Introducing the tasks:
Getting simple structured information out of text
Information Extraction

• Information extraction (IE) systems
  • Find and understand limited relevant parts of texts
  • Gather information from many pieces of text
  • Produce a structured representation of relevant information:
    • *relations* (in the database sense), a.k.a.,
    • a *knowledge base*
• Goals:
  1. Organize information so that it is useful to people
  2. Put information in a semantically precise form that allows further inferences to be made by computer algorithms
Information Extraction (IE)

- IE systems extract clear, factual information
  - Roughly: *Who did what to whom when?*
- E.g.,
  - Gathering earnings, profits, board members, headquarters, etc. from company reports
    - The headquarters of BHP Billiton Limited, and the global headquarters of the combined BHP Billiton Group, are located in Melbourne, Australia.
  - headquarters(“BHP Billiton Limited”, “Melbourne, Australia”)
- Learn drug-gene product interactions from medical research literature
Low-level information extraction

• Is now available – and I think popular – in applications like Apple or Google mail, and web indexing

• Often seems to be based on regular expressions and name lists
Low-level information extraction
Named Entity Recognition (NER)

- A very important sub-task: find and classify names in text, for example:
  - The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.
A very important sub-task: find and classify names in text, for example:

The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.
Named Entity Recognition (NER)

• A very important sub-task: find and classify names in text, for example:

• The decision by the independent MP Andrew Wilkie to withdraw his support for the minority Labor government sounded dramatic but it should not further threaten its stability. When, after the 2010 election, Wilkie, Rob Oakeshott, Tony Windsor and the Greens agreed to support Labor, they gave just two guarantees: confidence and supply.
Named Entity Recognition (NER)

- The uses:
  - Named entities can be indexed, linked off, etc.
  - Sentiment can be attributed to companies or products
  - A lot of IE relations are associations between named entities
  - For question answering, answers are often named entities.

- Concretely:
  - Many web pages tag various entities, with links to bio or topic pages, etc.
    - Reuters’ OpenCalais, Evri, AlchemyAPI, Yahoo’s Term Extraction, ...
  - Apple/Google/Microsoft/… smart recognizers for document content
Information Extraction and Named Entity Recognition

Introducing the tasks:
Getting simple structured information out of text
Evaluation of Named Entity Recognition

The extension of Precision, Recall, and the F measure to sequences
The Named Entity Recognition Task

Task: Predict entities in a text

Foreign	ORG
Ministry	ORG
spokesman	O
Shen	PER
Guofang	PER
told	O
Reuters	ORG

Standard evaluation is per entity, not per token
Precision/Recall/F1 for IE/NER

- Recall and precision are straightforward for tasks like IR and text categorization, where there is only one grain size (documents).
- The measure behaves a bit funny for IE/NER when there are boundary errors (which are common):
  - First Bank of Chicago announced earnings ...
- This counts as both a fp and a fn
- Selecting nothing would have been better
- Some other metrics (e.g., MUC scorer) give partial credit (according to complex rules)
Evaluation of Named Entity Recognition

The extension of Precision, Recall, and the F measure to sequences
Sequence Models for Named Entity Recognition
The ML sequence model approach to NER

Training
1. Collect a set of representative training documents
2. Label each token for its entity class or other (O)
3. Design feature extractors appropriate to the text and classes
4. Train a sequence classifier to predict the labels from the data

Testing
1. Receive a set of testing documents
2. Run sequence model inference to label each token
3. Appropriately output the recognized entities
# Encoding classes for sequence labeling

<table>
<thead>
<tr>
<th></th>
<th>IO encoding</th>
<th>IOB encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fred</td>
<td>PER</td>
<td>B-PER</td>
</tr>
<tr>
<td>showed</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Sue</td>
<td>PER</td>
<td>B-PER</td>
</tr>
<tr>
<td>Mengqiu</td>
<td>PER</td>
<td>B-PER</td>
</tr>
<tr>
<td>Huang</td>
<td>PER</td>
<td>I-PER</td>
</tr>
<tr>
<td>‘s</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>new</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>painting</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>
Features for sequence labeling

- **Words**
  - Current word (essentially like a learned dictionary)
  - Previous/next word (context)
- **Other kinds of inferred linguistic classification**
  - Part-of-speech tags
- **Label context**
  - Previous (and perhaps next) label
Features: Word substrings

- oxa
- : 708
- field 68

- drug: 18
- company: 6
- movie: 0
- place: 0
- person: 8

- Cotrimoxazole
- Wethersfield
- Alien Fury: Countdown to Invasion
Features: Word shapes

- Word Shapes
  - Map words to simplified representation that encodes attributes such as length, capitalization, numerals, Greek letters, internal punctuation, etc.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Varicella-zoster</td>
<td>Xx-xxx</td>
</tr>
<tr>
<td>mRNA</td>
<td>xXXX</td>
</tr>
<tr>
<td>CPA1</td>
<td>XXXd</td>
</tr>
</tbody>
</table>
Sequence Models for Named Entity Recognition
Maximum entropy sequence models

Maximum entropy Markov models (MEMMs) or Conditional Markov models
Sequence problems

• Many problems in NLP have data which is a sequence of characters, words, phrases, lines, or sentences ...

• We can think of our task as one of labeling each item

<table>
<thead>
<tr>
<th>VBG</th>
<th>NN</th>
<th>IN</th>
<th>DT</th>
<th>NN</th>
<th>IN</th>
<th>NN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chasing</td>
<td>opportunity</td>
<td>in</td>
<td>an</td>
<td>age</td>
<td>of</td>
<td>upheaval</td>
</tr>
</tbody>
</table>

POS tagging

<table>
<thead>
<tr>
<th>PERS</th>
<th>O</th>
<th>O</th>
<th>O</th>
<th>ORG</th>
<th>ORG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Murdoch</td>
<td>discusses</td>
<td>future</td>
<td>of</td>
<td>News</td>
<td>Corp.</td>
</tr>
</tbody>
</table>

Named entity recognition

Text segmentation
MEMM inference in systems

- For a Conditional Markov Model (CMM) a.k.a. a Maximum Entropy Markov Model (MEMM), the classifier makes a single decision at a time, conditioned on evidence from observations and previous decisions.
- A larger space of sequences is usually explored via search.

Local Context

<table>
<thead>
<tr>
<th></th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>+1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NNP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VBD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>???</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>???</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>The</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dow</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fell</td>
<td></td>
<td></td>
<td>22.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Features

- $W_0$: 22.6
- $W_{+1}$: %
- $W_{-1}$: fell
- $T_{-1}$: VBD
- $T_{-1} - T_{-2}$: NNP-VBD
- hasDigit?: true
- ...
- ...

(Ratnaparkhi 1996; Toutanova et al. 2003, etc.)
Example: POS Tagging

- Scoring individual labeling decisions is no more complex than standard classification decisions
  - We have some assumed labels to use for prior positions
  - We use features of those and the observed data (which can include current, previous, and next words) to predict the current label

<table>
<thead>
<tr>
<th>Local Context</th>
<th>Decision Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>DT</td>
</tr>
<tr>
<td>-2</td>
<td>NNP</td>
</tr>
<tr>
<td>-1</td>
<td>VBD</td>
</tr>
<tr>
<td>0</td>
<td>???</td>
</tr>
<tr>
<td>+1</td>
<td>The</td>
</tr>
<tr>
<td>-</td>
<td>Dow</td>
</tr>
<tr>
<td>-</td>
<td>fell</td>
</tr>
</tbody>
</table>

(Ratnaparkhi 1996; Toutanova et al. 2003, etc.)

<table>
<thead>
<tr>
<th>Features</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_0$</td>
<td>22.6</td>
</tr>
<tr>
<td>$W_{+1}$</td>
<td>%</td>
</tr>
<tr>
<td>$W_{-1}$</td>
<td>fell</td>
</tr>
<tr>
<td>$T_{-1}$</td>
<td>VBD</td>
</tr>
<tr>
<td>$T_{-1-T_{-2}}$</td>
<td>NNP-VBD</td>
</tr>
<tr>
<td>hasDigit?</td>
<td>true</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>
Example: POS Tagging

- POS tagging Features can include:
  - Current, previous, next words in isolation or together.
  - Previous one, two, three tags.
  - Word-internal features: word types, suffixes, dashes, etc.

Ratnaparkhi 1996; Toutanova et al. 2003, etc.

<table>
<thead>
<tr>
<th>Local Context</th>
<th>Decision Point</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
<tr>
<td>+1</td>
<td></td>
</tr>
<tr>
<td>DT</td>
<td></td>
</tr>
<tr>
<td>NNP</td>
<td></td>
</tr>
<tr>
<td>VBD</td>
<td></td>
</tr>
<tr>
<td>The</td>
<td></td>
</tr>
<tr>
<td>Dow</td>
<td></td>
</tr>
<tr>
<td>fell</td>
<td>22.6</td>
</tr>
</tbody>
</table>

Features

<table>
<thead>
<tr>
<th>Features</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$W_0$</td>
<td>22.6</td>
</tr>
<tr>
<td>$W_{+1}$</td>
<td>%</td>
</tr>
<tr>
<td>$W_{-1}$</td>
<td>fell</td>
</tr>
<tr>
<td>$T_{-1}$</td>
<td>VBD</td>
</tr>
<tr>
<td>$T_{-1}-T_{-2}$</td>
<td>NNP-VBD</td>
</tr>
<tr>
<td>hasDigit?</td>
<td>true</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Inference in Systems

Sequence Level

Sequence Model

Inference

Local Level

Classifier Type

Optimization

Smoothing

Maximum Entropy Models

Conjugate Gradient

Quadratic Penalties

Feature Extraction

Label

Features

Local Data

Local Data

Feature Extraction

Label

Features

Maximum Entropy Models

Conjugate Gradient

Quadratic Penalties

Label

Features

Sequence Data
Greedy Inference

- Greedy inference:
  - We just start at the left, and use our classifier at each position to assign a label
  - The classifier can depend on previous labeling decisions as well as observed data
- Advantages:
  - Fast, no extra memory requirements
  - Very easy to implement
  - With rich features including observations to the right, it may perform quite well
- Disadvantage:
  - Greedy. We make commit errors we cannot recover from
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; beam sizes of 3–5 are 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).
CRFs [Lafferty, Pereira, and McCallum 2001]

- Another sequence model: Conditional Random Fields (CRFs)
- A whole-sequence conditional model rather than a chaining of local models.

\[
P(c \mid d, \lambda) = \frac{\exp \sum \lambda_i f_i(c, d)}{\sum \exp \sum \lambda_i f_i(c', d)}
\]

- The space of \(c\)'s is now the space of sequences
  - But if the features \(f_i\) remain local, the conditional sequence likelihood can be calculated exactly using dynamic programming
- Training is slower, but CRFs avoid causal-competition biases
- These (or a variant using a max margin criterion) are seen as the state-of-the-art these days ... but in practice usually work much the same as MEMMs.
Maximum entropy sequence models

Maximum entropy Markov models (MEMMs) or Conditional Markov models