Unsupervised Learning on an Approximate Corpus

Jason Smith, Jason Eisner
Learning from n-grams

Sentences:

<table>
<thead>
<tr>
<th></th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>time flies like an arrow</td>
<td>20</td>
</tr>
<tr>
<td>fruit flies like an orange</td>
<td>3</td>
</tr>
<tr>
<td>your plane flies like an ostrich</td>
<td>2</td>
</tr>
</tbody>
</table>

n-grams:

<table>
<thead>
<tr>
<th></th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>time flies like</td>
<td>20</td>
</tr>
<tr>
<td>fruit flies like</td>
<td>3</td>
</tr>
<tr>
<td>flies like an</td>
<td>25</td>
</tr>
<tr>
<td>like an arrow</td>
<td>20</td>
</tr>
<tr>
<td>your plane flies</td>
<td>2</td>
</tr>
</tbody>
</table>

...
Contributions

• Learning from a finite-state distribution over sentences
  • e.g. an n-gram language model over sentences, instead of individual sentences
• Why?
  • Original corpus unavailable
  • Speed (learning from compressed data)
  • (Fundamental question about weighted grammars)
• Exact and approximate solutions
Task

- HMM POS tagging (Merialdo 94)
- Many approaches build off of EM
Previous Work

Gender

Count the possessive pronouns following a word after connectives or verbs. [Bergsma 2005]

Pattern: Word CC/V* PRP$

<table>
<thead>
<tr>
<th></th>
<th>its</th>
<th>his</th>
<th>her</th>
<th>their</th>
<th>my</th>
</tr>
</thead>
<tbody>
<tr>
<td>star</td>
<td>2805</td>
<td>2409</td>
<td>1594</td>
<td>250</td>
<td>247</td>
</tr>
<tr>
<td>Hollywood star</td>
<td>0</td>
<td>29</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>variable star</td>
<td>57</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

• (slide taken from Lin et al., 2009)
Previous Work

Gender

Count the possessive pronouns following a word after connectives or verbs. [Bergsma 2005]

Pattern: Word CC/V* PRP$

<table>
<thead>
<tr>
<th></th>
<th>its</th>
<th>his</th>
<th>her</th>
<th>their</th>
<th>my</th>
</tr>
</thead>
<tbody>
<tr>
<td>star and</td>
<td>2805</td>
<td>2409</td>
<td>1594</td>
<td>250</td>
<td>247</td>
</tr>
<tr>
<td>Hollywood star and</td>
<td>0</td>
<td>29</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>variable star and</td>
<td>57</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

• (slide taken from Lin et al., 2009)
Previous Work

**Gender**

- Count the possessive pronouns following a word after connectives or verbs. [Bergsma 2005]

<table>
<thead>
<tr>
<th></th>
<th>its</th>
<th>his</th>
<th>her</th>
<th>their</th>
<th>my</th>
</tr>
</thead>
<tbody>
<tr>
<td>star and</td>
<td>2805</td>
<td>2409</td>
<td>1594</td>
<td>250</td>
<td>247</td>
</tr>
<tr>
<td>Hollywood star and</td>
<td>0</td>
<td>29</td>
<td>13</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>variable star and</td>
<td>57</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- (slide taken from Lin et al., 2009)
Motivation: learning from n-grams

Full context:

- time flies like an arrow
- fruit flies like an orange
Motivation: learning from n-grams

Full context:

<table>
<thead>
<tr>
<th>N</th>
<th>V</th>
<th>Adv</th>
<th>Det</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>flies</td>
<td>like</td>
<td>an</td>
<td>arrow</td>
</tr>
<tr>
<td>Adj</td>
<td>N</td>
<td>V</td>
<td>Det</td>
<td>N</td>
</tr>
<tr>
<td>fruit</td>
<td>flies</td>
<td>like</td>
<td>an</td>
<td>orange</td>
</tr>
</tbody>
</table>

Local n-gram context:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>flies</td>
<td>like</td>
<td>an</td>
<td>?</td>
</tr>
</tbody>
</table>
Motivation: learning from n-grams

Local n-gram context:


? flies like an ?

Overlapping n-grams:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>flies</td>
<td>like</td>
<td>20</td>
</tr>
<tr>
<td>fruit</td>
<td>flies</td>
<td>like</td>
<td>3</td>
</tr>
<tr>
<td>plane</td>
<td>flies</td>
<td>like</td>
<td>2</td>
</tr>
</tbody>
</table>
Motivation: learning from n-grams

Local n-gram context:


- time 80%
- fruit 12%
- plane 8%

Overlapping n-grams:

<table>
<thead>
<tr>
<th>Count</th>
<th>time</th>
<th>fruit</th>
<th>plane</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>flies</td>
<td>flies</td>
<td>flies</td>
</tr>
<tr>
<td>20</td>
<td>like</td>
<td>like</td>
<td>like</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Motivation: learning from n-grams

Local n-gram context:

\[
\begin{align*}
\text{N} & \quad \text{V} & \quad \text{Adv} & \quad \text{Det} & \quad ? \\
\text{Adj} & \quad \text{N} & \quad \text{V} & \quad & \\
\text{time} & \quad \text{flies} & \quad \text{like} & \quad \text{an} & \quad ? \\
\text{fruit} & \quad & \quad & & \\
\text{plane} & \quad & \quad & & \\
\end{align*}
\]

Overlapping n-grams:

<table>
<thead>
<tr>
<th>time 80%</th>
<th>fruit 12%</th>
<th>plane 8%</th>
</tr>
</thead>
<tbody>
<tr>
<td>flies</td>
<td>flies</td>
<td>flies</td>
</tr>
<tr>
<td>like</td>
<td>like</td>
<td>like</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>2</td>
</tr>
</tbody>
</table>

Counts
Exploit Overlapping n-grams

<table>
<thead>
<tr>
<th></th>
<th>Counts</th>
</tr>
</thead>
<tbody>
<tr>
<td>time flies like</td>
<td>20</td>
</tr>
<tr>
<td>flies like an</td>
<td>25</td>
</tr>
<tr>
<td>like an arrow</td>
<td>20</td>
</tr>
<tr>
<td>fruit flies like</td>
<td>3</td>
</tr>
<tr>
<td>like an orange</td>
<td>3</td>
</tr>
<tr>
<td>your plane flies</td>
<td>2</td>
</tr>
<tr>
<td>plane flies like</td>
<td>2</td>
</tr>
<tr>
<td>like an ostrich</td>
<td>2</td>
</tr>
</tbody>
</table>
Exploit Overlapping n-grams

n-gram language model!
N-gram language models

\[ p(\text{orange}|\text{like an}) = 0.12 \]
\[ p(\text{arrow}|\text{like an}) = 0.80 \]
\[ p(\text{ostrich}|\text{like an}) = 0.08 \]
N-gram language models

\[ p(\text{orange} | \text{like an}) = 0.12 \]
\[ p(\text{arrow} | \text{like an}) = 0.80 \]
\[ p(\text{ostrich} | \text{like an}) = 0.07 \]
\[ p(-\text{backoff} | \text{like an}) = 0.01 \]

- **fruit flies**
- **time flies**
- **plane flies**

\[ p(\text{orange} | \text{like an}) = 0.12 \]
\[ p(\text{arrow} | \text{like an}) = 0.80 \]
\[ p(\text{ostrich} | \text{like an}) = 0.07 \]
\[ p(-\text{backoff} | \text{like an}) = 0.01 \]
N-gram language models

- fruit flies
- time flies
- plane flies
- flies like
- like
- an
- like an
- an orange
- an arrow
- an ostrich
N-gram language models $c(w)$:
- probability distribution over sentences
- “approximate corpus”
N-gram language models
\( c(w) \):
- probability distribution over sentences
- “approximate corpus”

\( n_c = 3 \)

- fruit flies
- time flies
- plane flies
- flies like
- like
- like
- an
- like an
- an
- orange
- arrow
- an arrow
- ostrich
- an ostrich
Task

• HMM POS tagging (Merialdo 94)
• Many approaches build off of EM

$$\max_{w \in \text{corpus}} \sum \frac{1}{n} \log \sum \limits_{t} p(t, w)$$

$$\max_{w \in \text{corpus}} \sum \limits_{w \in \text{corpus}} c(w) \log \sum \limits_{t} p(t, w)$$
HMM Tagging

Sentence: time flies like an arrow
HMM Tagging

Sentence: time flies like an arrow

\[ p(\text{Tag}|\text{Last Tag}) \]

<table>
<thead>
<tr>
<th>Det</th>
<th>N</th>
<th>V</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.3</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>0.1</td>
<td>0.6</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

...
HMM Tagging

Sentence: *time flies like an arrow*

\[
p(\text{Tag} | \text{Last Tag})
\]

<table>
<thead>
<tr>
<th></th>
<th>Det</th>
<th>N</th>
<th>V</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Det</td>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>0.8</td>
<td>0.3</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>0.1</td>
<td>0.6</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
p(\text{Word} | \text{Tag})
\]

<table>
<thead>
<tr>
<th></th>
<th>Det</th>
<th>N</th>
<th>V</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>0.01</td>
<td>0.3</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>flies</td>
<td>0.01</td>
<td>0.2</td>
<td>0.2</td>
<td></td>
</tr>
<tr>
<td>an</td>
<td>0.33</td>
<td>0.01</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HMM Tagging

Sentence: time flies like an arrow

\[
p(t, w) : n_p = 2
\]

\[n_c : c(w)\text{'s word context window}
\]

\[n_p : p(t, w)\text{'s tag context window}
\]
Supervised learning: HMM

N  time  V  flies  Adv  like  Det  an  N  arrow
Supervised learning: HMM

transition counts: estimating $p(\text{Tag}|\text{Last Tag})$

time flies like an arrow
Supervised learning: HMM

Emission counts: estimating $p(Word|Tag)$

- N: time
- V: flies
- Adv: like
- Det: an
- N: arrow
What if someone tagged our n-grams?

c(\textbf{w}):
What if someone tagged our n-grams?

\( c(w) : \)

```
fruit flies
```

```
time flies
```

```
plane flies
```

```
N  V  Adv  Det  N
time flies like an orange
Adj  N  V  Det  N
fruit flies like an arrow
N  V  Adv  Det  N

like
```

```
flies like
```

```
like an
```

```
an arrow
```

```
arrow
```

```
an ostrich
```

```
an ostrich
```
What if someone tagged our n-grams?

c(w) c(t, w):

Adj N fruit flies

N V flies like

Adv like

Adv like

V flies like

Det an

V like an

Det like an

Adv Det like an

Det N an orange

Det N an arrow

Det N an ostrich
What if someone tagged our n-grams?

\( c(w) \) \( c(t,w) \):

\( n_c \) : \( c(w) \)'s word context window

\( n_p \) : \( p(t,w) \)'s tag context window

\( n_q \) : \( c(t,w) \)'s tag context window

\( n_q = 3 \)
What if someone tagged our n-grams?

c(w) c(t,w):

Adj N fruit flies

V like

N flies

V like

Det an

V Det like an

... Det N orange

Det N arrow

Det N ostrich

What if someone tagged our n-grams?

transition counts: estimating \( p(\text{Tag}|\text{Last Tag}) \)
What if someone tagged our n-grams?

c(w) c(t,w): 

Adj N fruit flies V like N V flies like

N V time flies Adv like Adv like N V flies like

Adj like N V flies like Adv like Adv like N V flies like

det N V flies like det like an

Det N an orange Det N an arrow Det N an ostrich

emission counts: p(an|Det)
What if someone tagged our n-grams?

\[ c(w) \ c(t, w) : \]

- ? N fruit flies
- ? V time flies
- ? V plane flies
- V flies like
- Adv like
- ? Adv flies like
- Det an
- ? Det like an
- N orange
- N arrow
- N ostrich
- ? N an arrow
- ? N an ostrich

\[ n_q = 2 \]
What if someone tagged our n-grams?

c(w) c(t, w):

transition counts: estimating $p(\text{Tag} | \text{Last Tag})$
What if someone tagged our n-grams?

c(w), c(t, w):

- fruit flies
- time flies
- plane flies

V like

Adv like

Det an

like an

N orange

N arrow

N ostrich

What if someone tagged our n-grams?
What if someone tagged our n-grams?

\[ c(w) \cdot c(t, w) : \]

fruit flies

\[ V \]

like

flies like

Det an

like an

transition counts? must look at short paths

\[ n_q = 1 \]

an orange

an arrow

an ostrich

nq = 1

What if someone tagged our n-grams?

transition counts? must look at short paths

\[ n_q = 1 \]
Unsupervised learning

Sentence: `time flies like an arrow`

### $p(\text{Tag}|\text{Last Tag})$

<table>
<thead>
<tr>
<th></th>
<th>Det</th>
<th>N</th>
<th>V</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Det</td>
<td>0.1</td>
<td>0.1</td>
<td>0.5</td>
<td>...</td>
</tr>
<tr>
<td>N</td>
<td>0.8</td>
<td>0.3</td>
<td>0.4</td>
<td>...</td>
</tr>
<tr>
<td>V</td>
<td>0.1</td>
<td>0.6</td>
<td>0.1</td>
<td>...</td>
</tr>
</tbody>
</table>

### $p(\text{Word}|\text{Tag})$

<table>
<thead>
<tr>
<th></th>
<th>Det</th>
<th>N</th>
<th>V</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>0.01</td>
<td>0.3</td>
<td>0.1</td>
<td>...</td>
</tr>
<tr>
<td>flies</td>
<td>0.01</td>
<td>0.2</td>
<td>0.2</td>
<td>...</td>
</tr>
<tr>
<td>an</td>
<td>0.33</td>
<td>0.01</td>
<td>0.01</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
HMM Tagging Trellis

\[ p(t, w) \circ c(w): \]

Diagram showing the tagging process with states and transitions for the sentence "time flies like an arrow."
HMM Tagging Trellis

\[ p(t, w) \circ c(w) : \]

\begin{align*}
\text{Adj} & \quad \text{V} \quad \text{N} \quad \text{V} \\
\text{flies} & \quad \text{flies} \quad \text{like} \\
\text{time} & \quad \text{V} \\
\text{N} & \quad \text{Adv} \\
\text{like} & \quad \text{Det} \\
\text{an} & \quad \text{N} \\
\text{arrow} & \quad \text{N} \\
\end{align*}
HMM Tagging Trellis

\[ p(t, w) \circ c(w) : \]

\[
\max_{w \in \text{corpus}} \sum_{t} c(w) \log \sum_{t} p(t, w)
\]

\[
\text{time} \quad \text{flies} \quad \text{like} \quad \text{an} \quad \text{arrow}
\]
Let's tag our own n-grams (EM): 

\[ c(w) \cdot c(t,w) \cdot c(w)q(t|w) : \]

\[ n_q = 2 \]
Let’s tag our own n-grams (EM)

c(w) c(t,w) c(w)q(t|w):

\[ n_q = 2 \]

\[ n_c : c(w)’s \text{ word context window} \]
\[ n_p : p(t,w)’s \text{ tag context window} \]
\[ n_q : q(t|w)’s \text{ tag context window} \]
Variational EM

$$\log \sum_{t} p_\theta(t, w) = \log \sum_{t} q(t|w) \left( \frac{p_\theta(t, w)}{q(t|w)} \right)$$

$$\geq \sum_{t} q(t|w) \log \left( \frac{p_\theta(t, w)}{q(t|w)} \right)$$

$$= \mathbb{E}_{q(t|w)} [\log p_\theta(t, w) - \log q(t|w)]$$
Variational EM

\[
\log \sum_t p_\theta(t, w) = \log \sum_t q(t|w) \left( \frac{p_\theta(t, w)}{q(t|w)} \right) \\
\geq \sum_t q(t|w) \log \left( \frac{p_\theta(t, w)}{q(t|w)} \right) \\
= \mathbb{E}_{q(t|w)} \left[ \log p_\theta(t, w) - \log q(t|w) \right]
\]
\[ \mathbb{E}_c(w) \log \sum_t p_\theta(t, w) = \mathbb{E}_c(w) \log \sum_t q(t|w) \left( \frac{p_\theta(t, w)}{q(t|w)} \right) \]

\[ \geq \mathbb{E}_c(w) \sum_t q(t|w) \log \left( \frac{p_\theta(t, w)}{q(t|w)} \right) \]

\[ = \mathbb{E}_c(w) q(t|w) \left[ \log p_\theta(t, w) - \log q(t|w) \right] \]
Variational EM

\[ n_q = n_p : \text{variational bound is “tight”} \]
\[ n_q < n_p : \text{we are approximating} \]

\( n_c \): \( c(w) \)'s word context window
\( n_p \): \( p(t,w) \)'s tag context window
\( n_q \): \( q(t|w) \)'s tag context window
How to maximize this bound

\[ E_{c(w)} q(t|w) \left[ \log p_{\theta}(t, w) - \log q(t|w) \right] \]

- Updating \( p(t,w) \) (M-step): shown earlier
- Updating \( q(t|w) \) (E-step): more complex, but has a dynamic programming solution which makes use of finite-state machines
- Expectation semirings (Eisner 2002), details in paper
Experiments: EM vs. n-gram EM

• How does EM on a full corpus compare to n-gram EM on an approximate corpus?

• POS tagging accuracy and likelihood

• Standard setup for unsupervised POS tagging with a dictionary

• Reduced tag set (17 tags)

• Limited tag dictionary from WSJ (words must appear 5 times, otherwise all tags are possible)
Experiments: EM vs. n-gram EM

• n-gram EM parameter choices:
  • $n_c=5$ - $c(w)$ uses up to 5-grams
  • $n_p=2$ - $p(t,w)$ is a bigram HMM
  • $n_q=1$ - $q(t|w)$ conditions tag only on n-gram word context (approximate, but saves space)
Results: WSJ

1 million words, count cutoff of 3, 430k n-grams

- EM
- N-gram EM

Error rate

Time (seconds)

Negative log-likelihood

Time (seconds)
Results: 20m Gigaword

20 million words, count cutoff of 10, 2.8m n-grams

- EM
- N-gram EM

Error rate

Time (seconds)

Negative log-likelihood

Time (seconds)
Results: 200m Gigaword

200 million words, count cutoff of 20, 14m n-grams

EM

N-gram EM

Error rate

Time (seconds)

Negative log-likelihood

Time (seconds)
Conclusions

• **New problem**: train on an infinite corpus (distribution over sentences)

• **New algorithms**: exact and approximate likelihood maximization

• **New results**: faster (sublinear) training by compressing corpus into n-gram model