Global Linear Models

Michael Collins, Columbia University

Overview

- A brief review of history-based methods
- A new framework: Global linear models
- Parsing problems in this framework:
  Reranking problems
- Parameter estimation method 1:
  A variant of the perceptron algorithm

Techniques

- So far:
  - Smoothed estimation
  - Probabilistic context-free grammars
  - Log-linear models
  - Hidden markov models
  - The EM Algorithm
  - History-based models
- Today:
  - Global linear models

Supervised Learning in Natural Language

- General task: induce a function F from members of a set $\mathcal{X}$ to members of a set $\mathcal{Y}$. e.g.,

Problem	$x \in \mathcal{X}$	$y \in \mathcal{Y}$
Parsing	sentence	parse tree
Machine translation	French sentence	English sentence
POS tagging	sentence	sequence of tags

- Supervised learning:
  we have a *training set* (x_i, y_i) for $i = 1 \dots n$

The Models so far

- Most of the models we've seen so far are **history-based models**:
  - We break structures down into a derivation, or sequence of decisions
  - Each decision has an associated conditional probability
  - Probability of a structure is a product of decision probabilities
  - Parameter values are estimated using variants of maximum-likelihood estimation
  - Function $F : \mathcal{X} \rightarrow \mathcal{Y}$ is defined as

$$F(x) = \mathrm{argmax}_y p(x, y; \Theta) \quad \text{or} \quad F(x) = \mathrm{argmax}_y p(y|x; \Theta)$$

Example 1: PCFGs

- We break structures down into a derivation, or sequence of decisions
  We have a top-down derivation, where each decision is to expand some non-terminal α with a rule $\alpha \rightarrow \beta$
- Each decision has an associated conditional probability
  $\alpha \rightarrow \beta$ has probability $q(\alpha \rightarrow \beta)$
- Probability of a structure is a product of decision probabilities

$$p(T, S) = \prod_{i=1}^{n} q(\alpha_i \rightarrow \beta_i)$$

  where $\alpha_i \rightarrow \beta_i$ for $i = 1 \ldots n$ are the n rules in the tree
- Parameter values are estimated using variants of maximum-likelihood estimation

$$q(\alpha \rightarrow \beta) = \frac{Count(\alpha \rightarrow \beta)}{Count(\alpha)}$$

- Function $F : \mathcal{X} \rightarrow \mathcal{Y}$ is defined as

$$F(x) = \text{argmax}_y p(y, x; \Theta)$$

Example 2: Log-linear Taggers

- We break structures down into a derivation, or sequence of decisions
  For a sentence of length n we have n tagging decisions, in left-to-right order
- Each decision has an associated conditional probability

  $$p(t_i \mid t_{i-1}, t_{i-2}, w_1 \dots w_n)$$

  where t_i is the i'th tagging decision, w_i is the i'th word
- Probability of a structure is a product of decision probabilities

  $$p(t_1 \dots t_n \mid w_1 \dots w_n) = \prod_{i=1}^{n} p(t_i \mid t_{i-1}, t_{i-2}, w_1 \dots w_n)$$

- Parameter values are estimated using variants of maximum-likelihood estimation
  $p(t_i \mid t_{i-1}, t_{i-2}, w_1 \dots w_n)$ is estimated using a log-linear model
- Function $F : \mathcal{X} \rightarrow \mathcal{Y}$ is defined as

  $$F(x) = \mathrm{argmax}_y p(y \mid x; \Theta)$$

A New Set of Techniques: Global Linear Models

Overview of today's lecture:

- Global linear models as a framework
- Parsing problems in this framework:
  - Reranking problems
- A variant of the perceptron algorithm

Global Linear Models as a Framework

- We'll move away from history-based models
  No idea of a "derivation", or attaching probabilities to "decisions"
- Instead, we'll have feature vectors over entire structures
  "Global features"
- First piece of motivation:
  Freedom in defining features

A Need for Flexible Features

Example 1 Parallelism in coordination [Johnson et. al 1999]

Constituents with similar structure tend to be coordinated
⇒ how do we allow the parser to learn this preference?

Bars in New York and pubs in London
vs. Bars in New York and pubs

A Need for Flexible Features (continued)

Example 2 Semantic features

We might have an ontology giving properties of various nouns/verbs
$\Rightarrow$ how do we allow the parser to use this information?

pour the **cappucino**
vs. pour the **book**

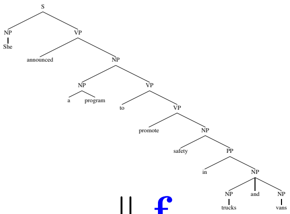
Ontology states that **cappucino** has the `+liquid` feature, **book** does not.

Three Components of Global Linear Models

- $\mathbf{f}$ is a function that maps a structure (x, y) to a **feature vector** $\mathbf{f}(x, y) \in \mathbb{R}^d$
- $\mathbf{GEN}$ is a function that maps an input x to a set of **candidates** $\mathbf{GEN}(x)$
- $\mathbf{v}$ is a parameter vector (also a member of $\mathbb{R}^d$)
- Training data is used to set the value of $\mathbf{v}$

Component 1: $\mathbf{f}$

- $\mathbf{f}$ maps a candidate to a **feature vector** $\in \mathbb{R}^d$
- $\mathbf{f}$ defines the **representation** of a candidate

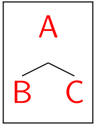


$$\Downarrow \mathbf{f}$$

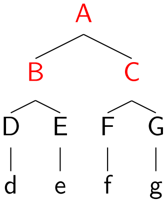
$$\langle 1, 0, 2, 0, 0, 15, 5 \rangle$$

Features

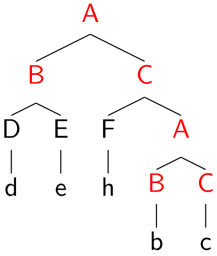
- A "feature" is a function on a structure, e.g.,

$h(x, y) =$ Number of times  is seen in (x, y)

(x_1, y_1)



(x_2, y_2)



$h(x_1, y_1) = 1$

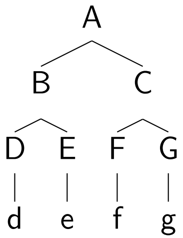
$h(x_2, y_2) = 2$

Feature Vectors

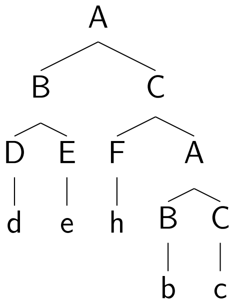
- A set of functions $h_1 \dots h_d$ define a **feature vector**

$$\mathbf{f}(x) = \langle h_1(x), h_2(x) \dots h_d(x) \rangle$$

T_1



T_2



$$\mathbf{f}(T_1) = \langle 1, 0, 0, 3 \rangle$$

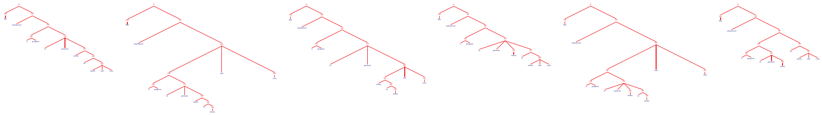
$$\mathbf{f}(T_2) = \langle 2, 0, 1, 1 \rangle$$

Component 2: **GEN**

- **GEN** enumerates a set of **candidates** for a sentence

She announced a program to promote safety in trucks and vans

$\Downarrow$ **GEN**



Component 2: $\mathbf{GEN}$

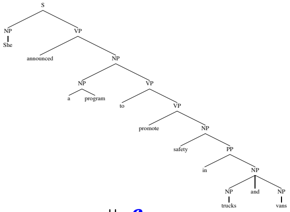
- $\mathbf{GEN}$ enumerates a set of **candidates** for an input x

- Some examples of how $\mathbf{GEN}(x)$ can be defined:
  - Parsing: $\mathbf{GEN}(x)$ is the set of parses for x under a grammar
  - Any task: $\mathbf{GEN}(x)$ is the top N most probable parses under a history-based model
  - Tagging: $\mathbf{GEN}(x)$ is the set of all possible tag sequences with the same length as x
  - Translation: $\mathbf{GEN}(x)$ is the set of all possible English translations for the French sentence x

Component 3: $\mathbf{v}$

- $\mathbf{v}$ is a **parameter vector** $\in \mathbb{R}^d$
- $\mathbf{f}$ and $\mathbf{v}$ together map a candidate to a real-valued score

---



$$\Downarrow \mathbf{f}$$

$$\langle 1, 0, 2, 0, 0, 15, 5 \rangle$$

$$\Downarrow \mathbf{f} \cdot \mathbf{v}$$

$$\langle 1, 0, 2, 0, 0, 15, 5 \rangle \cdot \langle 1.9, -0.3, 0.2, 1.3, 0, 1.0, -2.3 \rangle = 5.8$$

Putting it all Together

- $\mathcal{X}$ is set of sentences, $\mathcal{Y}$ is set of possible outputs (e.g. trees)
- Need to learn a function $F : \mathcal{X} \to \mathcal{Y}$
- $\mathbf{GEN}$, $\mathbf{f}$, $\mathbf{v}$ define

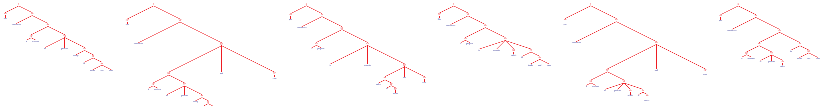
$$F(x) = \arg\max_{y \in \mathbf{GEN}(x)} \mathbf{f}(x, y) \cdot \mathbf{v}$$

Choose the highest scoring candidate as the most plausible structure

- Given examples (x_i, y_i), how to set $\mathbf{v}$?

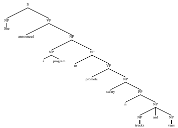
She announced a program to promote safety in trucks and vans

$\Downarrow$ **GEN**



$\Downarrow \mathbf{f}$	$\Downarrow \mathbf{f}$	$\Downarrow \mathbf{f}$	$\Downarrow \mathbf{f}$	$\Downarrow \mathbf{f}$	$\Downarrow \mathbf{f}$
$\langle 1, 1, 3, 5 \rangle$	$\langle 2, 0, 0, 5 \rangle$	$\langle 1, 0, 1, 5 \rangle$	$\langle 0, 0, 3, 0 \rangle$	$\langle 0, 1, 0, 5 \rangle$	$\langle 0, 0, 1, 5 \rangle$
$\Downarrow \mathbf{f} \cdot \mathbf{v}$	$\Downarrow \mathbf{f} \cdot \mathbf{v}$	$\Downarrow \mathbf{f} \cdot \mathbf{v}$	$\Downarrow \mathbf{f} \cdot \mathbf{v}$	$\Downarrow \mathbf{f} \cdot \mathbf{v}$	$\Downarrow \mathbf{f} \cdot \mathbf{v}$
13.6	12.2	12.1	3.3	9.4	11.1

$\Downarrow \arg\max$



Overview

- A brief review of history-based methods
- A new framework: Global linear models
- Parsing problems in this framework:
  Reranking problems
- Parameter estimation method 1:
  A variant of the perceptron algorithm

Reranking Approaches to Parsing

- Use a **baseline** parser to produce top N parses for each sentence in training and test data
  $\mathbf{GEN}(x)$ is the top N parses for x under the baseline model

- One method: use a lexicalized PCFG to generate a number of parses
  (in our experiments, around 25 parses on average for 40,000 training sentences, giving $\approx$ 1 million training parses)

- **Supervision:** for each x_i take y_i to be the parse that is "closest" to the treebank parse in $\mathbf{GEN}(x_i)$

The Representation $\mathbf{f}$

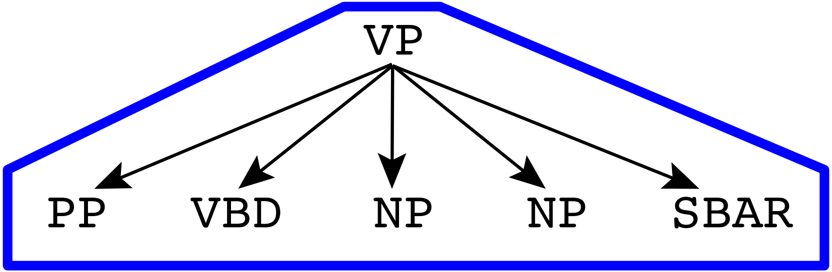
- Each component of $\mathbf{f}$ could be essentially *any* feature over parse trees
- For example:

$$f_1(x, y) = \text{log probability of } (x, y) \text{ under the baseline model}$$

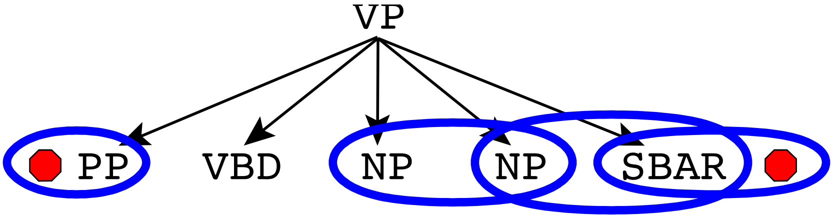
$$f_2(x, y) = \begin{cases} 1 & \text{if } (x, y) \text{ includes the rule } \texttt{VP} \rightarrow \texttt{PP VBD NP} \\ 0 & \text{otherwise} \end{cases}$$

From [Collins and Koo, 2005]:
The following types of features were included in the model. We will use the rule `VP -> PP VBD NP NP SBAR` with head `VBD` as an example. Note that the output of our baseline parser produces syntactic trees with headword annotations.

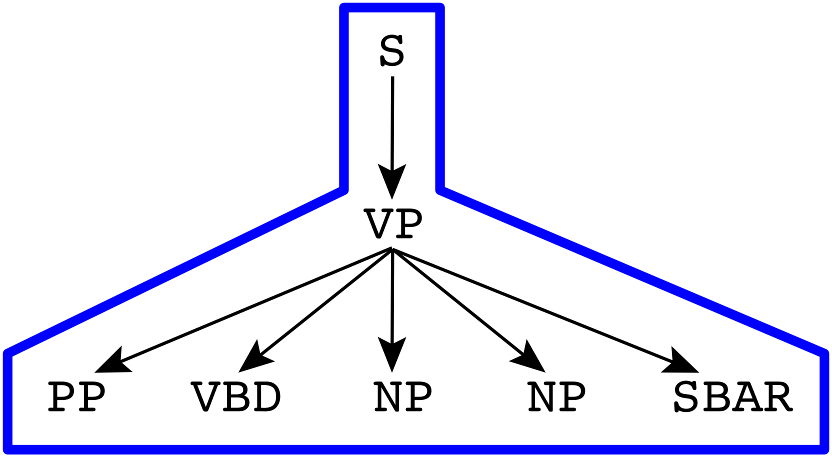
Rules These include all context-free rules in the tree, for example
`VP -> PP VBD NP NP SBAR`.



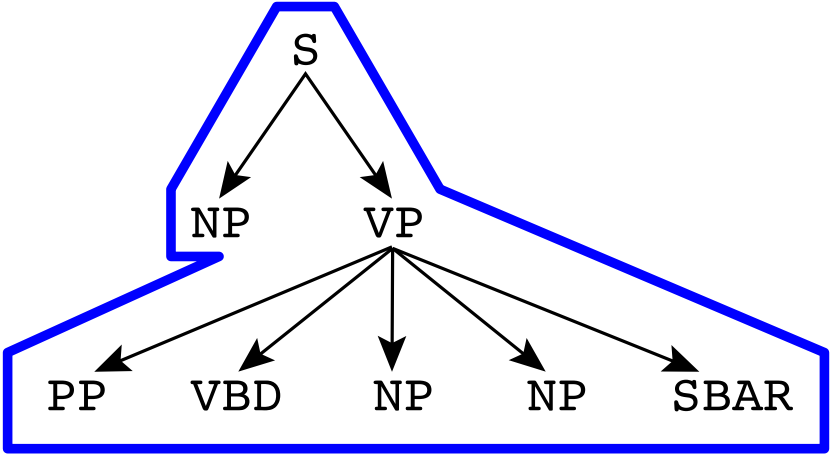
Bigrams These are adjacent pairs of non-terminals to the left and right of the head. As shown, the example rule would contribute the bigrams `(Right,VP,NP,NP)`, `(Right,VP,NP,SBAR)`, `(Right,VP,SBAR,STOP)`, and `(Left,VP,PP,STOP)` to the left of the head.



Grandparent Rules Same as **Rules**, but also including the non-terminal above the rule.



Two-level Rules Same as **Rules**, but also including the entire rule above the rule.



Overview

- A brief review of history-based methods
- A new framework: Global linear models
- Parsing problems in this framework:
  Reranking problems
- Parameter estimation method 1:
  A variant of the perceptron algorithm

A Variant of the Perceptron Algorithm

Inputs: Training set (x_i, y_i) for $i = 1 \ldots n$

Initialization: $\mathbf{v} = 0$

Define: $F(x) = \mathrm{argmax}_{y \in \mathbf{GEN}(x)} \mathbf{f}(x, y) \cdot \mathbf{v}$

Algorithm: For $t = 1 \ldots T$, $i = 1 \ldots n$

$z_i = F(x_i)$

If $(z_i \neq y_i)$ $\quad \mathbf{v} = \mathbf{v} + \mathbf{f}(x_i, y_i) - \mathbf{f}(x_i, z_i)$

Output: Parameters $\mathbf{v}$

Perceptron Experiments: Parse Reranking

Parsing the Wall Street Journal Treebank

Training set = 40,000 sentences, test = 2,416 sentences
Generative model (Collins 1999): 88.2% F-measure
Reranked model: 89.5% F-measure (11% relative error reduction)

- Results from Charniak and Johnson, 2005:
  - Improvement from 89.7% (baseline generative model) to 91.0% accuracy
  - Gains from improved n-best lists, better features, better baseline model

Summary

- A new framework: **global linear models**
  $\mathbf{GEN}, \mathbf{f}, \mathbf{v}$
- There are several ways to train the parameters $\mathbf{v}$:
  - Perceptron
  - Boosting
  - Log-linear models (maximum-likelihood)
- Applications:
  - Parsing
  - Generation
  - Machine translation
  - Tagging problems
  - Speech recognition