# Function Tagging

Don Blaheta

Committee:

Eugene Charniak (Advisor)

Mark Johnson (Cog. & Ling. Sciences)

Michael Collins (MIT Dept. of EE/CS)

6 August 2003

# Function Tagging
## Some types of markup

- Sentence segmentation
- Part of speech tagging
- Parse structure
- Phrase labelling
- Coreference annotation
- Named entity classification

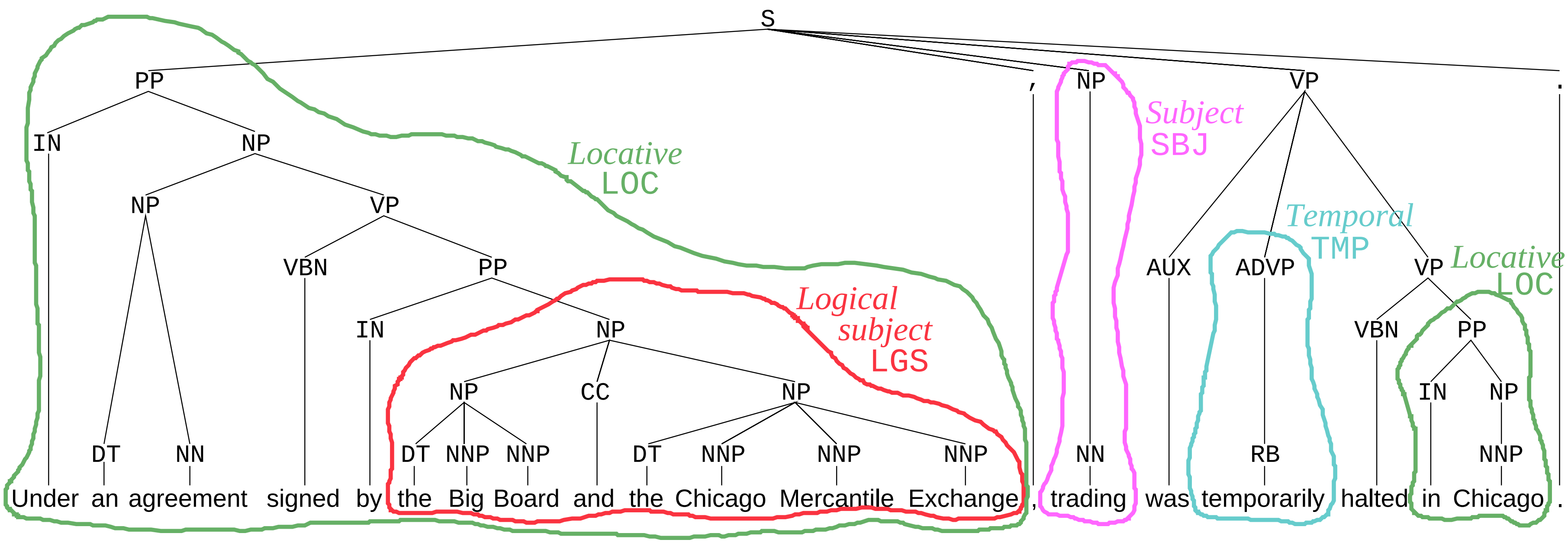
- *Function tagging*

# Function tags

A *function tag* is an annotation, chosen from a relatively small, discrete set of possible annotations, that is placed on a phrase to indicate that phrase's relationship to the rest of the utterance that contains it.

- subject vs. object
- topic
- theta role
- modifier (of time, of place, of . . . )

# Function tags: example



# Function tags: list

| | | Within Category | All Constits |
|---|---|---|---|
| **Grammatical** | | | **11.2%** |
| DTV | Dative | .5% | .1% |
| LGS | Logical subject | 3.0% | .3% |
| PRD | Predicate | 17.8% | 2.0% |
| PUT | 'Put' object | .3% | .0% |
| SBJ | Subject | 78.5% | 8.8% |
| VOC | Vocative | .0% | .0% |
| **Miscellaneous** | | | **.12%** |
| CLF | 'It'-cleft | 5.4% | .01% |
| HLN | Headline | 42.8% | .05% |
| TTL | Title | 51.8% | .06% |

| | | Within Category | All Constits |
|---|---|---|---|
| **Form/Function** | | | **7.8%** |
| ADV | Adverbial | 11.5% | .9% |
| BNF | Benefactive | .0% | .0% |
| DIR | Direction | 8.2% | .6% |
| EXT | Extent | 3.2% | .3% |
| LOC | Locative | 25.3% | 2.0% |
| MNR | Manner | 6.2% | .5% |
| NOM | Nominal | 6.8% | .5% |
| PRP | Purpose | 5.3% | .4% |
| TMP | Temporal | 33.4% | 2.6% |
| **Topicalisation** | | | **.5%** |
| TPC | Topicalised | 100.0% | .5% |

# Function tags: ambiguity

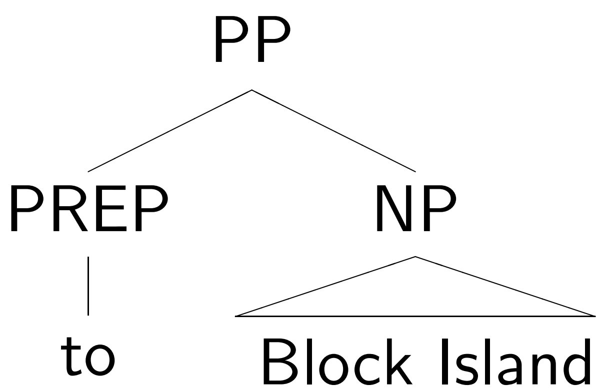
| | | |
|---|---|---|
| The volume was turned up | by eleven o'clock | . |
| | by John | |
| | by the DJ's table | |
| | by 30 decibels | |
| | by a twist of the knob | |

# Function tags: ambiguity

| | | |
|---|---|---|
| The volume was turned up | by eleven o'clock . | *Temporal* |
| | by John | *Log. Sbj.* |
| | by the DJ's table | *Locative* |
| | by 30 decibels | *Extent* |
| | by a twist of the knob | *Manner* |

# Function Tagging

## A mathematical reduction




$\Longrightarrow$ DIR

needs to be

$$\langle 0,0,0,0,0,1,0,0,0,1,0,0,\cdots\rangle \quad \Longrightarrow \quad 5$$

# Features

- Question whose answers come from predefined set
  - Of a person: gender, middle initial, favourite ivy league school
  - Of a class: professor, department

# Function Tagging

## Features

- Question whose answers come from predefined set
  - Of a person: gender, middle initial, favourite ivy league school
  - Of a class: professor, department
- Binary features

# Features

- Question whose answers come from predefined set
  - Of a person: gender, middle initial, favourite ivy league school
  - Of a class: professor, department
- Binary features

Favourite ivy league school? Brown

# Features

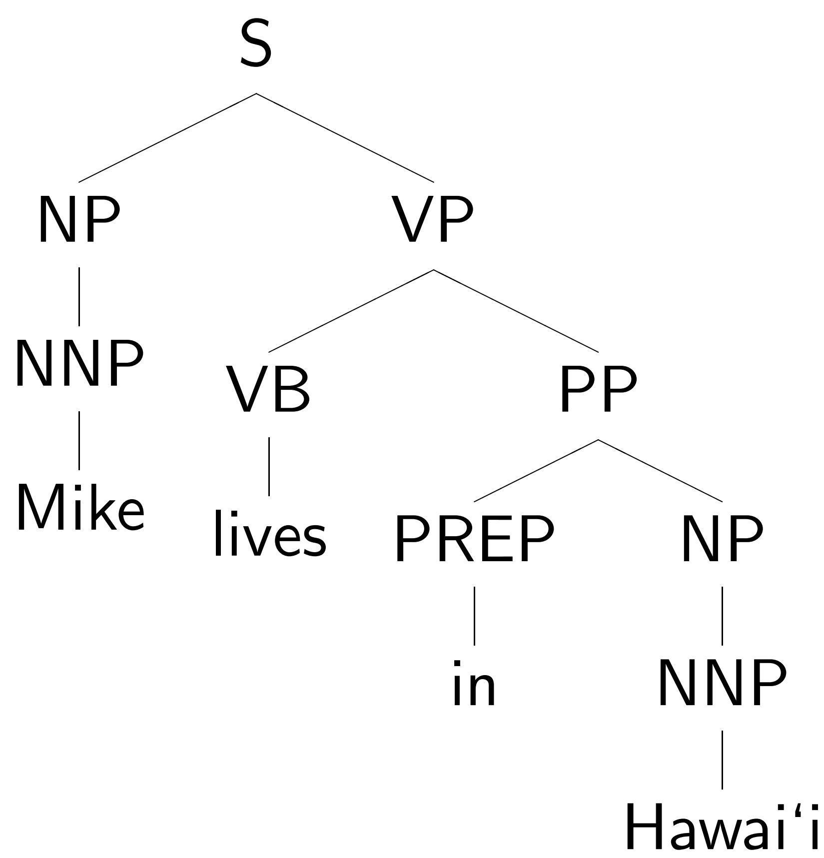
- Question whose answers come from predefined set
  - Of a person: gender, middle initial, favourite ivy league school
  - Of a class: professor, department
- Binary features

Favourite ivy league school? Brown

Fav. ivy is Dartmouth? No
Fav. ivy is Harvard? No
Fav. ivy is Brown? Yes
Fav. ivy is Cornell? No
$\vdots$

# Function Tagging

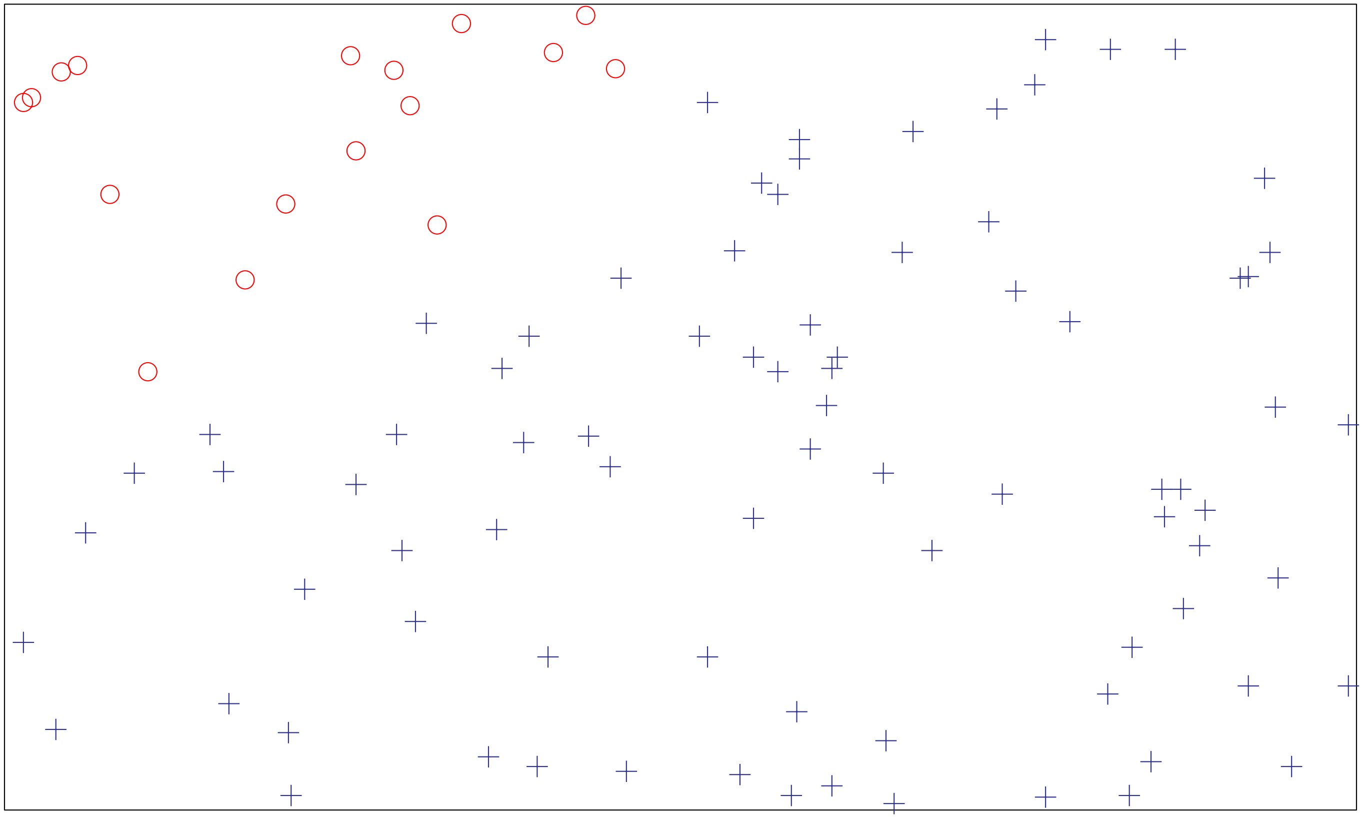
## Linguistic features




- label
- head
- head's POS
- parent's label
- sibling's label
- secondary head

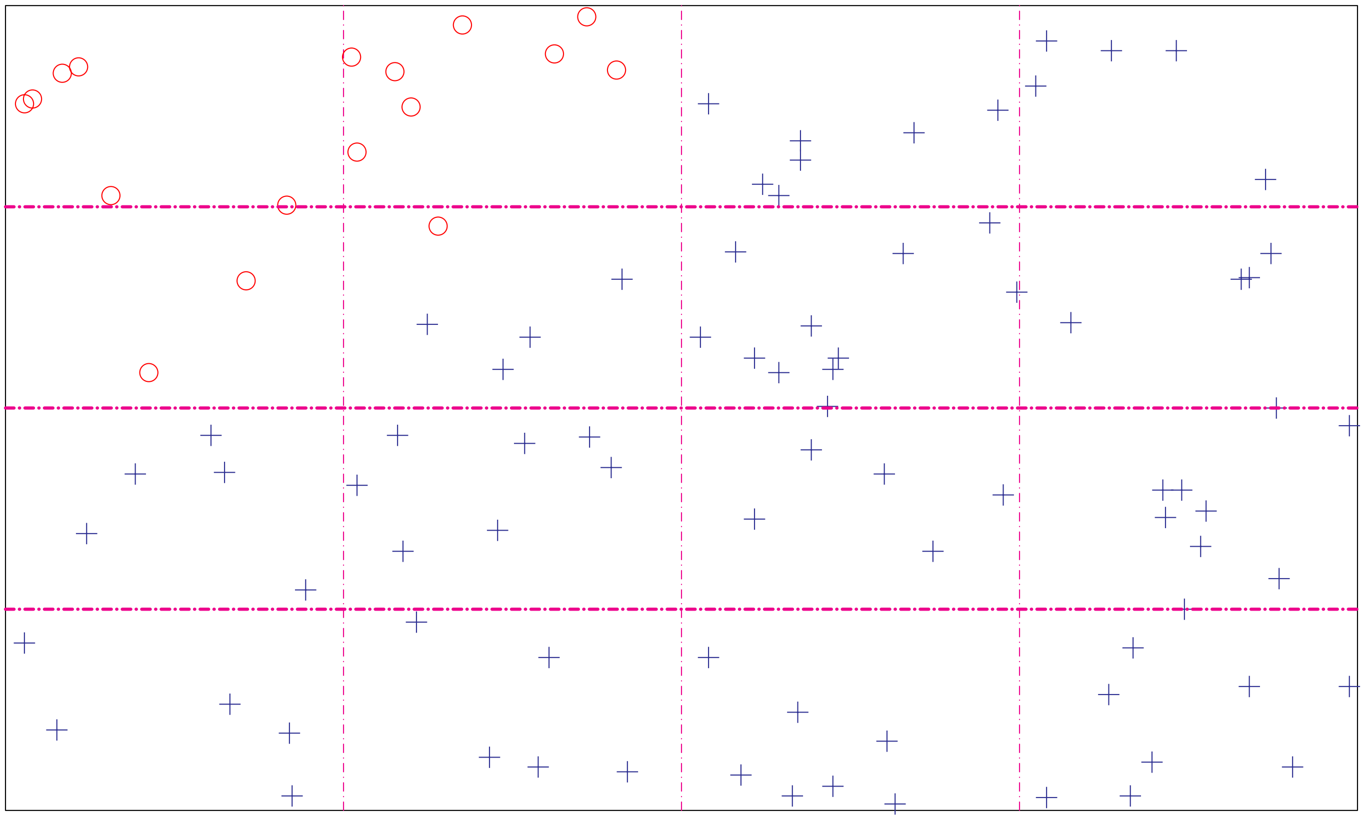
# Function Tagging

## A geometrical interpretation



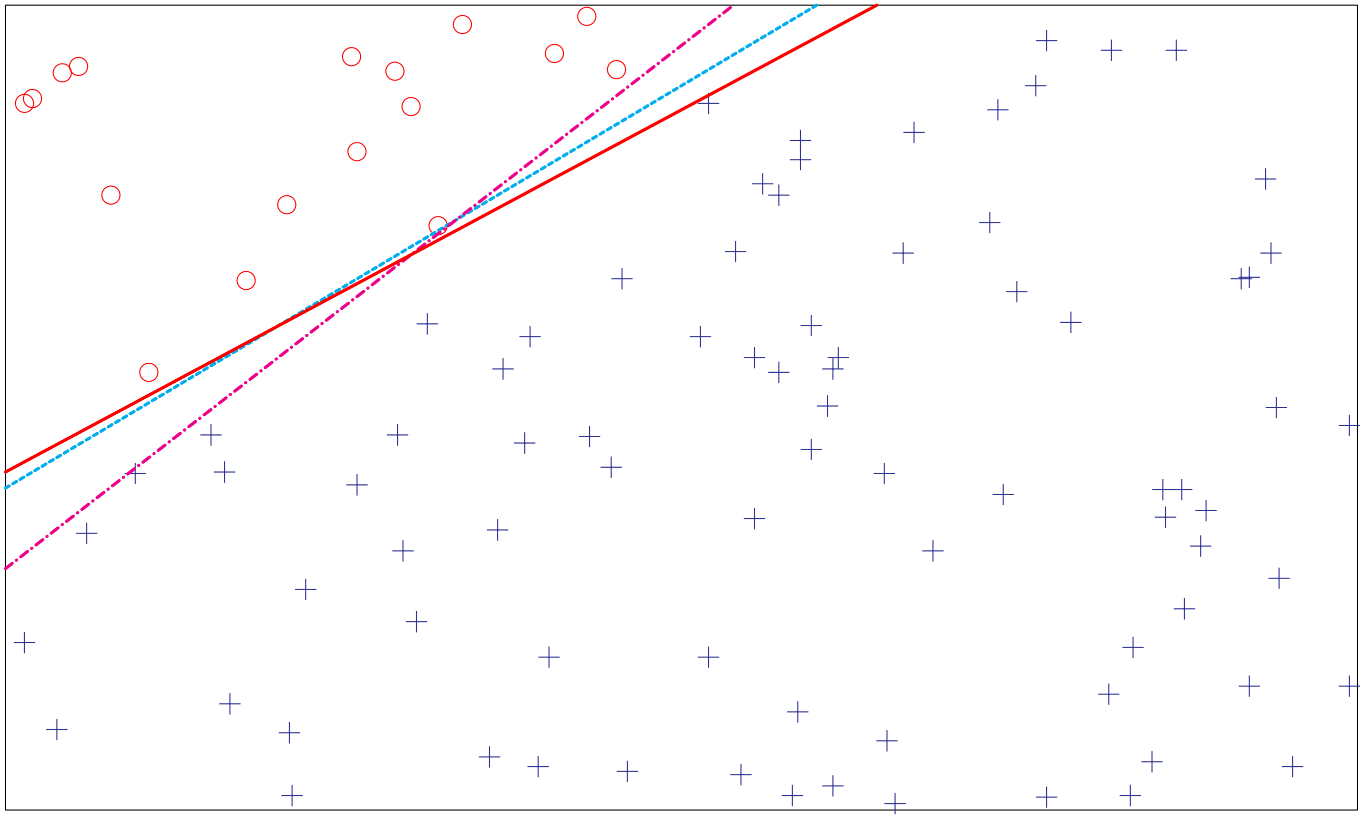
# Function Tagging

## Linear backoff, Decision tree



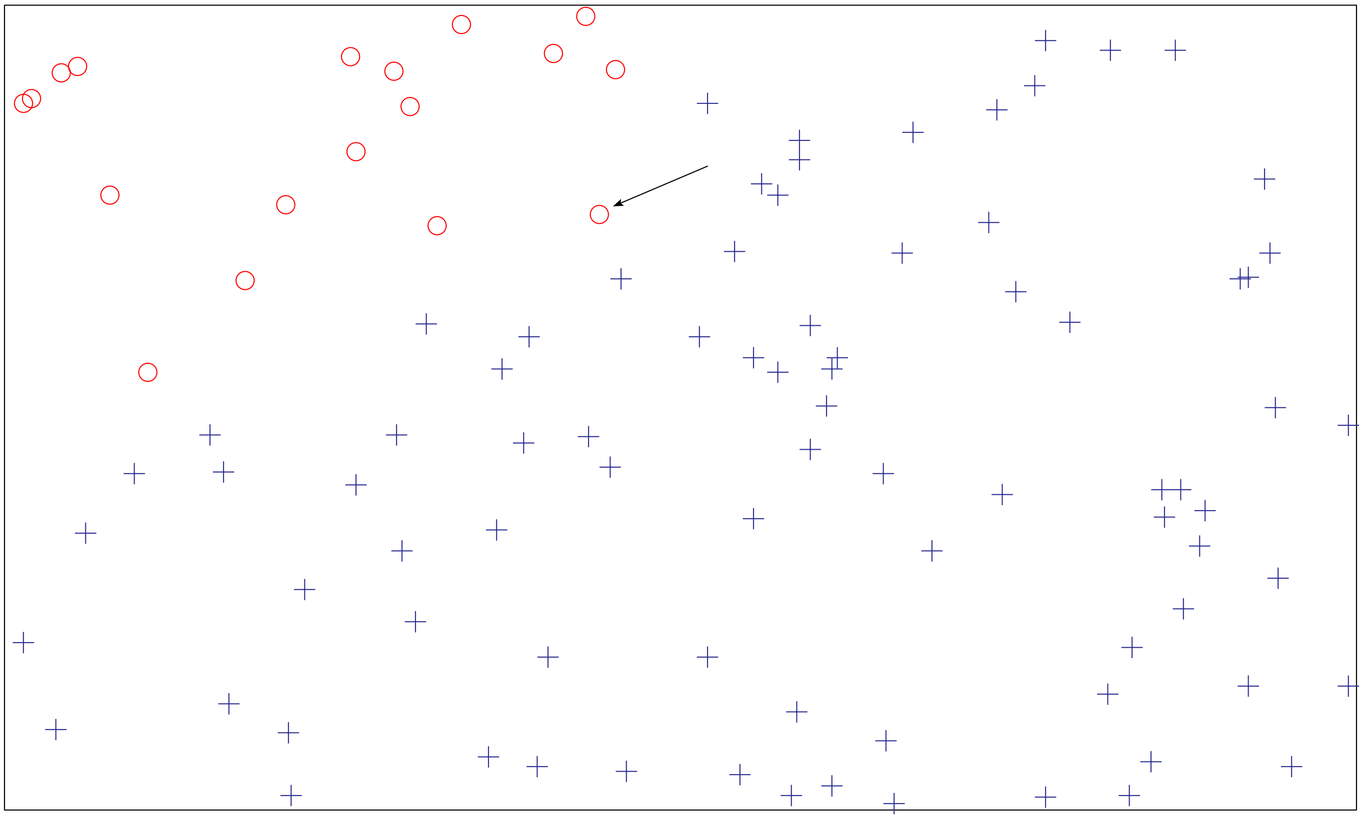
# Function Tagging

## Perceptrons



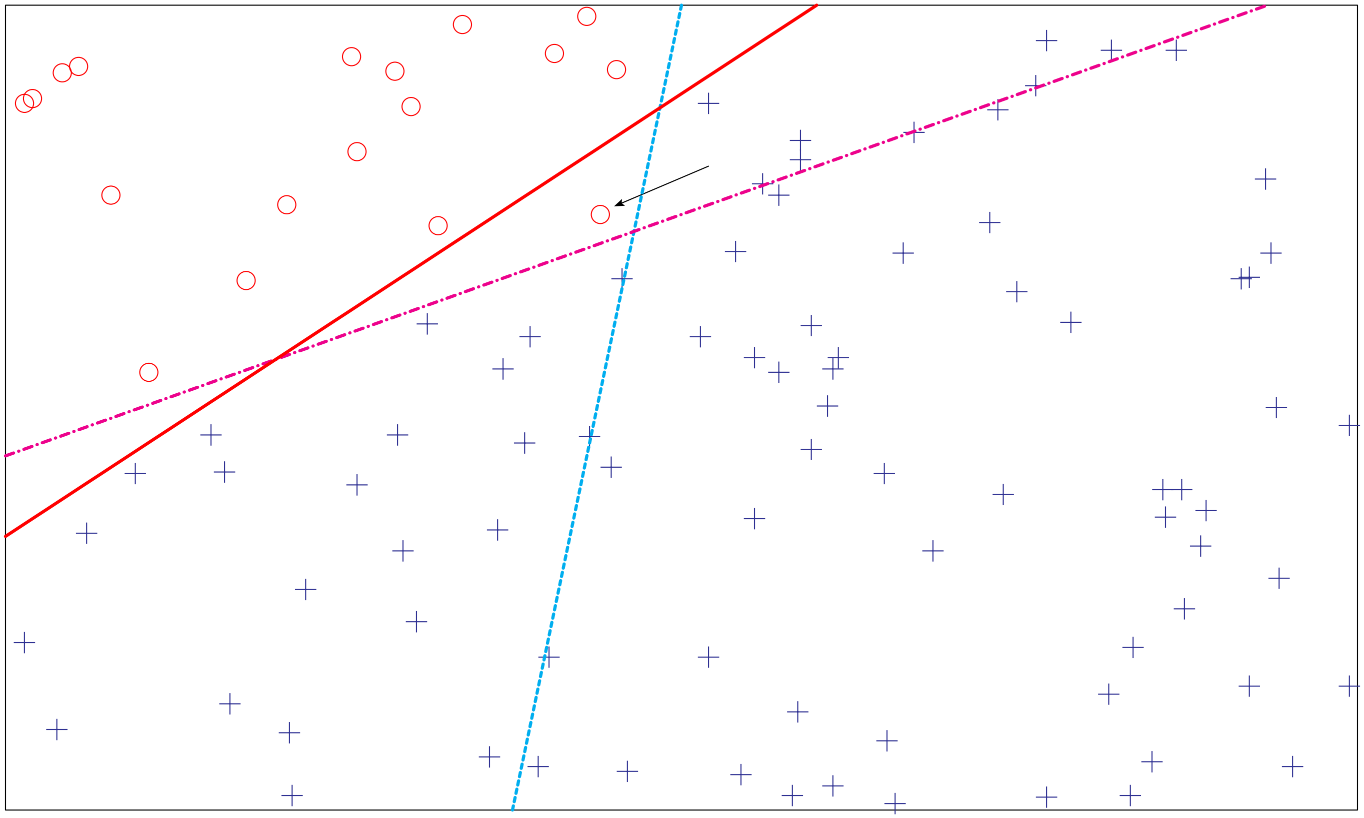
# Function Tagging

## Perceptrons



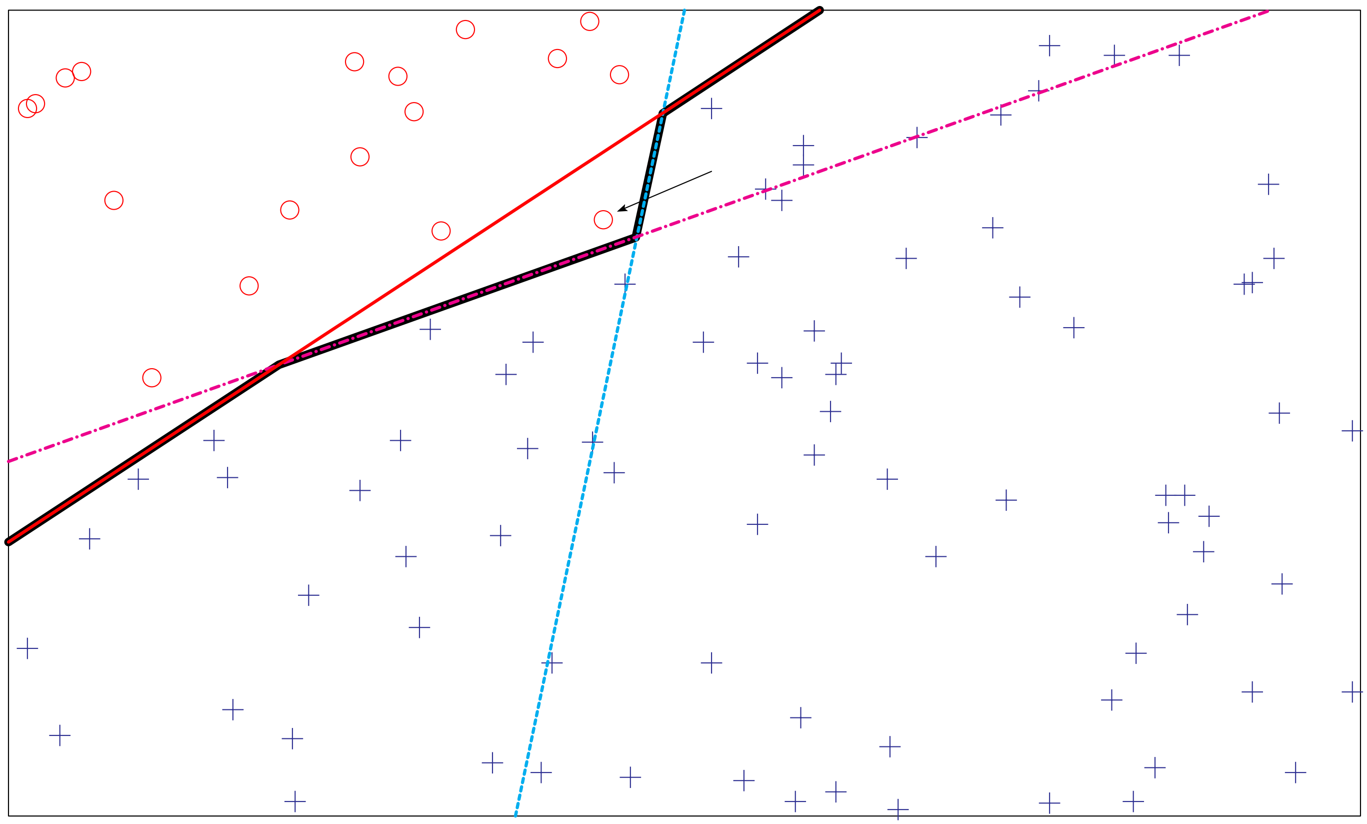
# Function Tagging

## Perceptrons: naïve



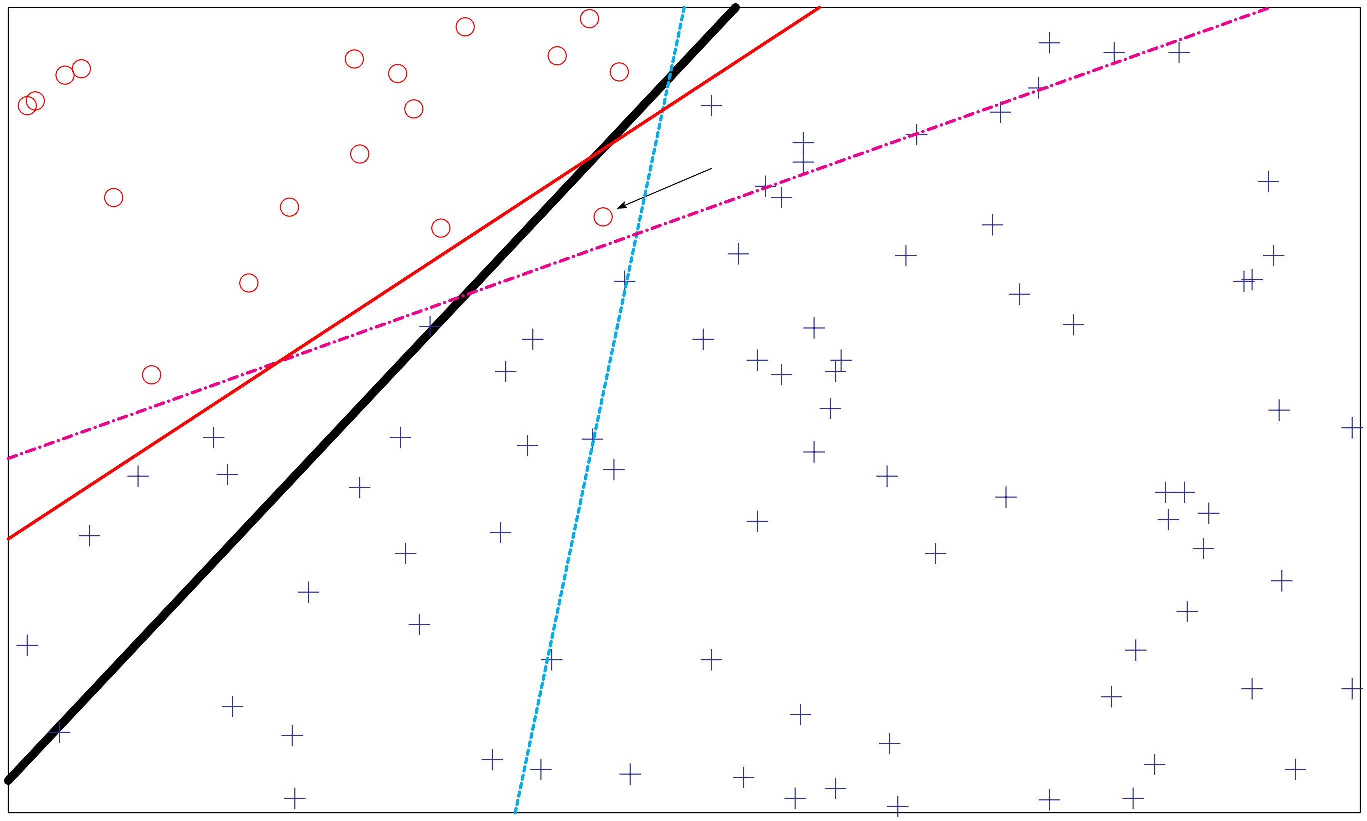
# Function Tagging

## Perceptrons: voted

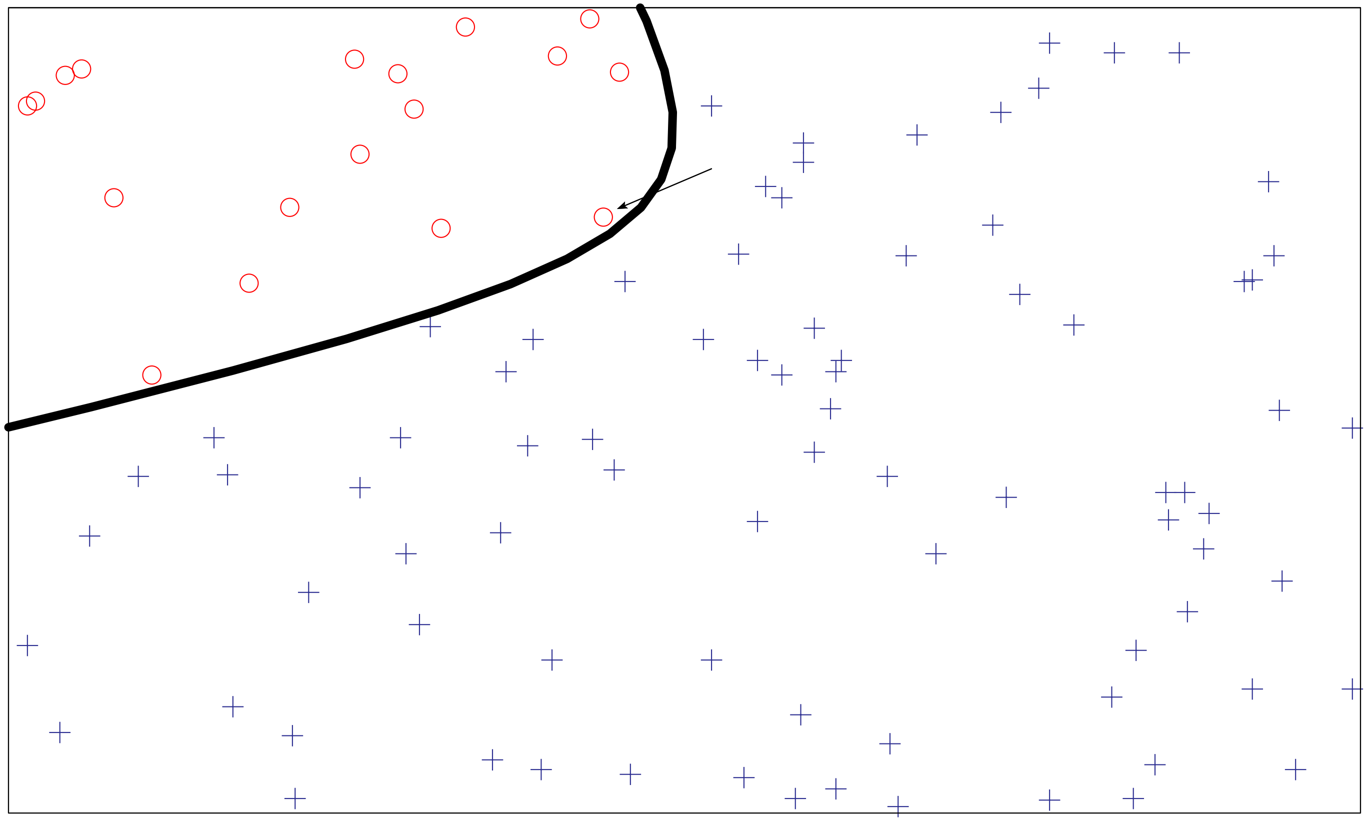


# Function Tagging

## Perceptrons: averaged

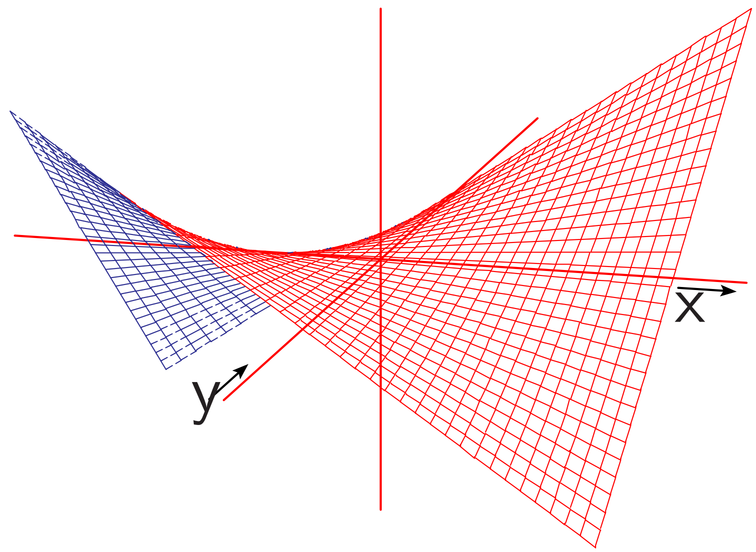


# Perceptrons: kernel-based

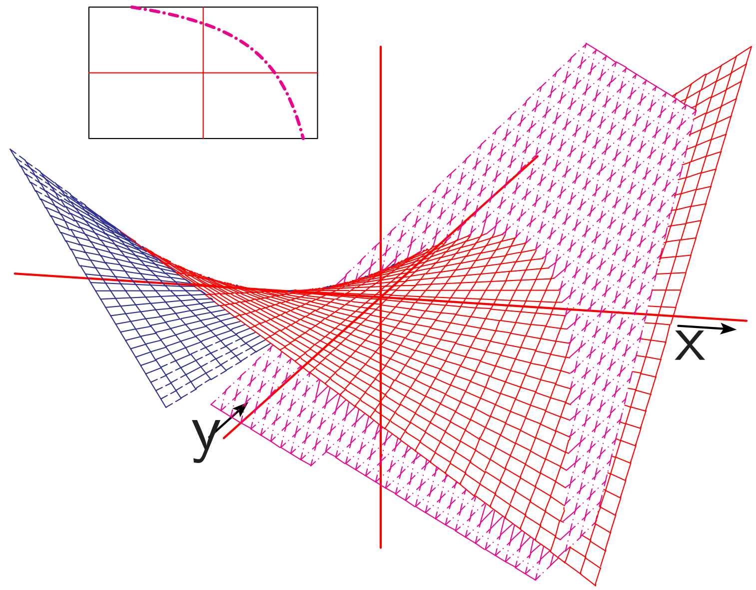


# Function Tagging

## Perceptrons: kernel-based



# Perceptrons: kernel-based

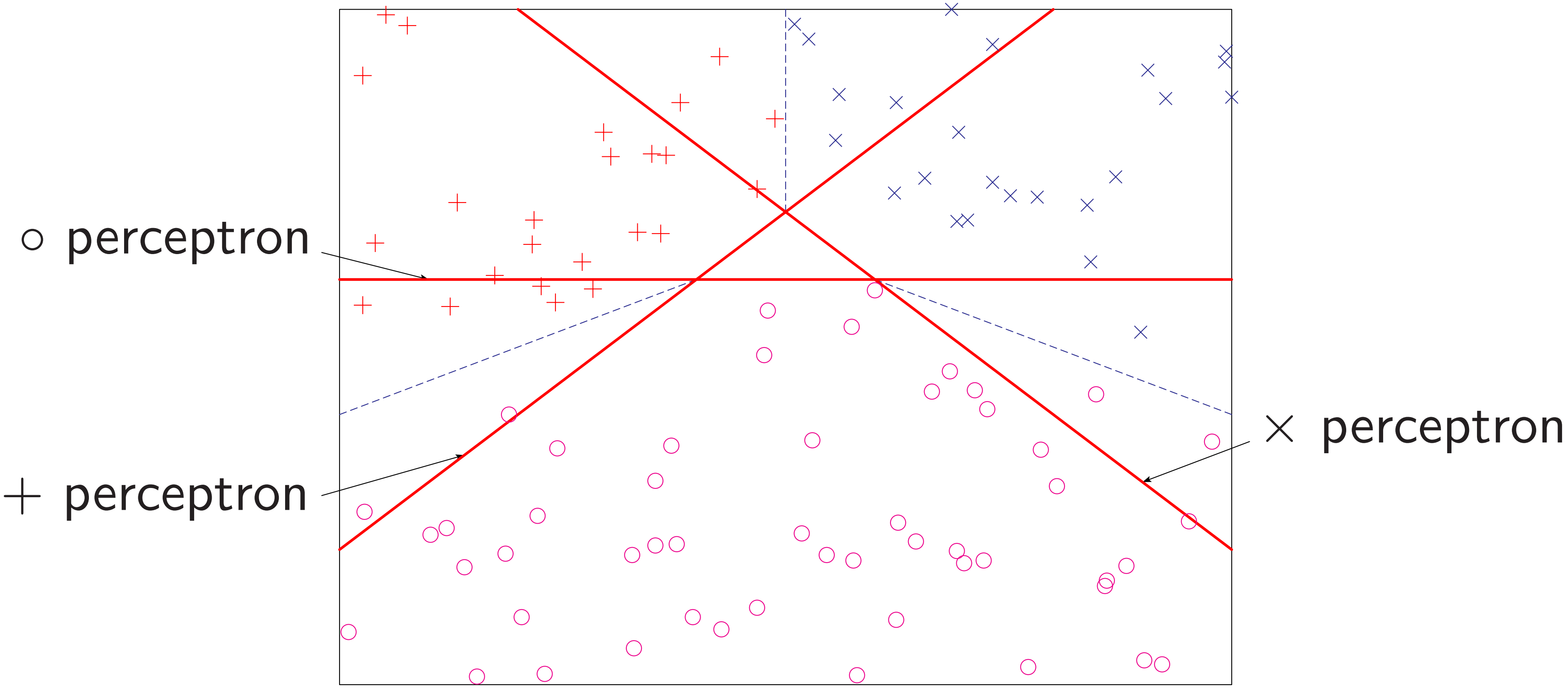


# Perceptrons: multi-valued

- $m$ “experts” (perceptrons)
- each expert $j$ knows only about tag $j$
- most confident expert applies his tag

# Function Tagging

## Perceptrons: multi-valued



# Perceptrons: training

For each training constituent $c_i$, whose correct tag is $f$

For each possible tag $j$

$score_j \leftarrow w_j \cdot c_i$

$a \leftarrow \text{argmax}_j\, score_j$

if $a \neq f$ (*guessed wrong*)

$w_a \leftarrow w_a - c_i$

$w_f \leftarrow w_f + c_i$

# Perceptrons: applying

For each testing constituent $c_i$,

For each possible tag $j$

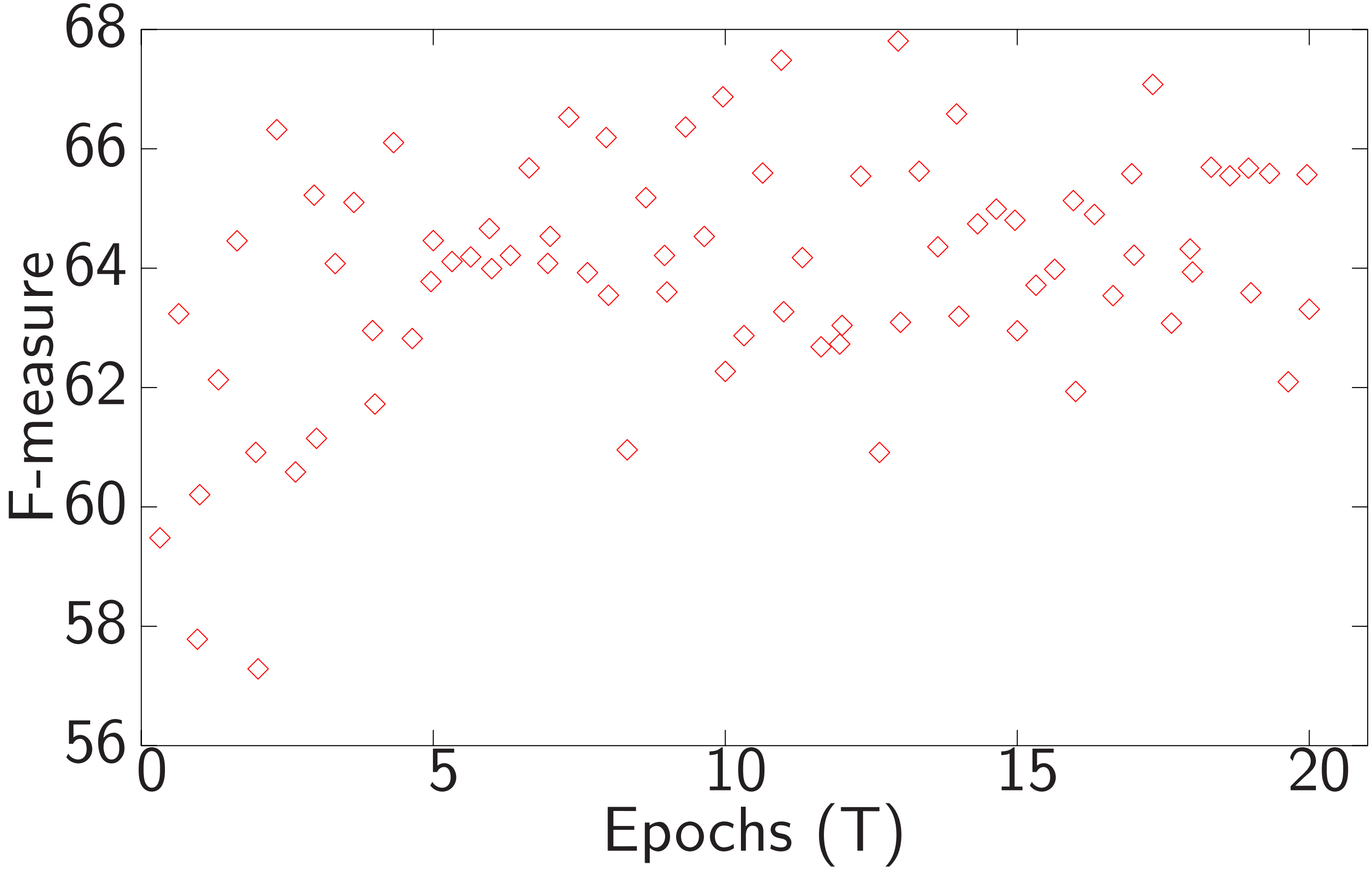
$score_j \leftarrow w_j \cdot c_i$

$a \leftarrow \text{argmax}_j \, score_j$

return tag $a$

# Function Tagging

## Perceptron performance



# Perceptron performance

| | Syntactic | Semantic |
|---|---|---|
| Naïve (average $5 \leq T \leq 20$) | 97.5 | 64.1 |

# Perceptron performance

| | Syntactic | Semantic |
|---|---|---|
| Naïve (average $5 \leq T \leq 20$) | 97.5 | 64.1 |
| Voted ($T = 1$) | 97.9 | 66.4 |

# Sparse voting

- Usual definition of voted perceptron:
  - Save all intermediate perceptrons
  - Calculate prediction according to each
  - Use most frequent prediction
- Each epoch $=$ 780K examples $\times$ 20 epochs $=$ 15.6M votes
- Only use 60 or so?

# Perceptron performance

| | Syntactic | Semantic |
|---|---|---|
| Naïve (average $5 \leq T \leq 20$) | 97.5 | 64.1 |
| Voted ($T = 1$) | 97.9 | 66.4 |
| Sparse voted ($5 \leq T \leq 20$) | 98.5 | 69.1 |

# Perceptron performance

| | Syntactic | Semantic |
|---|---|---|
| Naïve (average $5 \leq T \leq 20$) | 97.5 | 64.1 |
| Voted ($T = 1$) | 97.9 | 66.4 |
| Sparse voted ($5 \leq T \leq 20$) | 98.5 | 69.1 |
| Kernel ($T = 1; d = 2$) | 97.5 | 78.0 |
| Kernel voted ($T = 1; d = 2$) | 98.4 | 77.3 |

# Function Tagging

## Perceptron performance

| | Syntactic | Semantic | Time train | Time test |
|---|---|---|---|---|
| Naïve (average $5 \leq T \leq 20$) | 97.5 | 64.1 | 55m | 7s |
| Voted ($T = 1$) | 97.9 | 66.4 | 3m | 1h/13h |
| Sparse voted ($5 \leq T \leq 20$) | 98.5 | 69.1 | 55m | 7m |
| Kernel ($T = 1; d = 2$) | 97.5 | 78.0 | | |
| Kernel voted ($T = 1; d = 2$) | 98.4 | 77.3 | | |

- 27K non-terminal constituents; 1300 sentences; 33K words
- at 120wpm, 4.5 hours of text

# Function Tagging

## Perceptron performance

| | Syntactic | Semantic | Time: train | Time: test |
|---|---|---|---|---|
| Naïve (average $5 \leq T \leq 20$) | 97.5 | 64.1 | 55m | 7s |
| Voted ($T = 1$) | 97.9 | 66.4 | 3m | 1h/13h |
| Sparse voted ($5 \leq T \leq 20$) | 98.5 | 69.1 | 55m | 7m |
| Kernel ($T = 1; d = 2$) | 97.5 | 78.0 | 15h/10d | 1h/9h |
| Kernel voted ($T = 1; d = 2$) | 98.4 | 77.3 | | |

- 27K non-terminal constituents; 1300 sentences; 33K words
- at 120wpm, 4.5 hours of text

# Function Tagging

## Feature set performance

| | Syntactic | Semantic |
|---|---|---|
| self | 40.5 | 52.9 |
| self+parent's label | 90.8 | 61.2 |
| self+parent | 96.6 | 68.3 |
| self+sibs | 94.5 | 64.8 |
| self+parent+sibs | 97.9 | 69.9 |
| self+parent+sibs+gp (basic) | 98.6 | 68.7 |
| basic+sm/sy | 98.7 | 69.1 |
| basic+parent's sm | 98.5 | 69.3 |
| basic+twosib labels | 98.7 | 70.0 |
| basic+alt | 98.5 | 77.6 |
| basic+sm/sy+p's sm+2sib+alt (full) | 98.8 | 78.5 |
| full − lex | 95.7 | 49.2 |

# Final results

| **Syntactic tags** | Precision | Recall | F-measure |
|---|---|---|---|
| (Blaheta&Charniak, 2000) | 95.5% | 95.9% | 95.7% |
| Later feature trees | 96.5% | 95.3% | 95.9% |
| Sparse voted perceptron | 97.0% | 95.7% | 96.4% |

| **Semantic tags** | Precision | Recall | F-measure |
|---|---|---|---|
| (Blaheta&Charniak, 2000) | 80.4% | 77.6% | 79.0% |
| Later feature trees | 86.7% | 80.3% | 83.4% |
| Sparse voted perceptron | 88.7% | 79.4% | 83.8% |

# System comparison

| Feature trees | Perceptrons |
|---|---|
| Faster to train and run | Slower but comparable |

# System comparison

| Feature trees | Perceptrons |
|---|---|
| Faster to train and run | Slower but comparable |
| Uses for language modelling | No probability distribution |

# System comparison

| Feature trees | Perceptrons |
| --- | --- |
| Faster to train and run | Slower but comparable |
| Uses for language modelling | No probability distribution |
| Hard to add new features | New features: just add and retrain |

# System comparison

| Feature trees | Perceptrons |
|---|---|
| Faster to train and run | Slower but comparable |
| Uses for language modelling | No probability distribution |
| Hard to add new features | New features: just add and retrain |
| Complicated algorithm | Fast and easy to implement |

# System comparison

| Feature trees | Perceptrons |
|---|---|
| Faster to train and run | Slower but comparable |
| Uses for language modelling | No probability distribution |
| Hard to add new features | New features: just add and retrain |
| Complicated algorithm | Fast and easy to implement |
| Fairly accurate | Slightly more accurate |

# Contributions

- Tagger for semantic modifiers
- More accurate tagger for syntactic modifiers
- Comparison of several systems on function tagging task
- New features
- Analysis of important features
- Sparse voted perceptron, counting votes for $T > 5$ only

# Future work

- Re-try averaged perceptron
- Cluster/backoff features
- German NEGRA corpus—syntactic; Penn-style
- Czech PDT corpus—syntactic and semantic; different linguistic model
- Applications: Question answering, machine translation

# Thanks

- Any questions?

# Related work: Collins 1997

- Parsing can be improved with complement/adjunct knowledge
- Function tags are used to indicate this
  - e.g. SBJ is complement, TMP is adjunct
- Results reported only on parser quality

# Related work: Gildea and Jurafsky 2000

- FrameNet corpus project
- Composed primarily of “frames” of discourse, e.g. conversation
- Phrases tagged as “frame elements”, e.g. TOPIC, MEDIUM
- Every frame has different frame elements
- Both harder and easier; difficult to compare

# Related work: Gildea and Jurafsky 2000

- FrameNet corpus
  - Domain: communication (cognition, motion)
  - Frame: conversation (statement, judgement)
  - Words: argue, debate, discussion, tiff
  - Frame elements: PROTAGONIST, TOPIC, MEDIUM

- Probabilistic, with lattice backoff model

- Given a sentence with marked frame elements, label them: 81.2%

- Given a sentence, mark frame elements: 66% (+ 15% partial)

# Related work: Brants, Skut, and Krenn 1997

- German-language treebank from POS-tagged newspaper text
- *Every* item has "function label" e.g. SB, HD
- Order-2 Markov model, one per parent label type

| **Brants, Skut, and Krenn** | |
|---|---|
| PP children | 97.9% |
| S children | 89.1% |
| Overall accuracy | 94.2% |

| **Blaheta** | |
|---|---|
| No-null precision | 96.5% |
| No-null recall | 95.3% |
| No-null F-measure | 95.9% |
| With-null accuracy | 99.0% |

# Feature trees

- In the 'chain', each 'link' expresses a dependency relationship. What if some terms are independent?

- Each independence assumption causes a fork in the chain, yielding a feature *tree*.

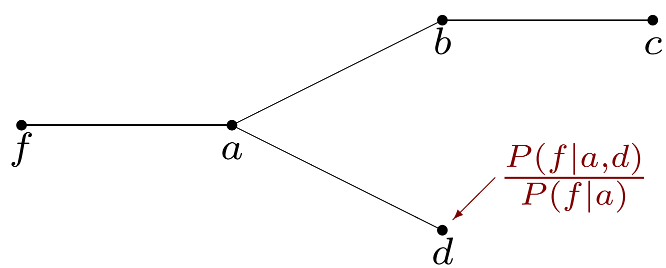
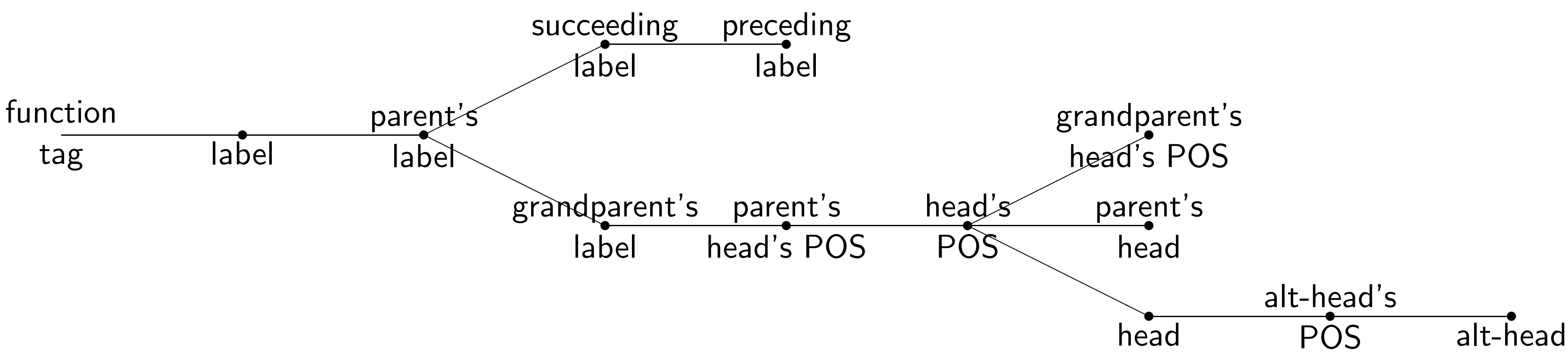



Figure 1: A feature tree: $d$ is independent of $b$ and $c$

# Function Tagging

## A feature tree



# Feature chains, technical

If a feature $f$ can be guessed from features $f_1, \ldots, f_n$, we usually estimate its probability as

$$P(f|f_1, f_2, \ldots, f_n) \approx \hat{P}(f|f_1, f_2, \ldots, f_j), \quad j \leq n\ .$$

This is equivalent to

$$P(f|f_1, f_2, \ldots, f_n) \approx \hat{P}(f)\,\frac{\hat{P}(f|f_1)}{\hat{P}(f)}\,\frac{\hat{P}(f|f_1, f_2)}{\hat{P}(f|f_1)} \cdots \frac{\hat{P}(f|f_1, f_2, \ldots, f_j)}{\hat{P}(f|f_1, f_2, \ldots, f_{j-1})}$$

or

$$P(f|f_1, f_2, \ldots, f_n) \approx \prod_{i=0}^{j} \frac{\hat{P}(f|f_1, \ldots, f_{i-1}, f_i)}{\hat{P}(f|f_1, \ldots, f_{i-1})}\ .$$

# Function Tagging

## Feature trees, technical

À propos Figure 1, if $d$ were still dependent on $c$ (and we had complete data), the probability estimate would be

$$P(f|a,b,c,d) \approx \hat{P}(f)\,\frac{\hat{P}(f|a)}{\hat{P}(f)}\,\frac{\hat{P}(f|a,b)}{\hat{P}(f|a)}\,\frac{\hat{P}(f|a,b,c)}{\hat{P}(f|a,b)}\,\frac{\hat{P}(f|a,b,c,d)}{\hat{P}(f|a,b,c)}\;.$$

Noting $d$'s independence from $b$ and $c$, this becomes

$$P(f|a,b,c,d) \approx \hat{P}(f)\,\frac{\hat{P}(f|a)}{\hat{P}(f)}\,\frac{\hat{P}(f|a,b)}{\hat{P}(f|a)}\,\frac{\hat{P}(f|a,b,c)}{\hat{P}(f|a,b)}\,\frac{\hat{P}(f|a,d)}{\hat{P}(f|a)}\;,$$

which cancels to

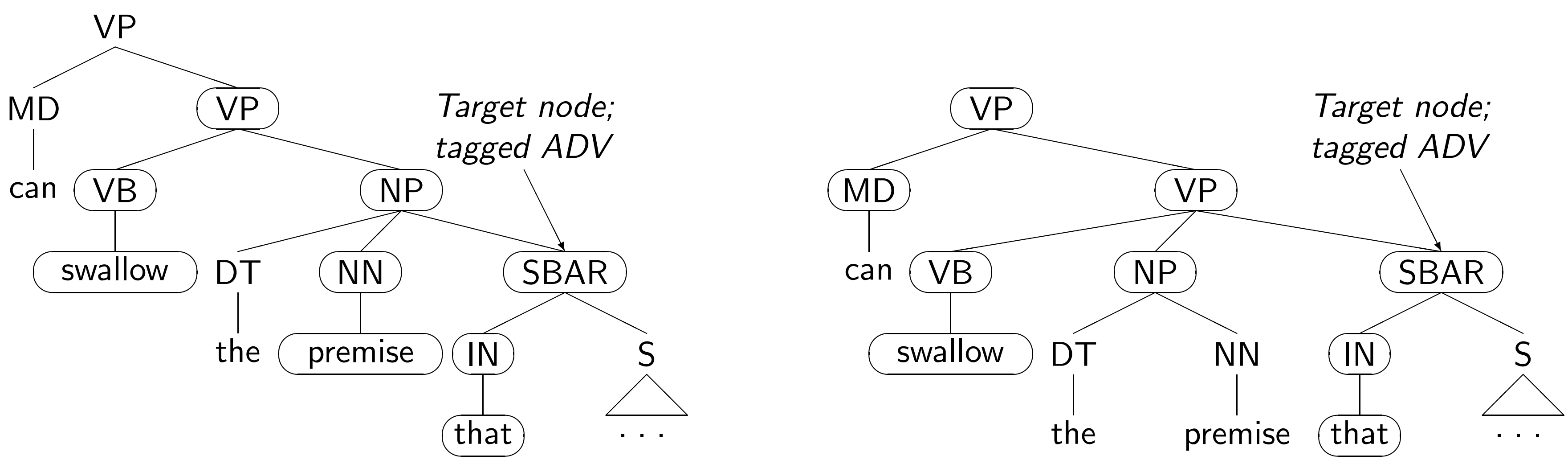
$$P(f|a,b,c,d) \approx \frac{P(f|a,b,c)P(f|a,d)}{P(f|a)}\;.$$

# Function Tagging

## Error analysis

| | |
|---|---|
| Parser error | 20% |
| Type A, B error | 18% |
| Type C error | 13% |
| Dubious | 6% |
| Algorithm error | 44% |

# Outside sources of error I: Parser error



# Outside sources of error II: Treebank error

- Type A: Detectable
  - LGS "attaches to the NP object of *by* and not to the PP node."
  - "President Bush has been weakened *by the Panama fiasco*."

- Type B: Fixable
  - LOC can be metaphorical, but not idiomatic
  - "think *about national service*" shouldn't be LOC

- Type C: Inconsistent
  - MNR indicates the manner in which an action is performed
  - "*impatiently*", "*suddenly*", "*significantly*", "*clearly*"