Sequence Models for Information Extraction, POS tagging, Word Segmentation, Chunking, ... 

CS224N 2007

(Some slides are mine; many slides are borrowed from Andrew McCallum and William Cohen’s IE Tutorial)
Statistical sequence models for Information Extraction

• There are several techniques for information extraction (template/wrapper learning, hand-coded rules) …

• But statistical sequence models (Hidden Markov Models, MaxEnt markov models, CRFs) are good methods for sequence-based information extraction

• Pros:
  – Well-understood underlying statistical model
  – Can do some form of optimal inference along sequence
  – Portable, broad coverage, robust, good recall

• Cons:
  – Not necessarily as good for complex or multi-slot patterns
  – Only doing the entity mention labeling task (in general)
Applying HMMs to IE
(Leek 1997, Freitag and McCallum 2000)

- **Multinomial HMMs** are sequential version of naïve Bayes/LM.
- **Document** ⇒ generated by a stochastic process
- **Observation** ⇒ word
- **State** ⇒ “reason/explanation” for a given token
  - ‘Background’ state emits tokens like ‘the’, ‘said’, …
  - ‘Money’ state emits tokens like ‘million’, ‘euro’, …
  - ‘Person’ state emits tokens like ‘Tony’, ‘Prithi’, …
- **Extraction**: via the Viterbi (max likelihood parse) algorithm
IE with Hidden Markov Models

Given a sequence of observations:

Yesterday Pedro Domingos spoke this example sentence.

and a trained HMM:

Find the most likely state sequence:  (Viterbi)

Yesterday Pedro Domingos spoke this example sentence.

Any words said to be generated by the designated “person name” state extract as a person name:

Person name: Pedro Domingos
HMM for research papers: transitions A

[Seymore et al., 99]
HMM for research papers: emissions B

[Seymore et al., 99]

Trained on 2 million words of BibTeX data from the Web
Freitag and McCallum (2000)
IE with HMMs details

• Partly fixed structure, partly hidden
  (constrained EM using remote supervision)
  – Class HMM (also used in comp. bio.)
• Parameter tying and shrinkage smoothing techniques
  – Better just to use a good unknown model?
• Structure learning of transition structure
  – Why not just plain EM?
• Results great on semi-structured data!
  – 92.9% token accuracy on paper/citations data
• Still rather modest on free form text
HMM IE results ($F_1$) on Freitag and McCallum Acquisitions data

Average $F_1$ over 10 folds

- FrMcC 99/00
- Jim
- Chris2
- S-Merged

Trained on 500 newswire documents
Other Sequence Modeling Tasks:
Chinese Word Segmentation
(also: Japanese, Thai, Ancient Greek, …)

• Basic units in written text are “characters”.

• A sentence is a sequence of “characters”, without explicit boundaries.

• Meaningful units in written texts are “words”
  • Word meaning can differ greatly from characters

• But definition of “words” is debatable
  – Different segmentation standards defined by linguists
  – It’s like whether you segment compounds (cf. German)
Sequence Model Chinese Word Segmenter

![Diagram of Chinese word segmentation with characters, index, and label]

- **Characters (C):** 已, 開, 發, 和, 尚, 在, 開
- **Index (i):** -3, -2, -1, 0, 1, 2, 3
- **Label (0/1):**
  - C_{i-3}: 0/1
  - C_{i-2}: 0/1
  - C_{i-1}: 0/1
  - C_i: 0/1
  - C_{i+1}: 0/1
  - C_{i+2}: 0/1
  - C_{i+3}: 0/1

**Indexing:**
- 0: NONSTART
- 1: START
Other sequence modeling tasks

• Base noun phrase chunking
  – Small noun phrases are a useful unit for many applications of terminology extraction, web search
  – **Mitsubishi** has just announced a new 21.3-inch flat panel monitor for the Japanese market, and even though it offers two DVI ports and a UXGA resolution of 1,600 x 1,200, we're not sure how many folks will be willing to part with close to 200,000 yen
  – Sequence model marks segment start, end
Other sequence modeling tasks

• Topic/FAQ segmentation (question, answer)
• Part-of-speech tagging
• Musical sequences
• DNA sequences
• …
HMM Tagging Models - Brants 2000

- Highly competitive with other state-of-the-art models
- Trigram HMM with smoothed transition probabilities
- Capitalization feature becomes part of the state – each tag state is split into two e.g.
  \[ \text{NN} \rightarrow <\text{NN}, \text{cap}>, <\text{NN}, \text{not cap}> \]
- Suffix features for unknown words

\[
P(w | \text{tag}) = P(\text{suffix} | \text{tag})(w | \text{suffix}) \\
\approx \hat{P}(\text{suffix})\tilde{P}(\text{tag} | \text{suffix}) / \hat{P}(\text{tag})
\]

\[
\tilde{P}(\text{tag} | \text{suffix}_n) = \lambda_1\hat{P}(\text{tag} | \text{suffix}_n) + \lambda_2\hat{P}(\text{tag} | \text{suffix}_{n-1}) + ... + \lambda_n\hat{P}(\text{tag})
\]
The European Commission said on Thursday it disagreed with German advice. Only France and Britain backed Fischler's proposal.

“What we have to be extremely careful of is how other countries are going to take Germany's lead”, Welsh National Farmers' Union (NFU) chairman John Lloyd Jones said on BBC radio.

The purpose:
- ... a lot of information is really associations between named entities.
- ... for question answering, answers are usually named entities.
- ... the same techniques apply to other slot-filling classifications.
HMM Example: “Nymble”

Task: Named Entity Extraction

Train on ~500k words of news wire text.

Results:

<table>
<thead>
<tr>
<th>Case</th>
<th>Language</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mixed</td>
<td>English</td>
<td>93%</td>
</tr>
<tr>
<td>Upper</td>
<td>English</td>
<td>91%</td>
</tr>
<tr>
<td>Mixed</td>
<td>Spanish</td>
<td>90%</td>
</tr>
</tbody>
</table>

Transitions:

\[
P(s_t | s_{t-1}, o_{t-1})
\]

Observation:

\[
P(o_t | s_t, s_{t-1})
\]

Back-off to:

\[
P(s_t | s_{t-1})
\]

\[
P(o_t | s_t)
\]

Other examples of shrinkage for HMMs in IE: [Freitag and McCallum ‘99]
What is a symbol?

Bikel *et al* mix symbols from two abstraction levels

- A word token (for known words, seen more than $k$ times)

<table>
<thead>
<tr>
<th>Word Feature</th>
<th>Example Text</th>
<th>Intuition</th>
</tr>
</thead>
<tbody>
<tr>
<td>twoDigitNum</td>
<td>90</td>
<td>Two-digit year</td>
</tr>
<tr>
<td>fourDigitNum</td>
<td>1990</td>
<td>Four digit year</td>
</tr>
<tr>
<td>containsDigitAndAlpha</td>
<td>A8956-67</td>
<td>Product code</td>
</tr>
<tr>
<td>containsDigitAndDash</td>
<td>09-96</td>
<td>Date</td>
</tr>
<tr>
<td>containsDigitAndSlash</td>
<td>11/9/89</td>
<td>Date</td>
</tr>
<tr>
<td>containsDigitAndComma</td>
<td>23,000.00</td>
<td>Monetary amount</td>
</tr>
<tr>
<td>containsDigitAndPeriod</td>
<td>1.00</td>
<td>Monetary amount, percentage</td>
</tr>
<tr>
<td>otherNum</td>
<td>456789</td>
<td>Other number</td>
</tr>
<tr>
<td>allCaps</td>
<td>BBN</td>
<td>Organization</td>
</tr>
<tr>
<td>capPeriod</td>
<td>M.</td>
<td>Person name initial</td>
</tr>
<tr>
<td>firstWord</td>
<td><em>first word of sentence</em></td>
<td>No useful capitalization information</td>
</tr>
<tr>
<td>initCap</td>
<td>Sally</td>
<td>Capitalized word</td>
</tr>
<tr>
<td>lowerCase</td>
<td>can</td>
<td>Uncapitalized word</td>
</tr>
<tr>
<td>other</td>
<td>,</td>
<td>Punctuation marks, all other words</td>
</tr>
</tbody>
</table>
What is a symbol?

Ideally we would like to use many, arbitrary, overlapping features of words. Useful, but this is hard with HMMs

- identity of word
- ends in “-ski”
- is capitalized
- is part of a noun phrase
- is in a list of city names
- is under node X in WordNet
- is in bold font
- is indented
- is in hyperlink anchor

Lots of learning systems are not confounded by multiple, non-independent features: decision trees, maxent models, neural nets, SVMs, …
What’s in a Name?

oxa

: 708

field

- company
- movie
- place
- person

- 14
- 0
- 6
- 8
- 6
What is a symbol?

identity of word
ends in “-ski”
is capitalized
is part of a noun phrase
is in a list of city names
is under node X in WordNet
is in bold font
is indented
is in hyperlink anchor
...

Idea: replace generative model in HMM with a maxent model, where state depends on observations

$$\Pr(s_t \mid x_t) = \ldots$$
What is a symbol?

Identity of word
ends in “-ski”
is capitalized
is part of a noun phrase
is in a list of city names
is under node X in WordNet
is in bold font
is indented
is in hyperlink anchor

Idea: replace generative model in HMM with a maxent model, where state depends on observations and previous state

$$\Pr(s_t \mid x_t, s_{t-1},) = \ldots$$
What is a symbol?

identity of word
ends in “-ski”
is capitalized
is part of a noun phrase
is in a list of city names
is under node X in WordNet
is in bold font
is indented
is in hyperlink anchor

Idea: replace generative model in HMM with a maxent model, where state depends on observations and previous state history

\[
Pr(s_t \mid x_t, s_{t-1}, s_{t-2}, \ldots) = \ldots
\]
Inference in Systems

Sequence Level

- Sequence Data
- Sequence Model
- Inference

Local Level

- Local Data
- Feature Extraction
- Label
- Features
- Classifier Type
  - Optimization
  - Smoothing

Maximum Entropy Models
- Conjugate Gradient
- Quadratic Penalties
- NLP Issues
Beam Inference

• Beam inference:
  – At each position keep the top $k$ complete sequences.
  – Extend each sequence in each local way.
  – The extensions compete for the $k$ slots at the next position.

• Advantages:
  – Fast; and beam sizes of 3–5 are as good or almost as good as exact inference in many cases.
  – Easy to implement (no dynamic programming required).

• Disadvantage:
  – Inexact: the globally best sequence can fall off the beam.
Viterbi Inference

- Viterbi inference:
  - Dynamic programming or memoization.
  - Requires small window of state influence (e.g., past two states are relevant).
- Advantage:
  - Exact: the global best sequence is returned.
- Disadvantage:
  - Harder to implement long-distance state-state interactions (but beam inference tends not to allow long-distance resurrection of sequences anyway).
POS tagging: Ratnaparkhi’s MXPOST

- Sequential learning problem: predict POS tags of words.
- Uses MaxEnt model described above.
- Rich feature set.
- To smooth, discard features occurring < 10 times.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>$w_i$ is not rare</td>
<td>$w_i = X$ &amp; $t_i = T$</td>
</tr>
<tr>
<td>$w_i$ is rare</td>
<td>$X$ is prefix of $w_i$, $</td>
</tr>
<tr>
<td></td>
<td>$X$ is suffix of $w_i$, $</td>
</tr>
<tr>
<td></td>
<td>$w_i$ contains number &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_i$ contains uppercase character &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_i$ contains hyphen &amp; $t_i = T$</td>
</tr>
<tr>
<td>$\forall w_i$</td>
<td>$t_{i-1} = X$ &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$t_{i-2}t_{i-1} = XY$ &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_{i-1} = X$ &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_{i-2} = X$ &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_{i+1} = X$ &amp; $t_i = T$</td>
</tr>
<tr>
<td></td>
<td>$w_{i+2} = X$ &amp; $t_i = T$</td>
</tr>
</tbody>
</table>

Table 1: Features on the current history $h_i$
Ratnaparkhi (1996): local distributions are estimated using maximum entropy models
  - Previous two tags, current word, previous two words, next two words, suffix, prefix, hyphenation, and capitalization features for unknown words

Toutanova et al. (2003)
  - Richer features, bidirectional inference, better smoothing, better unknown word handling

<table>
<thead>
<tr>
<th>Model</th>
<th>Overall Accuracy</th>
<th>Unknown Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMM (Brants 2000)</td>
<td>96.7</td>
<td>85.5</td>
</tr>
<tr>
<td>CMM (Ratn. 1996)</td>
<td>96.63</td>
<td>85.56</td>
</tr>
<tr>
<td>CMM (T. et al 2003)</td>
<td>97.24</td>
<td>89.04</td>
</tr>
</tbody>
</table>
Smoothing: POS Tagging

- From (Toutanova et al., 2003):
  
<table>
<thead>
<tr>
<th>Overall Accuracy</th>
<th>Unknown Word Acc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Smoothing</td>
<td>96.54</td>
</tr>
<tr>
<td>With Smoothing</td>
<td>97.10</td>
</tr>
</tbody>
</table>

- Smoothing helps:
  - Softens distributions.
  - Pushes weight onto more explanatory features.
  - Allows many features to be dumped (fairly) safely into the mix.
  - Speeds up convergence (if both are allowed to converge)!
Summary of POS Tagging

For tagging, the change from generative to discriminative model does not by itself result in great improvement.

One profits from discriminative models for specifying dependence on overlapping features of the observation such as spelling, suffix analysis, etc.

A CMM allows integration of rich features of the observations, but suffers strongly from assuming independence from following observations; this effect can be relieved by adding dependence on following words.

This additional power (of the CMM, CRF, Perceptron models) has been shown to result in improvements in accuracy.

The higher accuracy of discriminative models comes at the price of much slower training.
CoNLL (2003) Named Entity Recognition task

Task: Predict semantic label of each word in text

Foreign           NNP  I-NP  ORG
Ministry          NNP  I-NP  ORG
spokesman         NN   I-NP  O
Shen             NNP  I-NP  PER
Guofang          NNP  I-NP  PER
told             VBD  I-VP  O
Reuters           NNP  I-NP  ORG

Standard evaluation is per entity, not per token
NER Results: Discriminative Model

- Increases from better features, a better classification model.

CoNLL 2003 Shared Task: English NER; entity precision/recall F1
Sequence models? CoNLL 2003 NER shared task
Results on English Devset

<table>
<thead>
<tr>
<th>Model</th>
<th>Overall</th>
<th>Loc</th>
<th>Misc</th>
<th>Org</th>
<th>Person</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMM 1st</td>
<td>94</td>
<td>94</td>
<td>88</td>
<td>88</td>
<td>96</td>
</tr>
<tr>
<td>CRF</td>
<td>93</td>
<td>93</td>
<td>87</td>
<td>87</td>
<td>94</td>
</tr>
<tr>
<td>MMMN</td>
<td>95</td>
<td>95</td>
<td>89</td>
<td>89</td>
<td>96</td>
</tr>
</tbody>
</table>
CoNLL NER Results: CMM Order

![Bar chart showing performance metrics for different orders.](image-url)
Sequence Tagging Without Sequence Information: POS tagging

Using 3 words only works significantly better than using the previous two or three tags instead! (Toutanova et al. 2003)
CoNLL NER: A real difference

- A difference of about 0.7% gives significance among good CoNLL results
- Here we get one!
- It was done with some Perl regular expressions