Information Extraction from Text

Jing Jiang

Chapter 2 from *Mining Text Data* (2012)

Presented by Andrew Landgraf, September 13, 2013

What is Information Extraction?

- Goal is to discover structured information from unstructured text
- Early research was on **template filling** (e.g. Police report)
- "AOL merged with Time Warner in 2000"
- IBM Watson, Google, Wolfram Alpha use information extraction
- Information extraction can be broken down into **Named Entity Recognition** and **Relation Extraction**

Named Entity Recognition

- Task is to **identify** real-world named entities and **classify** them into entity types (e.g. person, organization, location)
- Challenge in words that can have multiple entity types (e.g. JFK)
- Most fundamental task in information extraction
- Building block for relation extraction, Q & A, search engine queries
- Rule-based systems came first but are expensive and domain dependent

Statistical Learning for Named Entity Recognition

- **Sequence labeling** is used in many natural language processing tasks
- Have a sequence of observations $x_i, i = 1 \ldots n$ and corresponding labels y_i
- x_i is a feature vector for the i^{th} word
- y_i is dependent on not just x_i but also $x_{\mathcal{N}(i)}$ and $y_{\mathcal{N}(i)}$
- Each word in a sentence is treated as an observation x
- Class labels, y_i, identify boundaries and types of named entities

Steve	Jobs	was	a	co-founder	of	Apple	Inc.
B-PER	I-PER	O	O	O	O	B-ORG	I-ORG

Figure 2.2. An example sentence with NER labels in the BIO notation. PER stands for person and ORG stands for organization.

Hidden Markov Models (HMMs)

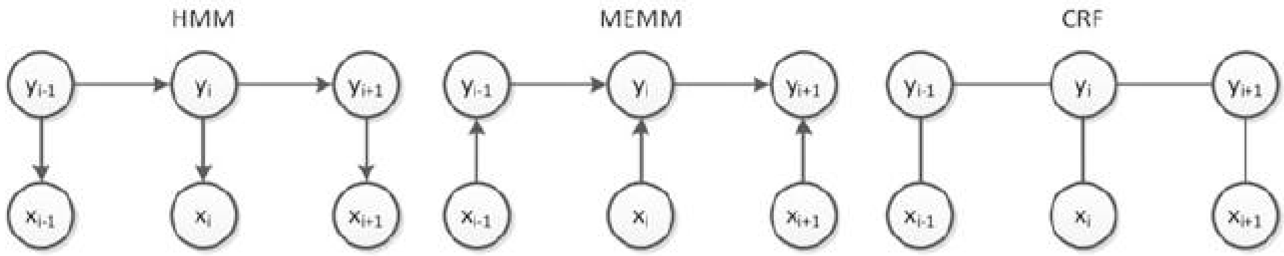

Figure 2.3. Graphical representations of linear-chain HMM, MEMM and CRF.

- Treat $\mathbf{y}$ as hidden states, maximize $p(\mathbf{x}, \mathbf{y})$
- A first order hidden Markov model is defined as follows
  - $p(y_i | y_{[-i]}, \mathbf{x}) = p(y_i | y_{i-1})$
  - $p(x_i | x_{[-i]}, \mathbf{y}) = p(x_i | y_i)$
  - $p(\mathbf{x}, \mathbf{y}) = \prod p(y_i | y_{i-1}) p(x_i | y_i)$

Maximum Entropy Markov Models

- Model the conditional distribution of y given x
- Discriminative models have been more successful than generative models
- $p(\mathbf{y}|\mathbf{x}) = \prod p(y_i|x_{\mathcal{N}_l(i)}, y_{i-1})$
- Functional form implies an exponential model

$$p(y_i|x_{\mathcal{N}_l(i)}, y_{i-1}) = \frac{\exp\left(\sum_j \lambda_j f_j(y_i, x_{\mathcal{N}_l(i)}, y_{i-1})\right)}{\sum_{y'} \exp\left(\sum_j \lambda_j f_j(y', x_{\mathcal{N}_l(i)}, y_{i-1})\right)}$$

- L-BFGS is a common method to training the model

Conditional Random Fields

- CRFs have undirected edges and the current label can depend on previous and future labels

- Linear chain CRF:

$$p(\mathbf{y}|\mathbf{x}) \propto \exp\left(\sum_i \sum_j \lambda_j f_j(y_i, y_{i-1}, \mathbf{x}, i)\right)$$

- More difficult to train because the normalizing constant is a sum over all possible label sequences

Relation Extraction

- **Detect** and **characterize** the semantic relations between two entities
- Limit the problem to relations between two entities in the same sentence
- Possible relation types include: physical, personal/social, employment/ affiliation, etc.
- Different methods for this task include feature-based, kernel-based, and weakly supervised

Feature-based Relation Classification

- If a pair of known **entities** co-occur in a sentence, it is a candidate for a relationship
- An additional relation type of ***nil*** is added
- Required to have a training data set in which the relation types are hand-labeled
- After adding independent variables (next slide) to the data set, train a multi-category classifier
  - Multinomial logistic regression, multi-category SVM, discriminant analysis, etc.
  - Can also separate the detection of a relationship and the characterization into two separate classifications

Feature-based Relation Classification

- Independent variables (features) are created by various domain specific **feature engineering** methods
  - Features related to the entity (e.g. If the entity is a person, family is a likely relation type)
  - Contextual features (e.g. If the word founded appears, especially if a person entity is the subject and an organization entity is the object)
  - Outside data (e.g. Do the two entities co-occur in the same Wikipedia article?)

Kernel Methods for Relation Extraction

- In typical SVMs, $h : \mathbb{R}^p \to \mathbb{R}^d, d \gg p$

$$
\begin{aligned}
f(x) &= \beta_0 + h(x)^T \beta \\
&= \beta_0 + \sum_{i=1}^{n} \alpha_i y_i K(x, x_i)
\end{aligned}
$$

  where $K(\cdot, \cdot)$ is the positive definite **kernel** corresponding to h

- The kernel function between two sentences is large if the two sentences are similar is some way
  - Shortest dependency paths (e.g. "protestors seized stations" and "troops raided churches" both have the form "person verb facility")
  - Tree-based kernels

Weakly Supervised Relation Extraction

- It is expensive/time consuming to label all the relations in a corpus
- These methods use require less training data
- **Bootstrapping** (not the usual kind)
  - With a small amount of data learn relations
  - Predict new relations
  - Use the predicted relations you are sure about as new training data
- **Distant Supervision**
  - Supplement data with labeled examples from other corpora (e.g. Wikipedia, Freebase)

Evaluation

- Manually annotated test documents from the same domain have to be used
- **Precision**: percent correct among those predicted to be positive
- **Recall**: percent of all positives that were identified as such
- **F-1**: geometric mean of precision and recall