Large–Margin Methods for Natural Language Learning

Michael Collins
AT&T Labs-Research

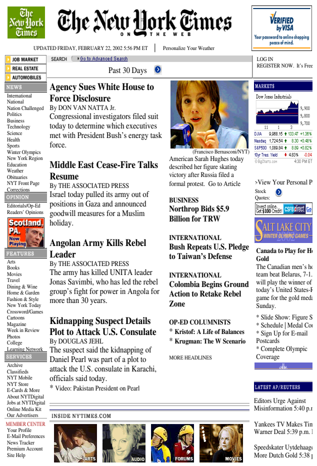
MOTIVATION: APPLICATIONS

- Sequential data is **Everywhere**

Biological data

UCUCUGCCCCAGAGCGAUACACGUGUCCGAACGUCAC
UAAGUGUCCGCGCUAGGGUGAUGACAGUCGUGCC. . .

Text



NEED TO INDUCE STRUCTURE

SEGMENTATIONS

UCUCUGCCCCAGAGCGAUACACG
UGUCCGAACGUCACUAAGUGUCC
GCGCUAGGGUGAUGACAGUCGUG

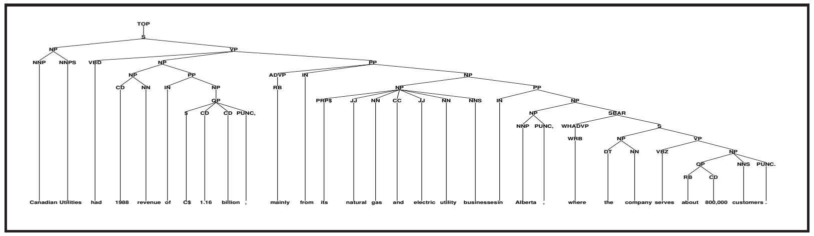
⇒

UCUCUGCCCCAGAGCGAUACACG
UGUCCGAACGUCACUAAGUGUCC
GCGCUAGGGUGAUGACAGUCGUG

HIERARCHICAL STRUCTURES (TREES)

Canadian Utilities had 1988 revenue of C$ 1.16 billion, mainly from its natural gas and electric utility businesses in Alberta, where the company serves about 800,000 customers.

⇒



OVERVIEW

- **The basic goal:** Supervised learning for recovering structure
  - Annotate examples of input/output pairs
  - Apply methods which generalize from these examples to learn the desired function

- **A challenge:** Not like "typical" problems in statistics
  - Functions from strings to other discrete structures

- **Some new methods:**
  - How the voted perceptron algorithm can be applied
  - **Kernels** over discrete structures
  - Efficient training of weighted automata
  - Advantages over existing methods:
    new representations, discriminative parameter estimation

THE BASIC GOAL

- Supervised learning for recovering structure
  - Annotate examples of input/output pairs
  - Apply methods which generalize from these examples to learn the desired function

STRINGS TO TAGGED SEQUENCES

a b e e a f h j $\Rightarrow$ a/C b/D e/C e/C a/D f/C h/D j/C

Linguistic Annotations

the/D man/N bit/V the/D dog/N

Speech Recognition

Acoustic sequences $\Rightarrow$ Phoneme sequences $\Rightarrow$ Word sequences

STRINGS TO SEGMENTATIONS

a b e e a f h j ⇒ a [b e] e [a] [f] h j

Segmentation can be treated as tagging:

a b e e a f h j ⇒ a/N b/S e/C e/N a/S f/S h/N j/N

N = none, S = start, C = continue

SEGMENTATION EXAMPLES

Linguistic Annotations

[Josef Stalin] was sentenced to 3 years exile in [Siberia]

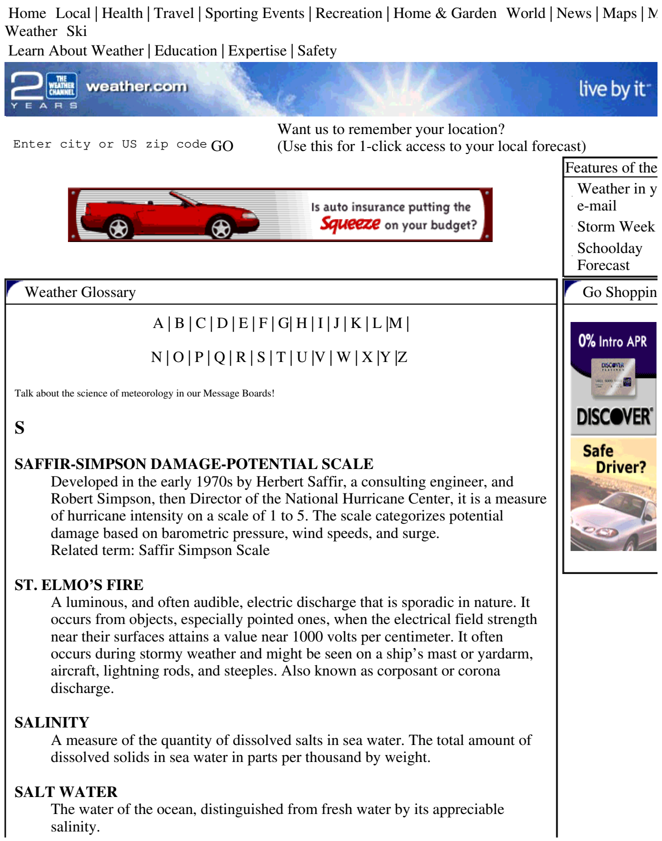
[It] had [a perfectly round door] like [a porthole].

Web Page Segmentations

Finding structure underlying FAQs, Glossaries etc.

EXTRACTING GLOSSARY ENTRIES FROM THE WEB

Input:



Home Local | Health | Travel | Sporting Events | Recreation | Home & Garden World | News | Maps | M
Weather Ski
Learn About Weather | Education | Expertise | Safety

weather.com live by it

Enter city or US zip code GO

Want us to remember your location?
(Use this for 1-click access to your local forecast)

Is auto insurance putting the Squeeze on your budget?

Features of the
Weather in y
e-mail
Storm Week
Schoolday Forecast

Go Shoppin

0% Intro APR
DISCOVER
Safe Driver?

Weather Glossary

A | B | C | D | E | F | G| H | I | J | K | L |M |

N | O | P | Q | R | S | T | U |V | W | X |Y |Z

Talk about the science of meteorology in our Message Boards!

S

SAFFIR-SIMPSON DAMAGE-POTENTIAL SCALE
Developed in the early 1970s by Herbert Saffir, a consulting engineer, and Robert Simpson, then Director of the National Hurricane Center, it is a measure of hurricane intensity on a scale of 1 to 5. The scale categorizes potential damage based on barometric pressure, wind speeds, and surge.
Related term: Saffir Simpson Scale

ST. ELMO'S FIRE
A luminous, and often audible, electric discharge that is sporadic in nature. It occurs from objects, especially pointed ones, when the electrical field strength near their surfaces attains a value near 1000 volts per centimeter. It often occurs during stormy weather and might be seen on a ship's mast or yardarm, aircraft, lightning rods, and steeples. Also known as corposant or corona discharge.

SALINITY
A measure of the quantity of dissolved salts in sea water. The total amount of dissolved solids in sea water in parts per thousand by weight.

SALT WATER
The water of the ocean, distinguished from fresh water by its appreciable salinity.

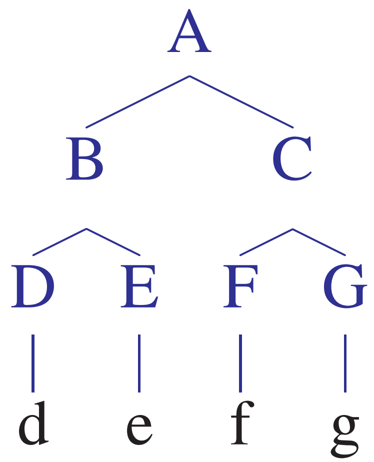
Output: **St. Elmo's Fire:** A luminous, and often audible, electric discharge that is sporadic in nature. It occurs from objects, especially pointed ones, when the electrical field strength near their surfaces attains a value near 100 volts per centimeter...

(Michael Littman's idea for answering "What is" questions)

STRINGS TO TREES

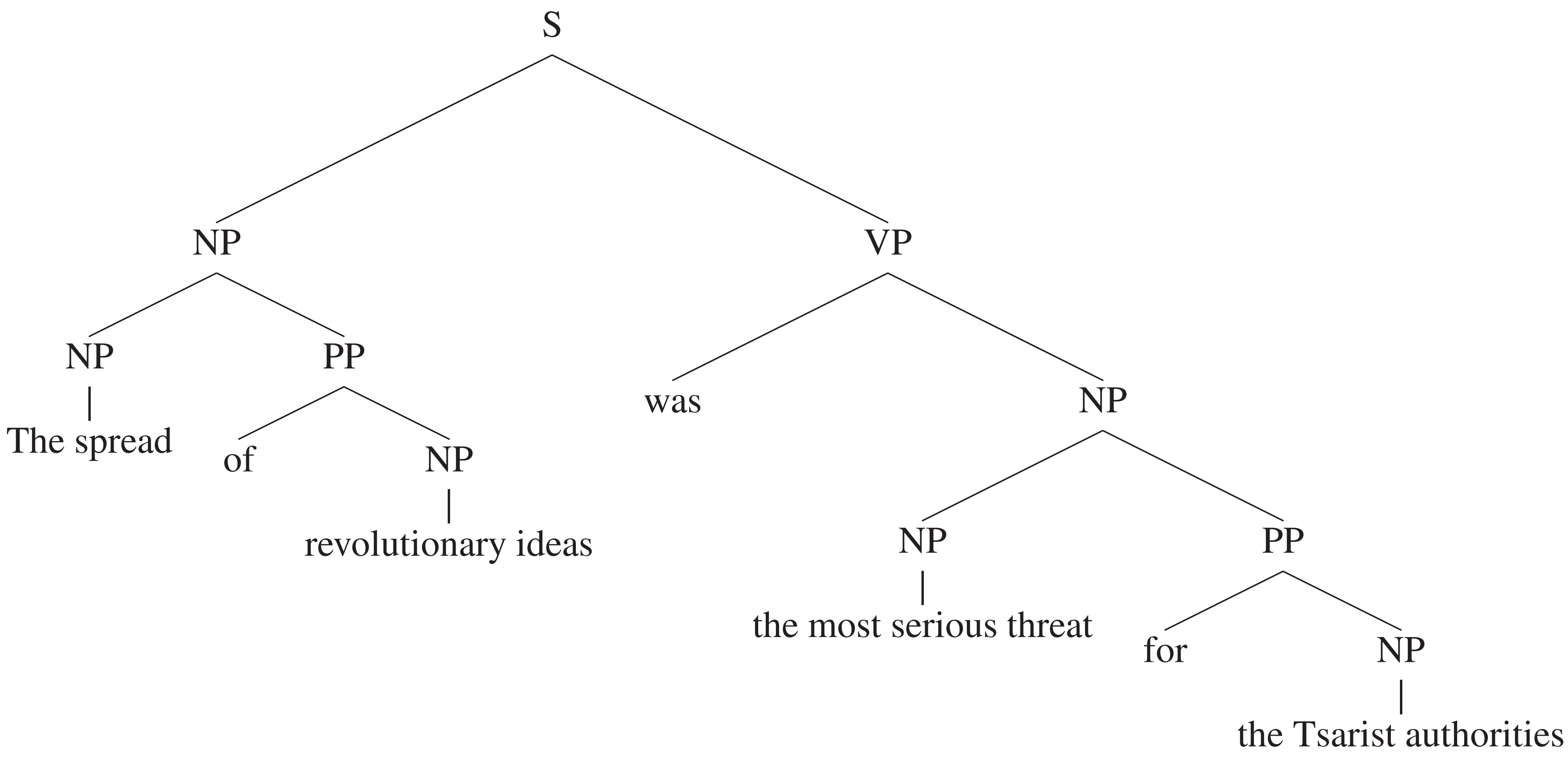
d e f g $\Rightarrow$ (A (B (D d) (E e)) (C (F f) (G g)))

d e f g $\Rightarrow$



EXAMPLES

Syntactic Structure



LANGUAGE GENERATION [Walker et al. 2001]

Input = text plan

```
implicit-confirm(orig-city:NEWARK)
implicit-confirm(dest-city:DALLAS)
implicit-confirm(month:9)
implicit-confirm(day-number:1)
request(depart-time)
```

Output = text

What time would you like to travel on September the 1st to Dallas from Newark?
Leaving on September 1st. What time would you like to travel from Newark to Dallas?
Leaving in September. Leaving on the 1st. What time would you, traveling from Newark to Dallas, like to leave?

LEARNING APPROACHES

- Basic idea:
  Annotate examples of desired input/output behavior
  Use statistical methods to induce a function with this behavior

- Long history of this methodology in speech recognition

- Why not build systems by hand?
  The problems are surprisingly complex:

  - Large number of rules
  - Rules interact in subtle ways

A CHALLENGE

- Interesting class of learning problems:

  Functions from strings to other discrete structures

- "Typical" problems in statistics:
  - Classification functions $F : \mathbb{R}^n \to \{-1, +1\}$
  - Regression functions $F : \mathbb{R}^n \to \mathbb{R}$

COMMON METHODS

- Stochastic automata
  - Probabilistic finite-state machines
    Hidden Markov Models (HMMs)
  - Probabilistic context-free grammars
    PCFGs
- Limitations
  - Representation
  - Strong assumptions in parameter estimation

NEW METHODS

- Large-margin methods
  (Support vector machines, Voted Perceptron, Boosting)

- Questions:
  - Can these methods be derived for tagging, parsing?
  - Do the statistical guarantees (e.g. about generalization) apply to tagging and parsing?
  - What representations can be used?

MAIN RESULTS

- A variant of the voted perceptron algorithm for parsing, tagging, other NLP problems
- Experiments: Improvements over state-of-the-art baselines, using "global" features
- New kernels for discrete structures such as trees (e.g., the "all subtrees" representation)
- Experiments: Improvements over state-of-the-art baselines, using these kernels
- Efficient discriminative algorithms for weighted automata

LINEAR MODELS FOR TAGGING AND PARSING

- Three components:

  GEN is a function from a string to a set of candidates

  Φ maps a candidate to a feature vector

  W is a parameter vector

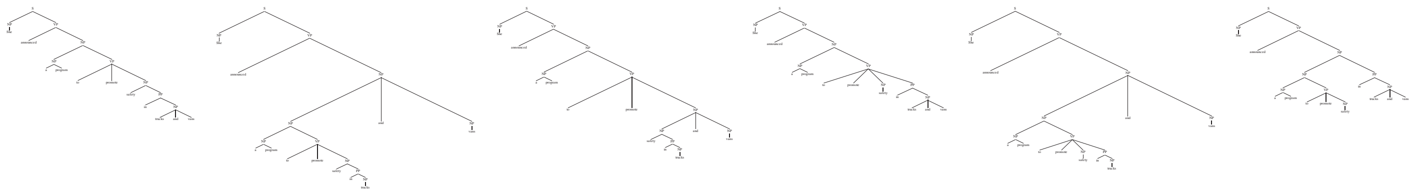
Related to Markov Random Field approaches
[Johnson, Geman, Canon, Chi, and Riezler 1999]
[Abney 1997]
[Della Pietra, Della Pietra, and Lafferty 1997]

COMPONENT 1: GEN

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

She announced a program to promote safety in trucks and vans

$\Downarrow$ **GEN**



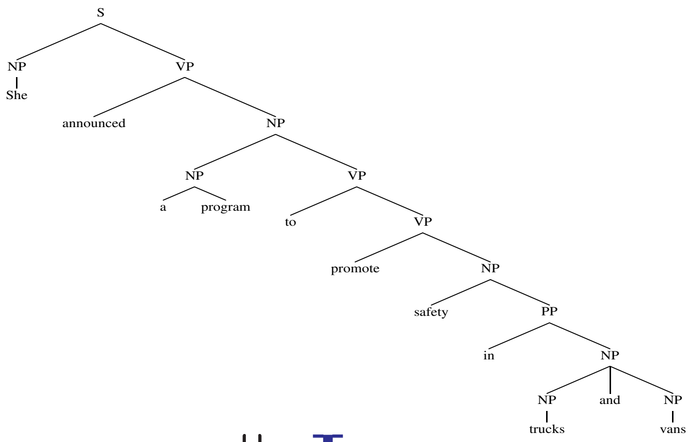
Note: Optimality Theory [Prince and Smolensky]

EXAMPLES OF GEN

- A context-free grammar
- A finite-state machine
- Top N most probable analyses from a probabilistic grammar

COMPONENT 2: Φ

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

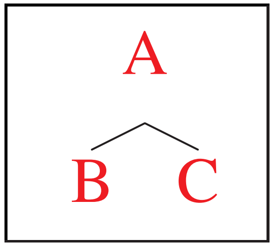


$$\Downarrow \Phi$$

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

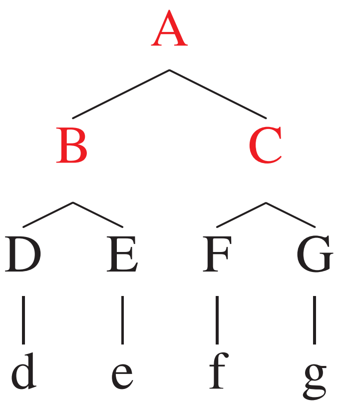
FEATURES

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

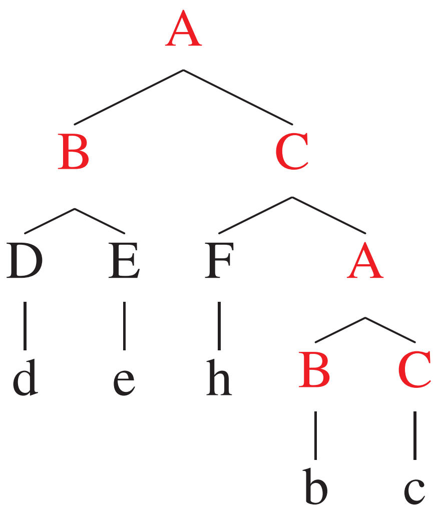
$h(x) =$ Number of times  is seen in x

---

T_1



T_2



$h(T_1) = 1$

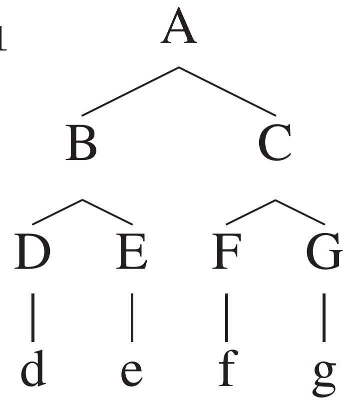
$h(T_2) = 2$

FEATURE VECTORS

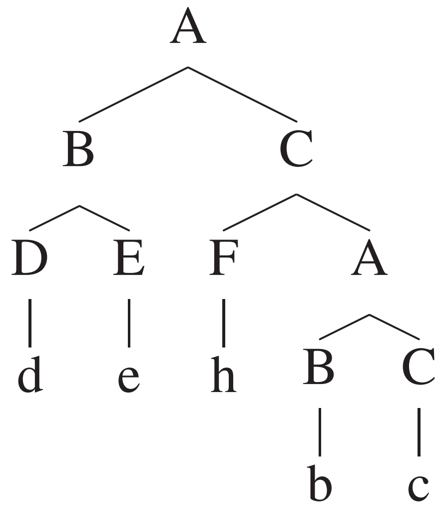
- A set of functions $h_1 \ldots h_d$ define a **feature vector**

$$\Phi(x) = \langle h_1(x), h_2(x) \ldots h_d(x) \rangle$$

T_1



T_2



$$\Phi(T_1) = \langle 1, 0, 0, 3 \rangle \qquad \Phi(T_2) = \langle 2, 0, 1, 1 \rangle$$

A NEED TO DEFINE ARBITRARY FEATURES

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

Constituents with similar structure tend to be coordinated

$h(x) =$ Number of instances of non-parallel coordination in x

	Bars in New York and pubs in London	h(x) = 0
vs.	Bars in New York and pubs	h(x) = 1

Example 2 Semantic features

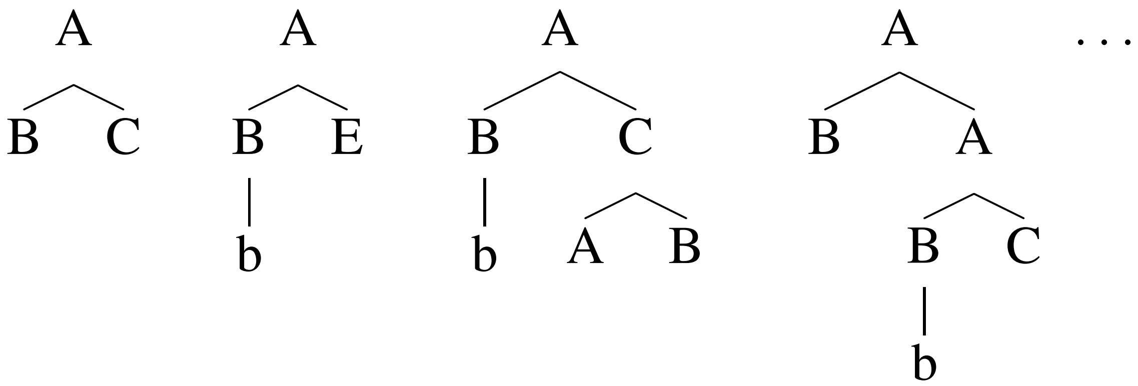
$h(x) =$ Number of times **pour** takes an object with a `+liquid` feature

	pour the **cappucino**	h(x) = 1
vs.	pour the **book**	h(x) = 0

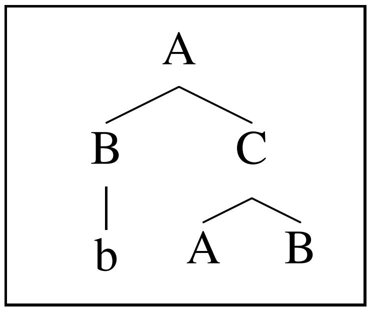
“ALL SUBTREES” REPRESENTATION

- Given: Non-Terminal symbols $\{A, B, \ldots\}$
  Terminal symbols $\{a, b, c \ldots\}$

- An infinite set of subtrees



- An infinite set of features, e.g.,

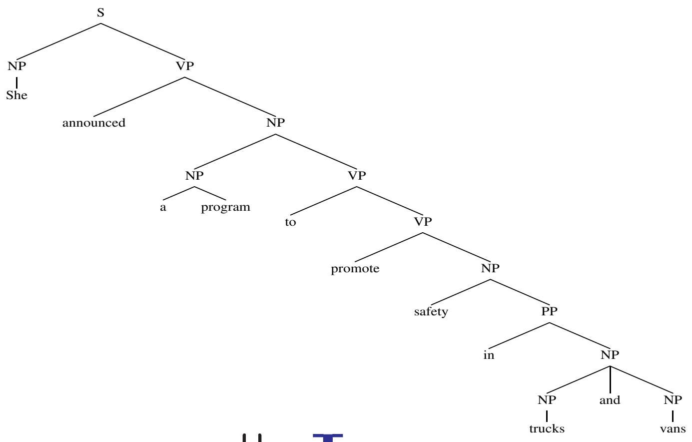
  $h_3(x) =$ Number of times  is seen in x

Representation studied by [Bod 98]

COMPONENT 3: W

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

---



$$\Downarrow \mathbf{\Phi}$$

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

$$\Downarrow \mathbf{\Phi} \cdot \mathbf{W}$$

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

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 $\mathbf{F} : \mathcal{X} \rightarrow \mathcal{Y}$

- $\mathbf{GEN}$, $\mathbf{\Phi}$, $\mathbf{W}$ define

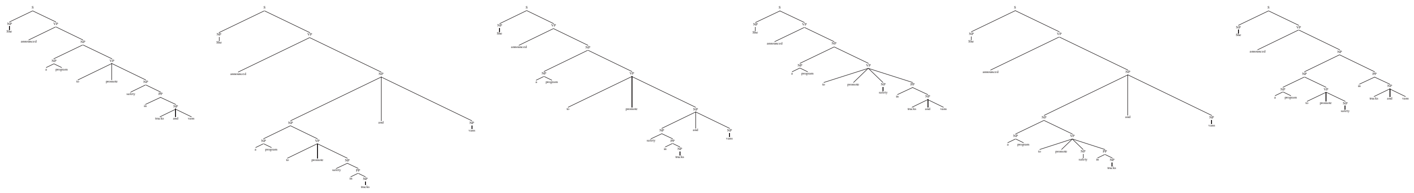
$$\mathbf{F}(x) = \arg\max_{y \in \mathbf{GEN}(x)} \mathbf{\Phi}(y) \cdot \mathbf{W}$$

Choose the highest scoring tree as the most plausible structure

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

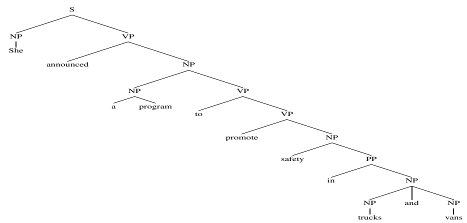
She announced a program to promote safety in trucks and vans

$\Downarrow$ **GEN**



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

$\Downarrow$ arg max



PREVIOUS WORK ON PARAMETER ESTIMATION

- **Markov Random Fields** [Johnson et. al 1999]

$$P(y_i \mid x_i, \mathbf{W}) = \frac{e^{\mathbf{\Phi}(y_i)\cdot\mathbf{W}}}{\sum_{y\in\mathbf{GEN}(x_i)} e^{\mathbf{\Phi}(y)\cdot\mathbf{W}}}$$

- **Boosting** (ICML 2000 paper):

Minimize: $$\sum_i \sum_{z\in\mathbf{GEN}(x_i), z\neq y_i} e^{\mathbf{W}\cdot\mathbf{\Phi}(z) - \mathbf{W}\cdot\mathbf{\Phi}(y_i)}$$

- Close connections between the two:
  [Friedman, Hastie, and Tibshirani 1998, Lafferty 1999]
  (COLT 2000 paper with Rob Schapire, Yoram Singer)

A VARIANT OF THE PERCEPTRON ALGORITHM

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

Initialization: $\mathbf{W} = 0$

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

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

$z_i = \mathbf{F}(x_i)$

If $(z_i \neq y_i)$ $\quad \mathbf{W} = \mathbf{W} + \mathbf{\Phi}(y_i) - \mathbf{\Phi}(z_i)$

Output: Parameters $\mathbf{W}$

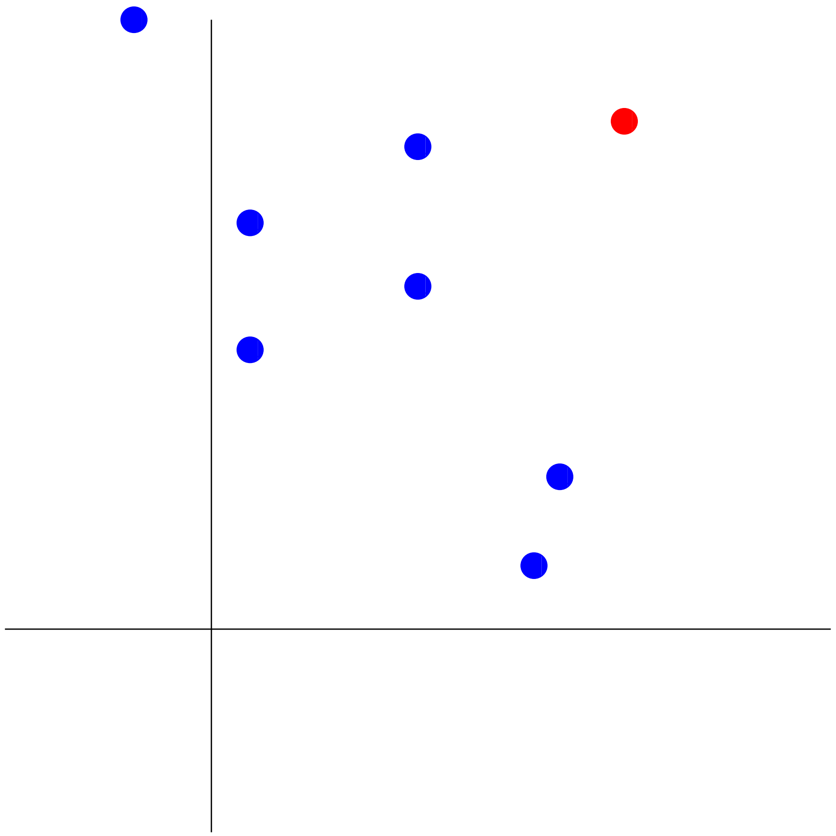
THEORY UNDERLYING THE ALGORITHM

- **Definition:** $\overline{\mathbf{GEN}}(x_i) = \mathbf{GEN}(x_i) - \{y_i\}$

- **Definition:** The training set is **separable with margin** δ, if there is a vector $\mathbf{U} \in \mathbb{R}^d$ with $||\mathbf{U}|| = 1$ such that

$$\forall i, \forall z \in \overline{\mathbf{GEN}}(x_i) \quad \mathbf{U} \cdot \mathbf{\Phi}(y_i) - \mathbf{U} \cdot \mathbf{\Phi}(z) \geq \delta$$

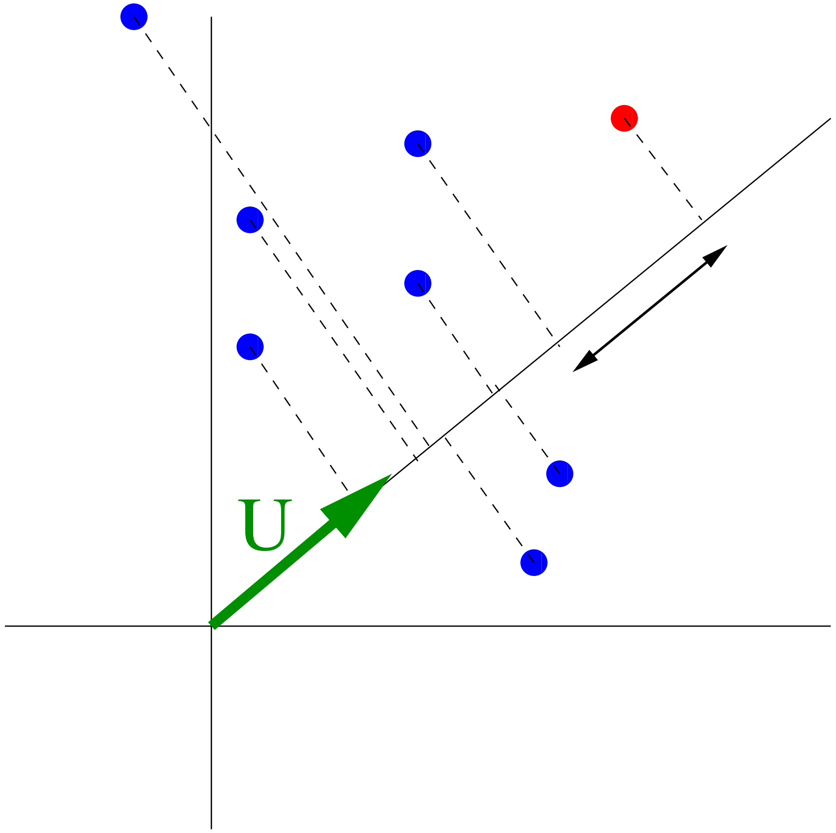
GEOMETRIC INTUITION BEHIND SEPARATION



= Correct candidate

= Incorrect candidates

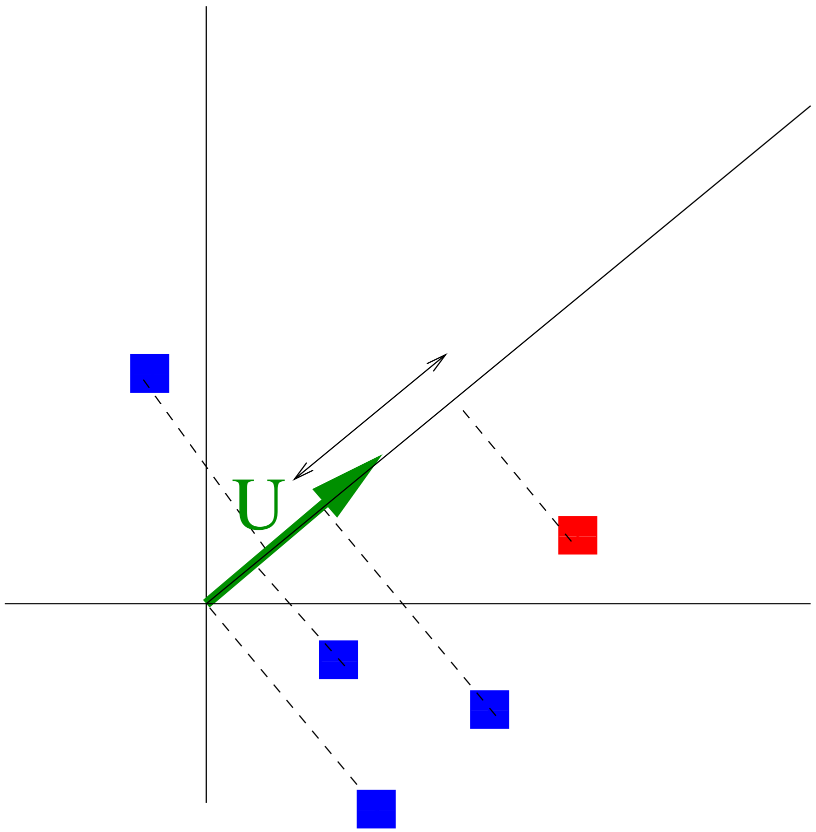
GEOMETRIC INTUITION BEHIND SEPARATION



= Correct candidate

= Incorrect candidates

ALL EXAMPLES ARE SEPARATED



= Correct candidate (2)

= Incorrect candidates (2)

THEORY UNDERLYING THE ALGORITHM

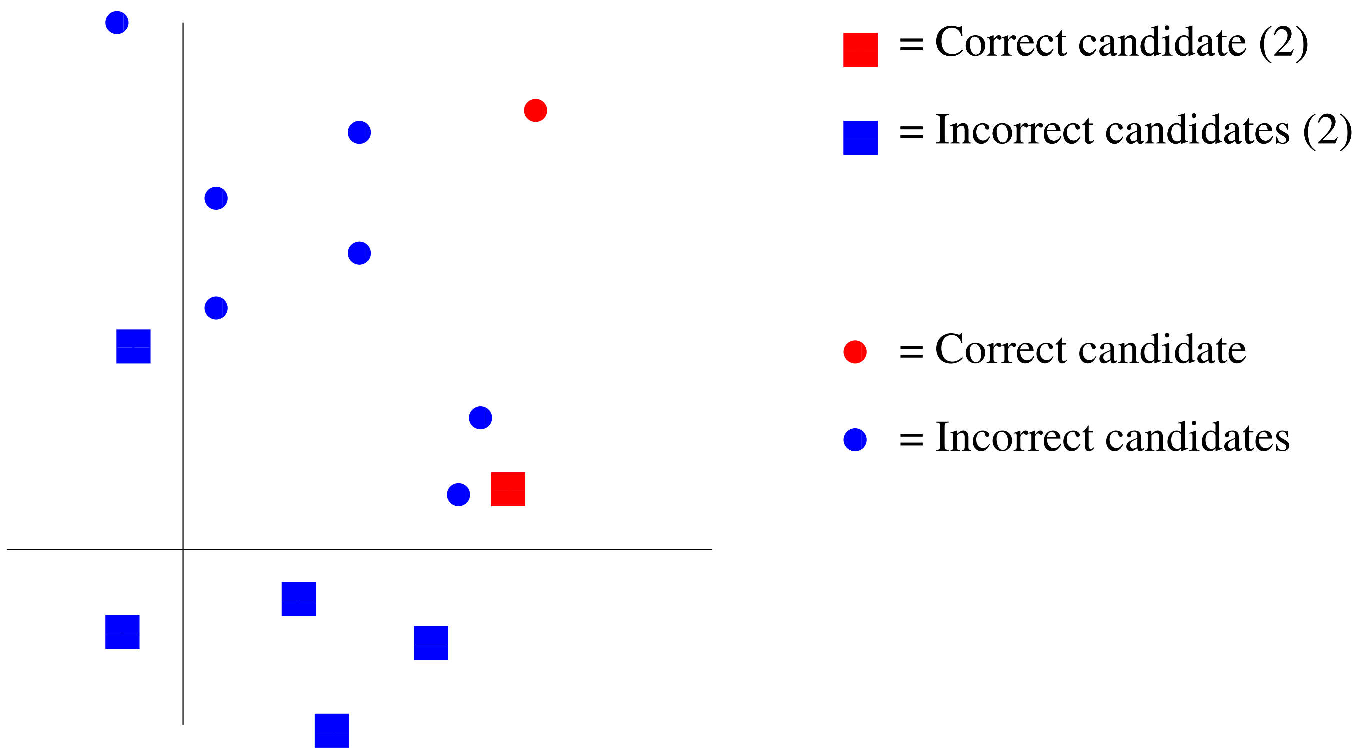
Theorem: For any training sequence (x_i, y_i) which is separable with margin δ, then for the perceptron algorithm

$$\text{Number of mistakes} \leq \frac{R^2}{\delta^2}$$

where R is a constant such that $\forall i, \forall z \in \overline{\mathbf{GEN}}(x_i) \quad ||\mathbf{\Phi}(y_i) - \mathbf{\Phi}(z)|| \leq R$

Proof: Direct modification of the proof for the classification case.

WHAT WE'RE NOT DOING



- We aim to separate the correct structure from its competitors
- We're not aiming to do **classification** of good vs. bad parses
  This is a harder (and unnecessary) problem

Proof:

Let $\mathbf{W}^k$ be the weights before the k'th mistake. $\mathbf{W}^1 = 0$
If the k'th mistake is made at i'th example,
and $z_i = \text{argmax}_{y \in \mathbf{GEN}(x_i)} \mathbf{\Phi}(y) \cdot \mathbf{W}^k$, then

$$\begin{aligned}
\mathbf{W}^{k+1} &= \mathbf{W}^k + \mathbf{\Phi}(y_i) - \mathbf{\Phi}(z_i) \\
\Rightarrow \mathbf{U} \cdot \mathbf{W}^{k+1} &= \mathbf{U} \cdot \mathbf{W}^k + \mathbf{U} \cdot \mathbf{\Phi}(y_i) - \mathbf{U} \cdot \mathbf{\Phi}(z_i) \\
&\geq \mathbf{U} \cdot \mathbf{W}^k + \delta \\
&\geq k\delta \\
\Rightarrow ||\mathbf{W}^{k+1}|| &\geq k\delta
\end{aligned}$$

Also,

$$\begin{aligned}
||\mathbf{W}^{k+1}||^2 &= ||\mathbf{W}^k||^2 + ||\mathbf{\Phi}(y_i) - \mathbf{\Phi}(z_i)||^2 + 2\mathbf{W}^k \cdot (\mathbf{\Phi}(y_i) - \mathbf{\Phi}(z_i)) \\
&\leq ||\mathbf{W}^k||^2 + R^2 \\
\Rightarrow ||\mathbf{W}^{k+1}||^2 &\leq kR^2 \\
\Rightarrow k^2\delta^2 &\leq ||\mathbf{W}^{k+1}||^2 \leq kR^2 \\
\Rightarrow k &\leq R^2/\delta^2
\end{aligned}$$

MORE THEORY FOR THE PERCEPTRON ALGORITHM

- Question 1: what if the data is not separable?

[Freund and Schapire 99] give a modified theorem for this case

- Question 2: performance on training data is all very well, but what about performance on new test examples?

Assume some distribution $P(x, y)$ underlying examples

Theorem [Helmbold and Warmuth 95]: For any distribution $P(x, y)$ generating examples, if e = expected number of mistakes of an online algorithm on a sequence of $m + 1$ examples, then a randomized algorithm trained on m samples will have probability $\frac{e}{m+1}$ of making an error on a newly drawn example from P.

[Freund and Schapire 99] use this to define the **Voted Perceptron**

QUESTION 1: WHAT IF THE DATA IS NOT SEPARABLE?

- A modified theorem follows from [Freund and Schapire 99]

For any $\mathbf{U}$, δ pair with $||\mathbf{U}|| = 1$ **define**

- $m_i = \mathbf{U} \cdot \mathbf{\Phi}(y_i) - \max_{z \in \overline{\mathbf{GEN}}(x_i)} \mathbf{U} \cdot \mathbf{\Phi}(z)$
- $\epsilon_i = \begin{cases} \delta - m_i & \text{if } m_i < \delta \\ 0 & \text{otherwise} \end{cases}$
- $D(\mathbf{U}, \delta) = \sqrt{\sum_i \epsilon_i^2}$

Then it can be shown that

$$\text{Number of mistakes} \leq \min_{\mathbf{U},\delta} \frac{(R + D(\mathbf{U}, \delta))^2}{\delta^2}$$

QUESTION 2: PERFORMANCE ON NEW EXAMPLES?

Theorem [Helmbold and Warmuth 95]: For any distribution $P(x, y)$ generating examples, if e = expected number of mistakes of an online algorithm on a sequence of $m+1$ examples, then a randomized algorithm trained on m samples and applied to an $m+1$'th example will have probability $\frac{e}{m+1}$ of making an error on the $m+1$'th example.

[Freund and Schapire 99]: For the **voted perceptron**

$$\text{Probability of error} \leq \frac{2}{m+1} E\left[\min_{\mathbf{U},\delta} \frac{(R+D(\mathbf{U},\delta))^2}{\delta^2}\right]$$

where $E[\ldots]$ is the expectation taken over a sample of $(m+1)$ examples drawn from P(x,y)

QUESTION 3: ARE LINEAR SEPARATORS GOOD ENOUGH?

Example: Hand-written digit recognition

- **Kernels** give non-linear separators

  Linear model + no voting $\Rightarrow$ 13.5% error rate
  Degree 3 polynomial kernel + voting $\Rightarrow$ 1.6% error rate

- **Voting** helps in non-separable cases

  Linear model + voting $\Rightarrow$ 8.1% error rate

THREE SCENARIOS

- Reranking output from a probabilistic model
- Reranking output using **kernels**
- Training a weighted grammar

RERANKING APPROACHES

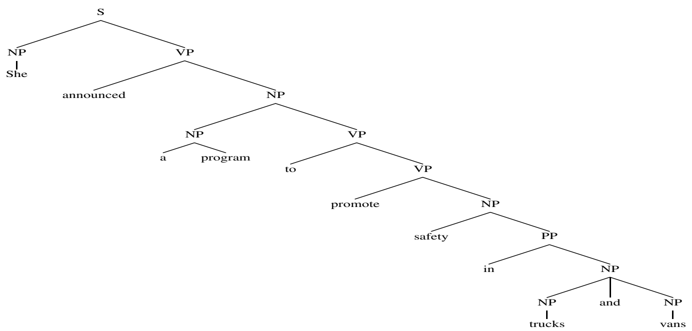
- **GEN** is the top n most probable candidates from a base model
  - Parsing: a lexicalized probabilistic context-free grammar
  - Tagging: “maximum entropy” tagger
  - Speech recognition: existing recogniser

PARSING EXPERIMENTS

GEN Beam search used to parse training and test sentences: around 27 parses for each sentence

$\Phi = \langle L(x), h_1(x) \ldots h_m(x) \rangle$, where $L(x)$ = log-likelihood from first-pass parser, $h_1 \ldots h_m$ are $\approx 500,000$ indicator functions

$$e.g., \quad h_1(x) = \begin{cases} 1 & \text{if } x \text{ contains} \langle S \rightarrow NP \;\; VP \rangle \\ 0 & \text{otherwise} \end{cases}$$



$$\Downarrow \Phi$$

$$\langle -15.65, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, \ldots 1, 0, 0 \rangle$$

NAMED ENTITIES

GEN Top 20 segmentations from a "maximum-entropy" tagger

$\mathbf{\Phi} = \langle L(x), h_1(x) \ldots h_m(x) \rangle,$

$$e.g., \quad h_1(x) = \begin{cases} 1 & \text{if } x \text{ contains a boundary} = \boxed{\text{“[The}} \\ 0 & \text{otherwise} \end{cases}$$

Whether you're an aging flower child or a clueless [Gen-Xer], "[The Day They Shot John Lennon]," playing at the [Dougherty Arts Center], entertains the imagination.

$$\Downarrow \mathbf{\Phi}$$

$$\langle -3.17, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, \ldots 0, 1, 1 \rangle$$

Whether you're an aging flower child or a clueless [Gen-Xer], "[The Day They Shot John Lennon]," playing at the [Dougherty Arts Center], entertains the imagination.

$$\Downarrow \boldsymbol{\Phi}$$

$$\langle -3.17, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, \ldots 0, 1, 1\rangle$$

Whether you're an aging flower child or a clueless Gen-Xer, "The Day [They Shot John Lennon]," playing at the [Dougherty Arts Center], entertains the imagination.

$$\Downarrow \boldsymbol{\Phi}$$

$$\langle -3.51, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, \ldots 0, 1, 0\rangle$$

Whether you're an aging flower child or a clueless [Gen-Xer], "The Day [They Shot John Lennon]," playing at the [Dougherty Arts Center], entertains the imagination.

$$\Downarrow \boldsymbol{\Phi}$$

$$\langle -2.87, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, \ldots 0, 1, 0\rangle$$

EXPERIMENTS

Parsing Wall Street Journal Treebank

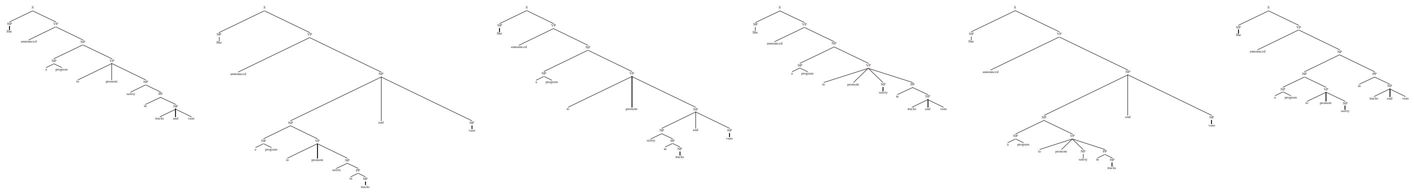
Training set = 40,000 sentences, test = 2,416 sentences
State-of-the-art parser: 88.2% F-measure
Reranked model: 89.5% F-measure (11% relative error reduction)
Boosting: 89.7% F-measure (13% relative error reduction)

Recovering Named-Entities in Web Data

Training data = 53,609 sentences (1,047,491 words),
test data = 14,717 sentences (291,898 words)
State-of-the-art tagger: 85.3% F-measure
Reranked model: 87.9% F-measure (17.7% relative error reduction)
Boosting: 87.6% F-measure (15.6% relative error reduction)

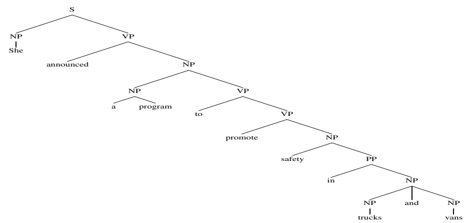
She announced a program to promote safety in trucks and vans

$\Downarrow$ **GEN**



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

$\Downarrow$ arg max



THREE SCENARIOS

- Reranking output from a probabilistic model

- Reranking output using **kernels**
  (with Nigel Duffy)

- Training a weighted grammar

A VARIANT OF THE PERCEPTRON ALGORITHM

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

Initialization: $\mathbf{W} = 0$

Define: $\mathbf{F}(x) = \text{argmax}_{y \in \mathbf{GEN}(x)} \mathbf{\Phi}(y) \cdot \mathbf{W}$

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

$z_i = \mathbf{F}(x_i)$

If $(z_i \neq y_i)$ $\quad \mathbf{W} = \mathbf{W} + \mathbf{\Phi}(y_i) - \mathbf{\Phi}(z_i)$

Output: Parameters $\mathbf{W}$

“KERNEL” FORM OF THE PERCEPTRON ALGORITHM

- Manipulate sets $\mathcal{A}$ and $\mathcal{B}$ of training examples such that

$$\mathbf{W} = \sum_{a \in \mathcal{A}} \mathbf{\Phi}(a) - \sum_{b \in \mathcal{B}} \mathbf{\Phi}(b)$$

$$\mathbf{W} \cdot \mathbf{\Phi}(y) = \left(\sum_{a \in \mathcal{A}} \mathbf{\Phi}(y) \cdot \mathbf{\Phi}(a) - \sum_{b \in \mathcal{B}} \mathbf{\Phi}(y) \cdot \mathbf{\Phi}(b) \right)$$

- Intuition:
  - $\mathcal{A}$ is a set of “good parses”
  - $\mathcal{B}$ is a set of “bad parses”
  - $\mathbf{\Phi}(y) \cdot \mathbf{\Phi}(x)$ is a similarity measure

“KERNEL” FORM OF THE PERCEPTRON ALGORITHM

Define:

$$\mathbf{F}(x) = \arg\max_{y \in \mathbf{GEN}(x)} \left(\sum_{a \in \mathcal{A}} \mathbf{\Phi}(y) \cdot \mathbf{\Phi}(a) - \sum_{b \in \mathcal{B}} \mathbf{\Phi}(y) \cdot \mathbf{\Phi}(b) \right)$$

Initialization: $\mathcal{A} = \{\}, \mathcal{B} = \{\}$

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

$z_i = \mathbf{F}(x_i)$

If $(z_i \neq y_i)$ $\quad \mathcal{A} = \mathcal{A} \cup \{y_i\}$,

$\mathcal{B} = \mathcal{B} \cup \{z_i\}$

Output: Sets $\mathcal{A}$ and $\mathcal{B}$

Equivalence: $\mathbf{W} = \sum_{a \in \mathcal{A}} \mathbf{\Phi}(a) - \sum_{b \in \mathcal{B}} \mathbf{\Phi}(b)$

Equivalence: $\mathbf{W} = \sum_{a \in \mathcal{A}} \mathbf{\Phi}(a) - \sum_{b \in \mathcal{B}} \mathbf{\Phi}(b)$

Original Form

Initialization: $\mathbf{W} = 0$

Define: $\mathbf{F}(x) = \text{argmax}_{y \in \mathbf{GEN}(x)} \mathbf{\Phi}(y) \cdot \mathbf{W}$

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

$z_i = \mathbf{F}(x_i)$

If $(z_i \neq y_i)$ $\quad \mathbf{W} = \mathbf{W} + \mathbf{\Phi}(y_i) - \mathbf{\Phi}(z_i)$

Dual Form

Initialization: $\mathcal{A} = \{\}, \mathcal{B} = \{\}$

Define: $\mathbf{F}(x) = \text{argmax}_{y \in \mathbf{GEN}(x)} \left(\sum_{a \in \mathcal{A}} \mathbf{\Phi}(y) \cdot \mathbf{\Phi}(a) - \sum_{b \in \mathcal{B}} \mathbf{\Phi}(y) \cdot \mathbf{\Phi}(b)\right)$

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

$z_i = \mathbf{F}(x_i)$

If $(z_i \neq y_i)$ $\quad \mathcal{A} = \mathcal{A} \cup \{y_i\}, \mathcal{B} = \mathcal{B} \cup \{z_i\}$

MOTIVATION FOR THE KERNEL FORM

- Let $N = \sum_i |\mathbf{GEN}(x_i)|$
  (i.e., size of the training set)

- One pass over training set takes $O(Nd)$ time for regular form

- Say it takes I time to calculate $\mathbf{\Phi}(x) \cdot \mathbf{\Phi}(y)$, perceptron makes M mistakes, then dual form takes $O(NMI)$ time

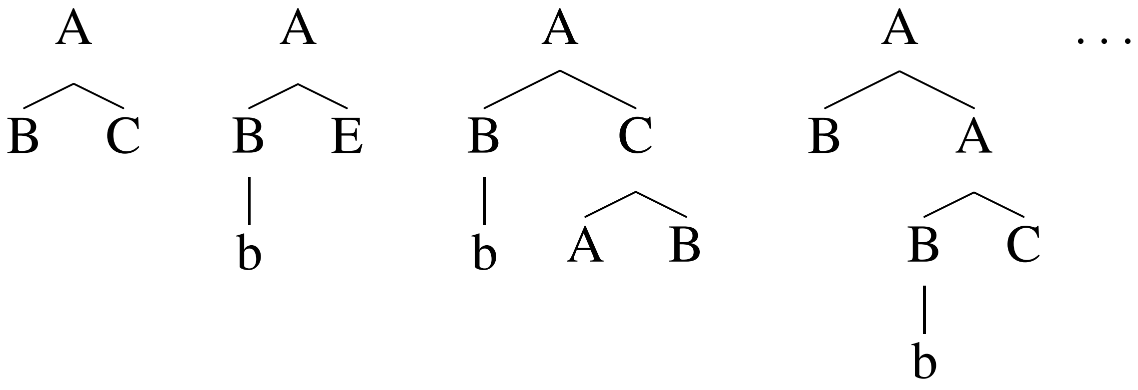
- For some representations, $MI << d$

If we can compute $\mathbf{\Phi}(x) \cdot \mathbf{\Phi}(y)$ efficiently
we can learn efficiently with the representation $\mathbf{\Phi}$

“ALL SUBTREES” REPRESENTATION

- Given: Non-Terminal symbols $\{A, B, \ldots\}$
  Terminal symbols $\{a, b, c \ldots\}$

- An infinite set of subtrees



- **Step 1:**
  Choose an (arbitrary) mapping from subtrees to integers

  $h_i(x) =$ Number of times subtree i is seen in x

  $\mathbf{\Phi}(x) = \langle h_1(x), h_2(x), h_3(x) \ldots \rangle$

ALL SUBTREES REPRESENTATION

- $\mathbf{\Phi}$ is now huge

- **But** inner product $\mathbf{\Phi}(T_1) \cdot \mathbf{\Phi}(T_2)$ can be computed efficiently using dynamic programming.

COMPUTING THE INNER PRODUCT

Define – N_1 and N_2 are sets of nodes in T_1 and T_2 respectively.

– $I_i(x) = \begin{cases} 1 \text{ if } i\text{'th subtree is rooted at } x. \\ 0 \text{ otherwise.} \end{cases}$

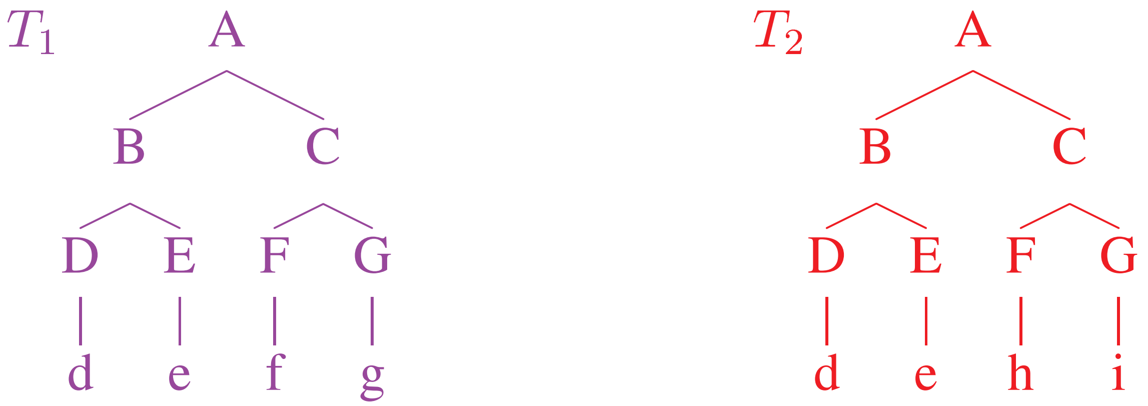
Follows that:

$h_i(T_1) = \sum_{n_1 \in N_1} I_i(n_1)$ and $h_i(T_2) = \sum_{n_2 \in N_2} I_i(n_2)$

$$\begin{aligned}
\mathbf{\Phi}(T_1) \cdot \mathbf{\Phi}(T_2) = \sum_i h_i(T_1) h_i(T_2) &= \sum_i \left(\sum_{n_1 \in N_1} I_i(n_1)\right) \left(\sum_{n_2 \in N_2} I_i(n_2)\right) \\
&= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \sum_i I_i(n_1) I_i(n_2) \\
&= \sum_{n_1 \in N_1} \sum_{n_2 \in N_2} \Delta(n_1, n_2)
\end{aligned}$$

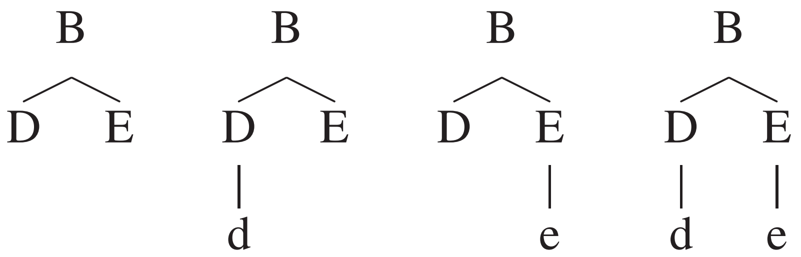
where $\Delta(n_1, n_2) = \sum_i I_i(n_1) I_i(n_2)$ is the number of common subtrees at n_1, n_2

AN EXAMPLE



$$\mathbf{\Phi}(T_1) \cdot \mathbf{\Phi}(T_2) = \Delta(A, A) + \Delta(A, B) \ldots + \Delta(B, A) + \Delta(B, B) \ldots + \Delta(G, G)$$

- Most of these terms are 0 (e.g. $\Delta(A, B)$).
- Some are non-zero, e.g. $\Delta(B, B) = 4$



RECURSIVE DEFINITION OF $\Delta(n_1, n_2)$

- If the productions at n_1 and n_2 are different

$$\Delta(n_1, n_2) = 0$$

- Else if n_1, n_2 are pre-terminals,

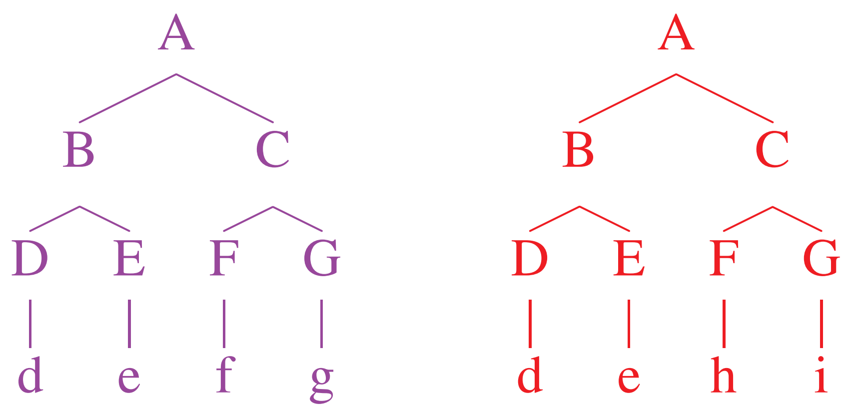
$$\Delta(n_1, n_2) = 1$$

- Else

$$\Delta(n_1, n_2) = \prod_{j=1}^{nc(n_1)} (1 + \Delta(ch(n_1, j), ch(n_2, j)))$$

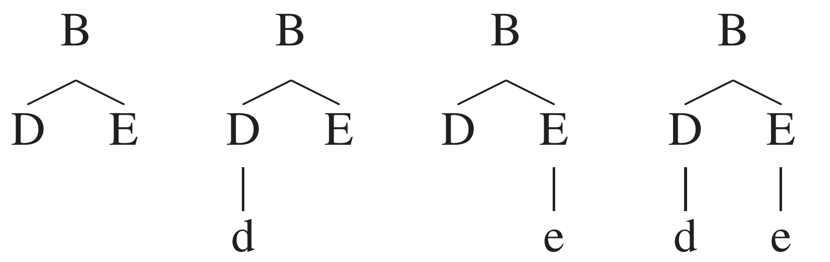
$nc(n_1)$ is number of children of node n_1;
$ch(n_1, j)$ is the j'th child of n_1.

ILLUSTRATION OF THE RECURSION

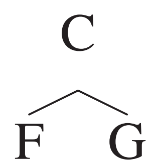


How many subtrees do nodes A and A have in common? i.e., What is $\Delta(A, A)$?

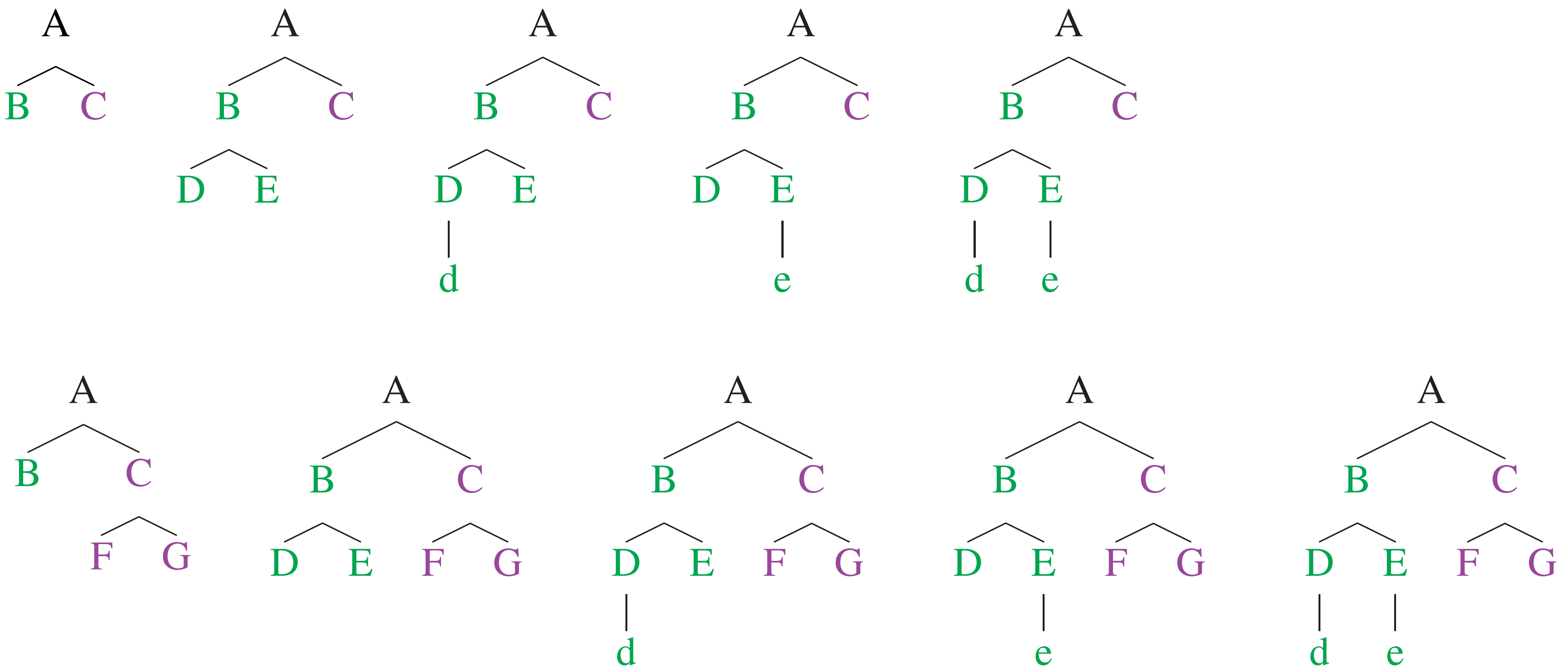
$$\Delta(B, B) = 4$$



$$\Delta(C, C) = 1$$



$$\boxed{\Delta(A, A) = (\Delta(B, B) + 1) \times (\Delta(C, C) + 1) = 10}$$


A
B C
A
B C
D E
A
B C
D E
d
A
B C
D E
e
A
B C
D E
d e
A
B C
F G
A
B C
D E F G
A
B C
D E F G
d
A
B C
D E F G
e
A
B C
D E F G
d e

SIMILAR KERNELS CAN BE DERIVED FOR TAGGED SEQUENCES

Whether you're an aging flower child or a clueless [Gen-Xer], "[The Day They Shot John Lennon]," playing at the [Dougherty Arts Center], entertains the imagination.

$$\Downarrow \Phi$$

$\boxed{\textbf{Whether}}$ $\boxed{\textbf{[Gen-Xer],}}$ $\boxed{\textbf{Day They}}$ $\boxed{\textbf{John Lennon],'' playing}}$

$\boxed{\textbf{Whether you're an aging flower child or a clueless [Gen}}$

. . .

EXPERIMENTS

Parsing Wall Street Journal Treebank

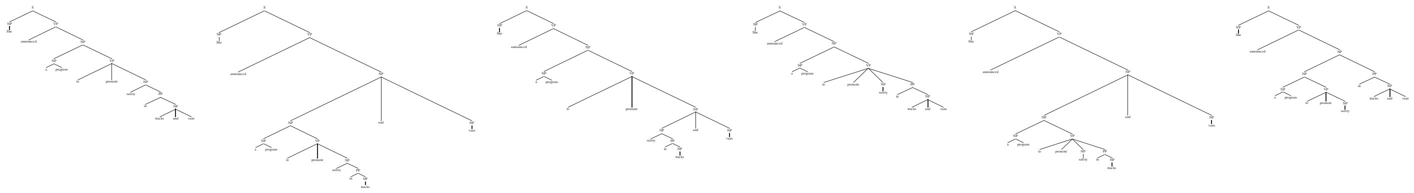
Training set = 40,000 sentences, test = 2,416 sentences
State-of-the-art parser: 88.5% F-measure
Reranked model: 89.1% F-measure
(5% relative error reduction)

Recovering Named-Entities in Web Data

Training data = 53,609 sentences (1,047,491 words),
test data = 14,717 sentences (291,898 words)
State-of-the-art tagger: 85.3% F-measure
Reranked model: 87.6% F-measure
(15.6% relative error reduction)

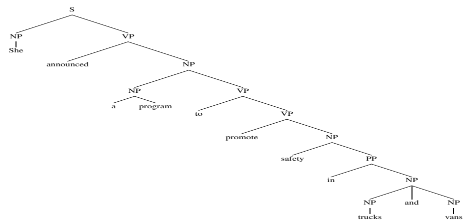
She announced a program to promote safety in trucks and vans

$\Downarrow$ **GEN**



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

$\Downarrow$ arg max



THREE SCENARIOS

- Reranking output from a probabilistic model
- Reranking output using **kernels**
- Training a weighted grammar

TRAINING WEIGHTED AUTOMATA

- Main computational expense:

$$\mathbf{F}(x) = \arg\max_{y \in \mathbf{GEN}(x)} \mathbf{\Phi}(y) \cdot \mathbf{W}$$

- Worst case: O($|\mathbf{GEN}(x)|$) time

- Can be better for some **GEN**, $\mathbf{\Phi}$ combinations

EXAMPLE: TRAINING AN HMM

- Given: m state symbols $\{A, B, \ldots\}$

- A sentence of length n has m^n possibilities

  a b e e a f h j $\Rightarrow$

  a/A b/A e/A e/A a/A f/A h/A j/A

  a/B b/A e/A e/A a/A f/A h/A j/A

  a/C b/A e/A e/A a/A f/A h/A j/A

  a/A b/B e/A e/A a/A f/A h/A j/A

  a/B b/B e/A e/A a/A f/A h/A j/A

  a/C b/B e/A e/A a/A f/A h/A j/A

  ...

THE VITERBI ALGORITHM

- Decode from left-to-right: potentially maintain m^p prefixes for the first p words:

a b e e a f h j $\Rightarrow$

Prefix (p=4)	Score
a/A b/A e/A e/A	0.5
a/B b/A e/A e/A	0.2
a/C b/A e/A e/A	-1.5
a/A b/B e/A e/A	-1.6
a/B b/B e/A e/A	-1.7
a/C b/B e/A e/A	-2.0
. . .	

THE VITERBI ALGORITHM (CONTINUED)

- **Assume** representation $\mathbf{\Phi}$ contains features with max "span" $\leq L$ states. Can then discard lower scoring prefixes with identical $(L-1)$ states.

  E.g., If $L = 3$

Prefix (p=4)	Score
a/A b/A e/A e/A	0.5
a/B b/A e/A e/A	0.2
a/C b/A e/A e/A	-1.5
a/A b/B e/A e/A	-1.6
a/B b/B e/A e/A	-1.7
a/C b/B e/A e/A	-2.0
$\ldots$	

- Can find $\mathbf{F}(x) = \arg\max_{y \in \mathbf{GEN}(x)} \mathbf{\Phi}(y) \cdot \mathbf{W}$ in $O(nm^L)$ time

A FINAL TWIST: TRAINING USING BEAM SEARCH

- Even if representation $\mathbf{\Phi}$ contains features with potentially large "span", can **approximate** search for best hypothesis by keeping top N highest scoring prefixes at each stage

- Can also use Viterbi algorithm, beam-search for parsing

- **Main point:**
  if we can (approximately) find the highest scoring hypothesis under a representation $\mathbf{\Phi}$, we can train using the perceptron algorithm

- **Second point:**
  Convergence rate, generalization ability of the algorithm depends on **separability**, not on the size of $\mathbf{GEN}$

SUMMARY

- Margin–based algorithm for NLP: the Voted perceptron
- Three alternatives within the framework:
  - “Arbitrary” features
  - Kernels
  - Training weighted grammars
- Methods are applicable to many other tasks:
  - Language generation, Speech recognition
- Flexibility in representation:
  - Freedom to define global features
  - The tree kernel generalizes to other recursive structures
- Future questions:
  - How to do away with first pass altogether?

References

[Abney 1997] Abney

[ABR64] Aizerman, M., Braverman, E., and Rozonoer, L. (1964). Theoretical Foundations of the Potential Function Method in Pattern Recognition Learning. *Automation and Remote Control*, 25:821–837.

[Bod 98] Bod, R. (1998). *Beyond Grammar: An Experience-Based Theory of Language*. CSLI Publications/Cambridge University Press.

[CHA97] Charniak, E. (1997). Statistical techniques for natural language parsing. In *AI Magazine*, Vol. 18, No. 4.

[COL00] Collins, M. (2000). Discriminative Reranking for Natural Language Parsing. *Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco: Morgan Kaufmann.

[CD01] Collins, M. and Duffy, N. (2001). Parsing with a Single Neuron: Convolution Kernels for Natural Language Problems. Technical report UCSC-CRL-01-01, University of California at Santa Cruz.

[CV95] Cortes, C. and Vapnik, V. (1995). Support–Vector Networks. *Machine Learning*, 20(3):273–297.

[Work with Rob Schapire, Yoram Singer at COLT 2000]

[Friedman, Hastie, and Tibshirani 1998] Friedman.

[Lafferty 1999] lafferty.

[Della Pietra, Della Pietra, and Lafferty 1997] Lafferty

[Freund and Schapire 99] Freund, Y. and Schapire, R. (1999). Large Margin Classification using the Perceptron Algorithm. In *Machine Learning*, 37(3):277–296.

[FISS98] Freund, Y., Iyer, R.,Schapire, R.E., & Singer, Y. (1998). An efficient boosting algorithm for combining preferences. In *Machine Learning: Proceedings of the Fifteenth International Conference*. San Francisco: Morgan Kaufmann.

[HAU99] Haussler, D. (1999). *Convolution Kernels on Discrete Structures.* Technical report, University of Santa Cruz.

[Helmbold and Warmuth 95] Helmbold, D., and Warmuth, M. On Weak Learning.

[LCSW01] Lodhi, H., Cristianini, N., Shawe-Taylor, J., and Watkins, C. (2001). Text Classification using String Kernels. To appear in *Advances in Neural Information Processing Systems 13*, MIT Press.

[Johnson et. al 1999] Johnson, M., Geman, S., Canon, S., Chi, S., & Riezler, S. (1999). Estimators for stochastic 'unification-based" grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. San Francisco: Morgan Kaufmann.

[Johnson, Geman, Canon, Chi, and Riezler 1999] Johnson, M., Geman, S., Canon, S., Chi, S., & Riezler, S. (1999). Estimators for stochastic 'unification-based" grammars. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*. San Francisco: Morgan Kaufmann.

[MSM93] Marcus, M., Santorini, B., & Marcinkiewicz, M. (1993). Building a large annotated corpus of english: The Penn treebank. *Computational Linguistics, 19*, 313-330.

[Prince and Smolensky] Prince and Smolensky. Optimality theory.

[SSM99] Scholkopf, B., Smola, A.,and Muller, K.-R. (1999). Kernel principal component analysis. In B. Scholkopf, C. J. C. Burges, and A. J. Smola, editors, Advances in Kernel Methods – SV Learning, pages 327-352. MIT Press, Cambridge, MA.

[Walker et al. 2001] Walker, M., Rambow, O., and Rogati, M. (2001). SPoT: a trainable sentence planner. In *Proceedings of the 2nd Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2001)*.

[WAT00] Watkins, C. (2000). Dynamic alignment kernels. In A.J. Smola, P.L. Bartlett, B. Schlkopf, and D. Schuurmans, editors, Advances in Large Margin Classifiers, pages 39-50, MIT Press.