**Lecture 12 – Discriminative Parsing**

1. Grammars
    a. Dependency Grammars
2. Discriminative Parsers
    a. Graph based parsing
        i. Parameterization
        ii. Inference
        iii.
        iv. Learning
3. Two extensions
    a. Neural representation (Kipperwassser and Goldberg)
    b. Randomized greedy parsing

**Regular Languages**

- $\Sigma$
  1. $\emptyset$
  2. $a \in \Sigma$
  3. $A$ and $B$ are regular $\rightarrow A \cup B$, $AB$, $A^*$ is regular
- Not all languages are regular

**Context Free Languages**

- $S \rightarrow \alpha S \beta$ (recursive construction/center embedding), cannot be represented by regular languages
    - i.e. The dog that snores sleeps.
- It was though that context free was sufficient…
    - But it was found that it was not; in fact we need context sensitive

**Dependency Grammars**

- Only one type of relationship between words
    - Parent child or governs or depends…
- Easier to train
- Problems
    - No constituents
    - Can be hard to annotate

**Tree Adjoining Grammar**

- More powerful than PCFG (can be context sensitive)

**Generative vs Discriminative**

- Discriminative $p(y|x)$
    - Slower to get good performance
    - Easily add more features

- Generative $p(x, y)$
    - This distribution can be sampled
    - Can very quickly get good performance
    - Difficult to add more features
    - NLP is mostly generative models

**Dependency Parsing**

- Words represented as $x$
- Arcs represented as $y$ (pairs $(i, j)$)
- $y(x)$ = all the direct rooted trees over $x$ (universe of possibilities)
- Scoring $score(x, y, \theta)$
    - Arc-factored scoring
        - Sum of arc scores $s = \sum_{i,j} score(a(i, j), \theta) = \sum f(x, i, j)\theta$
        - Can have many features $f(x, i, j)$…is this a problem?
            - No, can rely on other features…unlike language model where its possible no features exist.
- Inference $\underset{\theta}{\operatorname{argmax}}\, score(x, y, \theta)$
    - Find the maximum weight spanning tree for a directed graph
- Estimation of $\theta$
    - For each $x$
        - Compute $\hat{y} = \max_{y} f(x, y)\theta$
        - If $y \neq \hat{y}$
            - $\theta = \theta + f(x, y) - f(x, \hat{y})$