 \) for all \( i \), it’s called “Laplace smoothing” and corresponds to a uniform prior over all multinominal distributions.
here come old flattop
MOD V MOD N

a crowd of people stopped and stared
DET N PREP N V CONJ V

gotta get you into my life
V V PRO PREP PRO V

and I love her
CONJ PRO V PRO
Training Sentences

\[ x \quad \text{here} \quad \text{come} \quad \text{old} \quad \text{flattop} \]
\[ \text{MOD} \quad \text{V} \quad \text{MOD} \quad \text{N} \]

\[ \text{a} \quad \text{crowd} \quad \text{of} \quad \text{people} \quad \text{stopped} \quad \text{and} \quad \text{stared} \]
\[ \text{DET} \quad \text{N} \quad \text{PREP} \quad \text{N} \quad \text{V} \quad \text{CONJ} \quad \text{V} \]

\[ \text{gotta} \quad \text{get} \quad \text{you} \quad \text{into} \quad \text{my} \quad \text{life} \]
\[ \text{V} \quad \text{V} \quad \text{PRO} \quad \text{PREP} \quad \text{PRO} \quad \text{V} \]

\[ \text{and} \quad \text{I} \quad \text{love} \quad \text{her} \]
\[ \text{CONJ} \quad \text{PRO} \quad \text{V} \quad \text{PRO} \]
Training Sentences

\[
x \quad \text{here} \quad \text{come} \quad \text{old} \quad \text{flattop}
\]
\[
z \quad \text{MOD} \quad \text{V} \quad \text{MOD} \quad \text{N}
\]

\[
a \quad \text{a} \quad \text{crowd} \quad \text{of} \quad \text{people} \quad \text{stopped} \quad \text{and} \quad \text{stared}
\]
\[
\text{DET} \quad \text{N} \quad \text{PREP} \quad \text{N} \quad \text{V} \quad \text{CONJ} \quad \text{V}
\]

\[
gotta \quad \text{get} \quad \text{you} \quad \text{into} \quad \text{my} \quad \text{life}
\]
\[
\text{V} \quad \text{V} \quad \text{PRO} \quad \text{PREP} \quad \text{PRO} \quad \text{V}
\]

\[
\text{and} \quad \text{I} \quad \text{love} \quad \text{her}
\]
\[
\text{CONJ} \quad \text{PRO} \quad \text{V} \quad \text{PRO}
\]
### Initial Probability $\pi$

<table>
<thead>
<tr>
<th>POS</th>
<th>Frequency</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOD</td>
<td>1.1</td>
<td>0.234</td>
</tr>
<tr>
<td>DET</td>
<td>1.1</td>
<td>0.234</td>
</tr>
<tr>
<td>CONJ</td>
<td>1.1</td>
<td>0.234</td>
</tr>
<tr>
<td>N</td>
<td>0.1</td>
<td>0.021</td>
</tr>
<tr>
<td>PREP</td>
<td>0.1</td>
<td>0.021</td>
</tr>
<tr>
<td>PRO</td>
<td>0.1</td>
<td>0.021</td>
</tr>
<tr>
<td>V</td>
<td>1.1</td>
<td>0.234</td>
</tr>
</tbody>
</table>

Remember, we’re taking MAP estimates, so we add 0.1 (arbitrarily chosen) to each of the counts before normalizing to create a probability distribution. This is easy; one sentence starts with an adjective, one with a determiner, one with a verb, and one with a conjunction.
Training Sentences

here come old flattop
MOD V MOD N

a crowd of people stopped and stared
DET N PREP N V CONJ V

gotta get you into my life
V V PRO PREP PRO N

and I love her
CONJ PRO V PRO
Training Sentences

here come old flattop
MOD V MOD N

a crowd of people stopped and stared
DET N PREP N V CONJ V

gotta get you into my life
V V PRO PREP PRO N

and I love her
CONJ PRO V PRO
Training Sentences

here come old flattop

MOD V MOD N

a crowd of people stopped and stared

DET N PREP N V CONJ V

gotta get you into my life

V V PRO PREP PRO N

and I love her

CONJ PRO V PRO
We can ignore the words; just look at the parts of speech. Let’s compute one row, the row for verbs.

We see the following transitions: $V \rightarrow \text{MOD}$, $V \rightarrow \text{CONJ}$, $V \rightarrow V$, $V \rightarrow \text{PRO}$, and $V \rightarrow \text{PRO}$.

<table>
<thead>
<tr>
<th>POS</th>
<th>Frequency</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOD</td>
<td>1.1</td>
<td>0.193</td>
</tr>
<tr>
<td>DET</td>
<td>0.1</td>
<td>0.018</td>
</tr>
<tr>
<td>CONJ</td>
<td>1.1</td>
<td>0.193</td>
</tr>
<tr>
<td>N</td>
<td>0.1</td>
<td>0.018</td>
</tr>
<tr>
<td>PREP</td>
<td>0.1</td>
<td>0.018</td>
</tr>
<tr>
<td>PRO</td>
<td>2.1</td>
<td>0.368</td>
</tr>
<tr>
<td>V</td>
<td>1.1</td>
<td>0.193</td>
</tr>
</tbody>
</table>

And do the same for each part of speech ...
Training Sentences

here come old flattop
MOD V MOD N

da crowd of people stopped and stared
DET N PREP N V CONJ V

gotta get you into my life
V V PRO PREP PRO N

and I love her
CONJ PRO V PRO
here come old flattop
MOD V MOD N

a crowd of people stopped and stared
DET N PREP N V CONJ V

gotta get you into my life
V V PRO PREP PRO N

and I love her
CONJ PRO V PRO
Let’s look at verbs . . .

<table>
<thead>
<tr>
<th>Word</th>
<th>a</th>
<th>and</th>
<th>come</th>
<th>crowd</th>
<th>flattop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>0.1</td>
<td>0.1</td>
<td>1.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Probability</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.1375</td>
<td>0.0125</td>
<td>0.0125</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word</th>
<th>get</th>
<th>gotta</th>
<th>her</th>
<th>here</th>
<th>i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>1.1</td>
<td>1.1</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Probability</td>
<td>0.1375</td>
<td>0.1375</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.0125</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word</th>
<th>into</th>
<th>it</th>
<th>life</th>
<th>love</th>
<th>my</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>1.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Probability</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.1375</td>
<td>0.0125</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word</th>
<th>of</th>
<th>old</th>
<th>people</th>
<th>stared</th>
<th>stopped</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>0.1</td>
<td>0.1</td>
<td>0.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>Probability</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.0125</td>
<td>0.1375</td>
<td>0.1375</td>
</tr>
</tbody>
</table>