Phrase Recognition by Filtering and Ranking with Perceptrons

Xavier Carreras and **Lluís Màrquez**

TALP Research Center

Software Department

Techical University of Catalonia

RANLP 2003

Borovets, September 10, 2003

Outline

- Introduction
- Phrase Recognition Model
- Global Learning Algorithm
- Experimental Evaluation
- Conclusions and Current Work

Phrase Recognition

A very general definition of phrase:

A sequence of contiguous lexical items that forms a unit of a certain type (e.g., named entities, syntactic chunks, clauses, etc.)

Phrase Recognition Problems$_{(1)}$

Chunking

[NP He] [VP reckons] [NP the current account deficit] [VP will narrow] [PP to] [NP only 1.8 billion] [PP in] [NP September] .

Named Entity Recognition

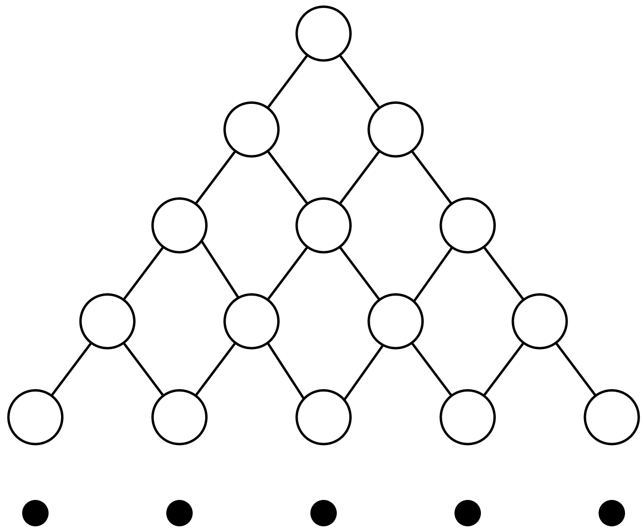
[PER Wolff] , currently a journalist in [LOC Argentina] , played with [PER Del Bosque] in the final years of the seventies in [ORG Real Madrid] .

Phrase Recognition Problems$_{(2)}$

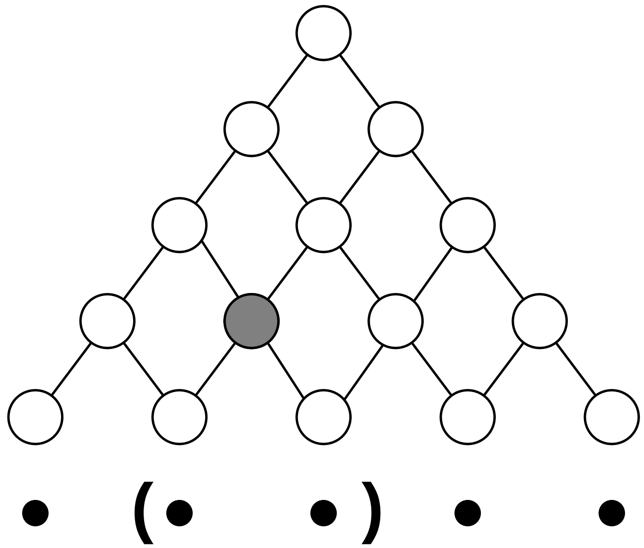
Clausing

(S The deregulation of railroads and trucking companies (SBAR that (S began in 1980)) enabled (S shippers to bargain for transportation) .)

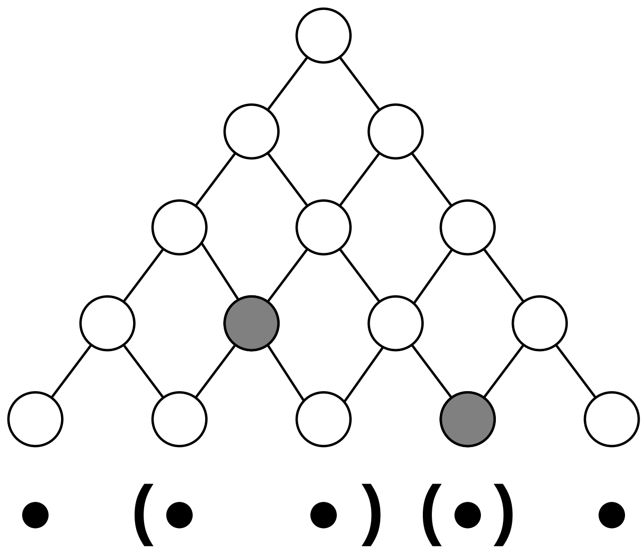
Phrase Recognition



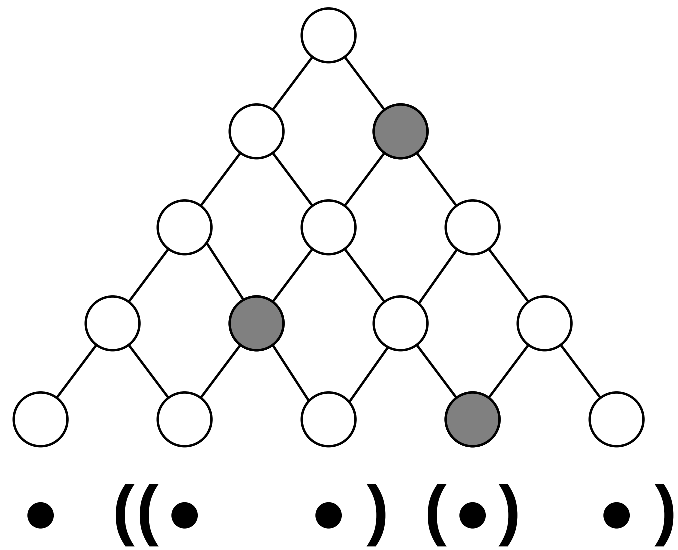
Phrase Recognition



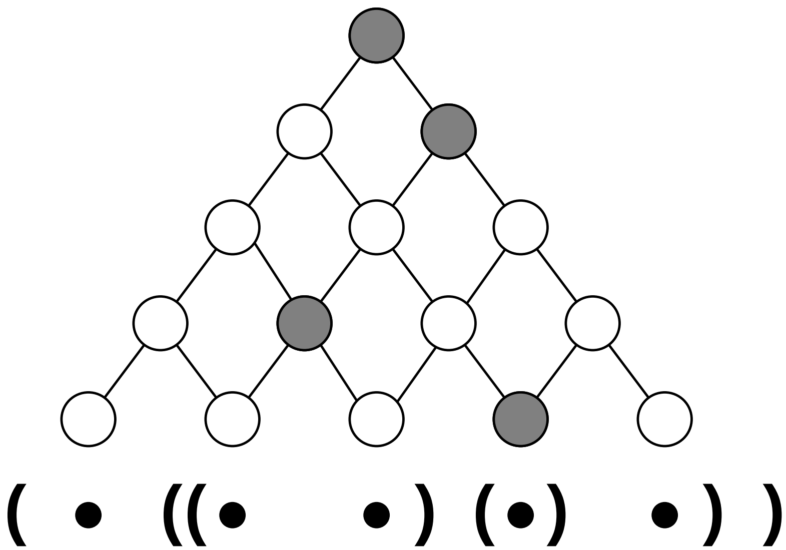
Phrase Recognition



Phrase Recognition



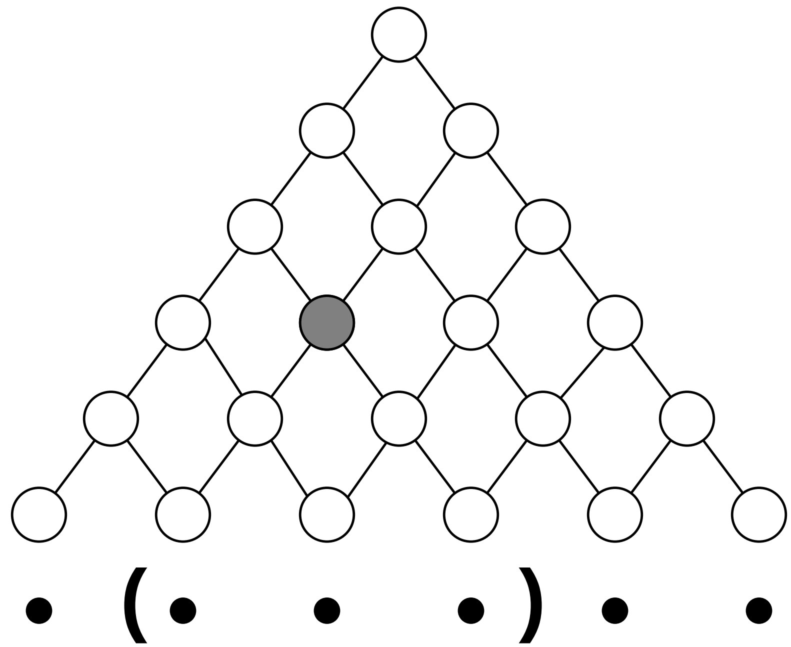
Phrase Recognition



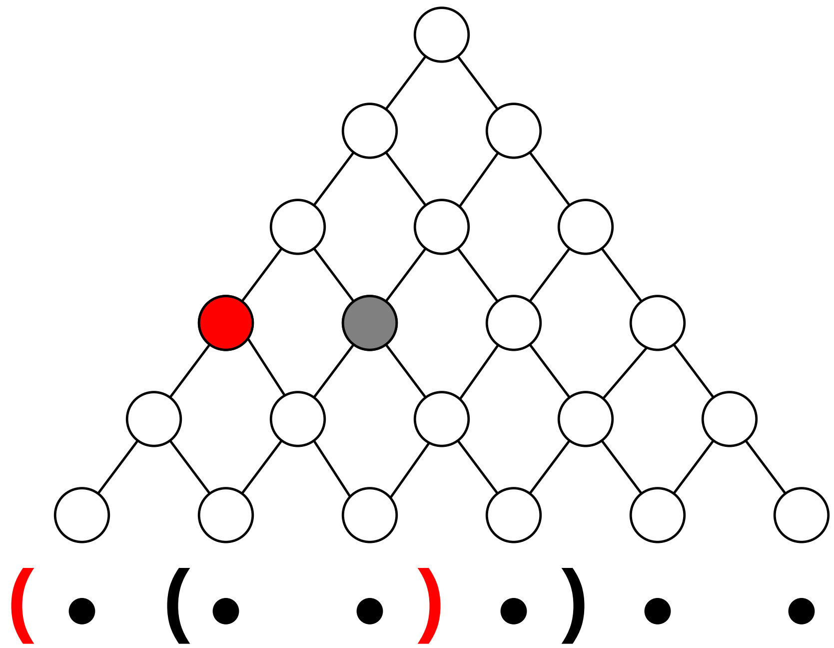
A solution is a coherent set of (embedded) phrases

$$x = x_0, x_1, x_2, x_3, x_4$$
$$y = \{(1, 2), (3, 3), (1, 4), (0, 4)\}$$

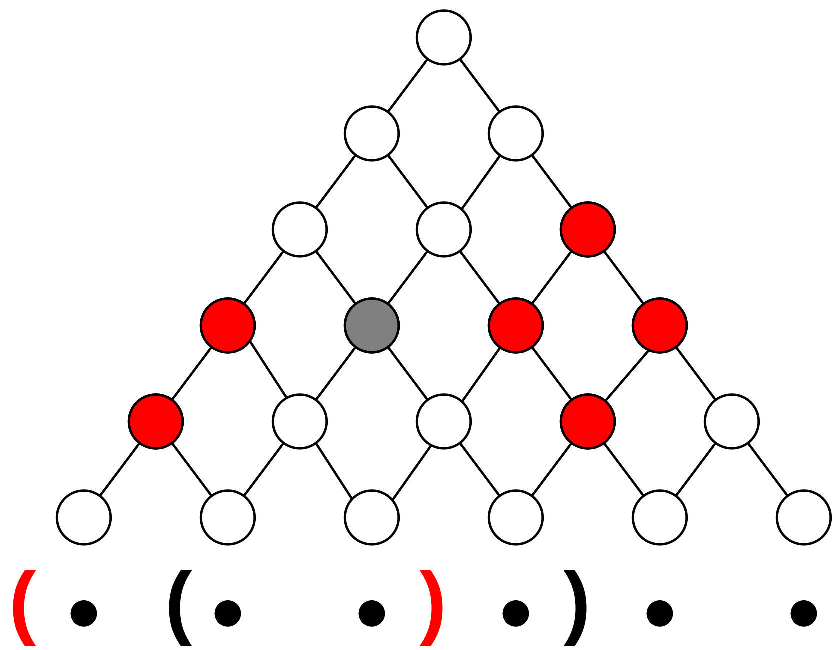
Phrase Overlapping



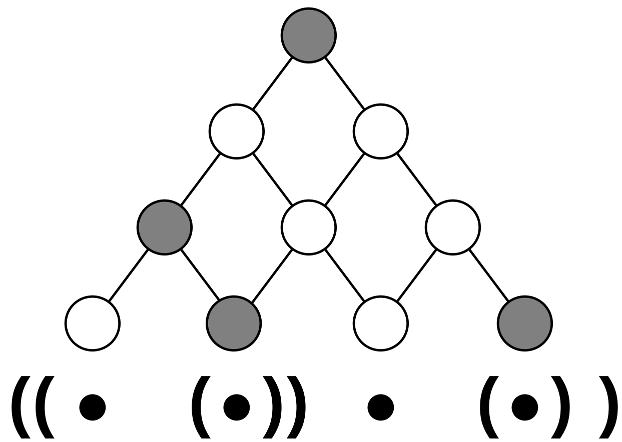
Phrase Overlapping



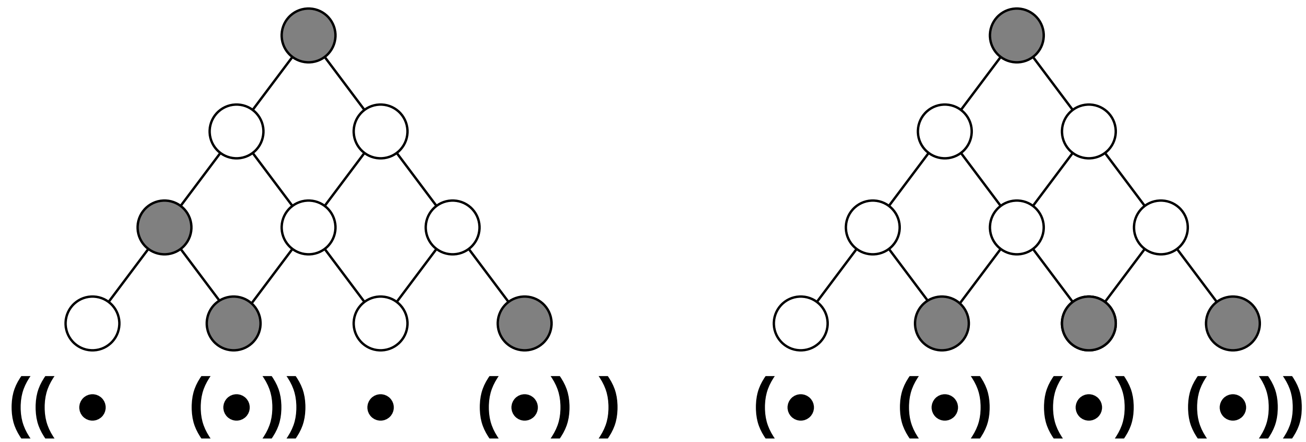
Phrase Overlapping



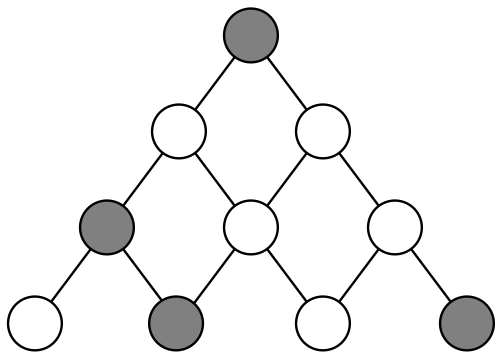
Some Solution Candidates



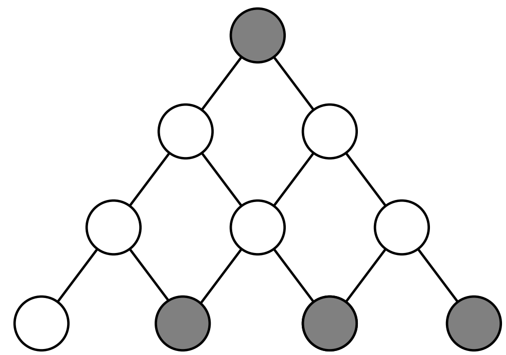
Some Solution Candidates



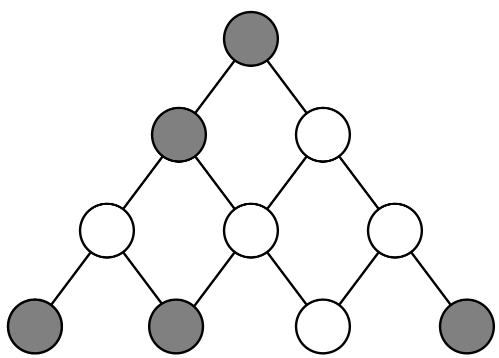
Some Solution Candidates



((• (•)) • (•))

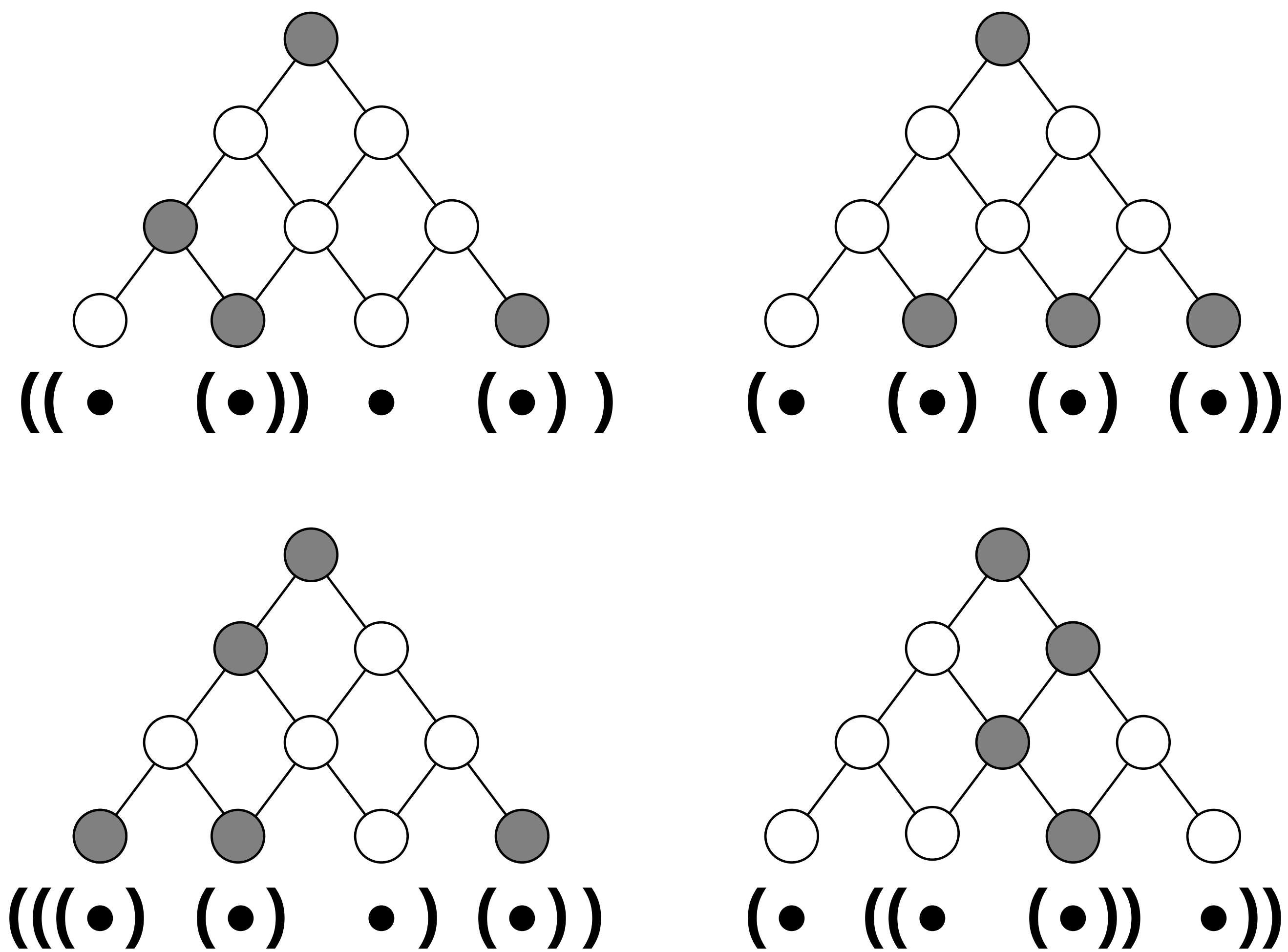


(• (•) (•) (•))



(((•) (•) •) (•))

Some Solution Candidates



Framework

- General algorithm for phrase recognition
  - ⋆ Machine Learning on local decisions/contexts
    - ∗ 1st layer: filtering at word level
    - ∗ 2nd layer: ranking at phrase level
  - ⋆ Inference Process to obtain the global solution

Framework

- General algorithm for phrase recognition
  - ⋆ Machine Learning on local decisions/contexts
    - ∗ 1st layer: filtering at word level
    - ∗ 2nd layer: ranking at phrase level
  - ⋆ Inference Process to obtain the global solution
- Usually, learning components are trained independently. In this work a global training strategy is proposed

Outline

- Introduction
- Phrase Recognition Model
- Global Learning Algorithm
- Experimental Evaluation
- Conclusions and Current Work

Phrase Score

We learn to score phrases. $\forall k \in \mathcal{K}$:

$$\mathrm{score}_k(s, e) \longrightarrow \mathbb{R}$$

Given the score of (s, e):

- The sign tells whether (s, e) is a k-phrase or not.
- The magnitude indicates the confidence of the decision.

Phrase Recognition Model

$\mathcal{Y}$: solution space, i.e. set of all coherent phrase sets.

$$\mathrm{PhRec}(x) = \arg\max_{y \in \mathcal{Y}} \sum_{(s,e)_k \in y} \mathrm{score}_k(s, e)$$

Phrase Recognition Model

$\mathcal{Y}$: solution space, i.e. set of all coherent phrase sets.

$$\mathrm{PhRec}(x) = \arg\max_{y \in \mathcal{Y}} \sum_{(s,e)_k \in y} \mathrm{score}_k(s,e)$$

- Sequential case: $O(n^2)$ Dynamic Prog. search
- Hierarchical case: $O(n^3)$ Dynamic Prog. search

Phrase Recognition Model: Start-End Candidates + Phrase Scoring

$\mathcal{Y}$: solution space, i.e. set of all coherent phrase sets.

$\mathcal{Y}_{SE}$: practical solution space, filtered at word level.

$$\mathrm{PhRec}(x) = \arg\max_{y \in \mathcal{Y}_{SE}} \sum_{(s,e)_k \in y} \mathrm{score}_k(s,e)$$

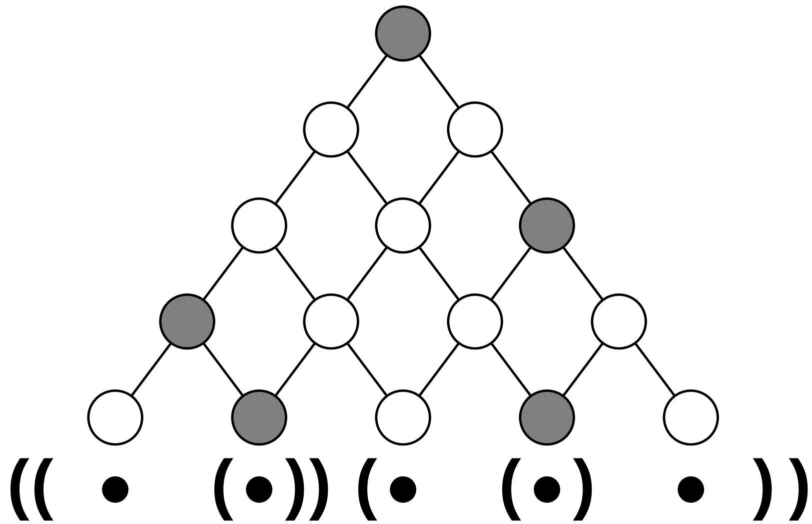
Phrase Recognition Model: Start-End Candidates + Phrase Scoring

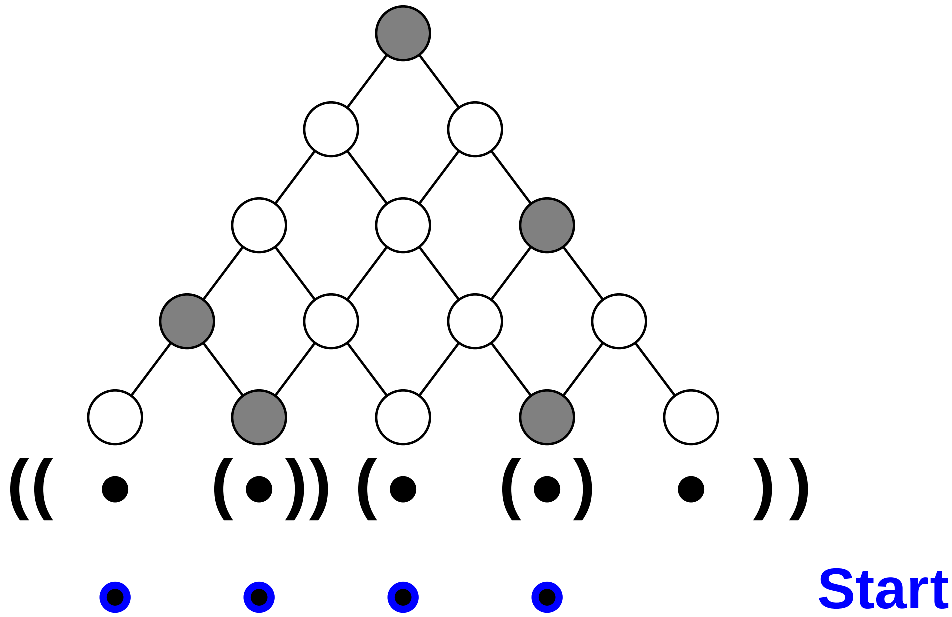
$\mathcal{Y}$: solution space, i.e. set of all coherent phrase sets.
$\mathcal{Y}_{SE}$: practical solution space, filtered at word level.

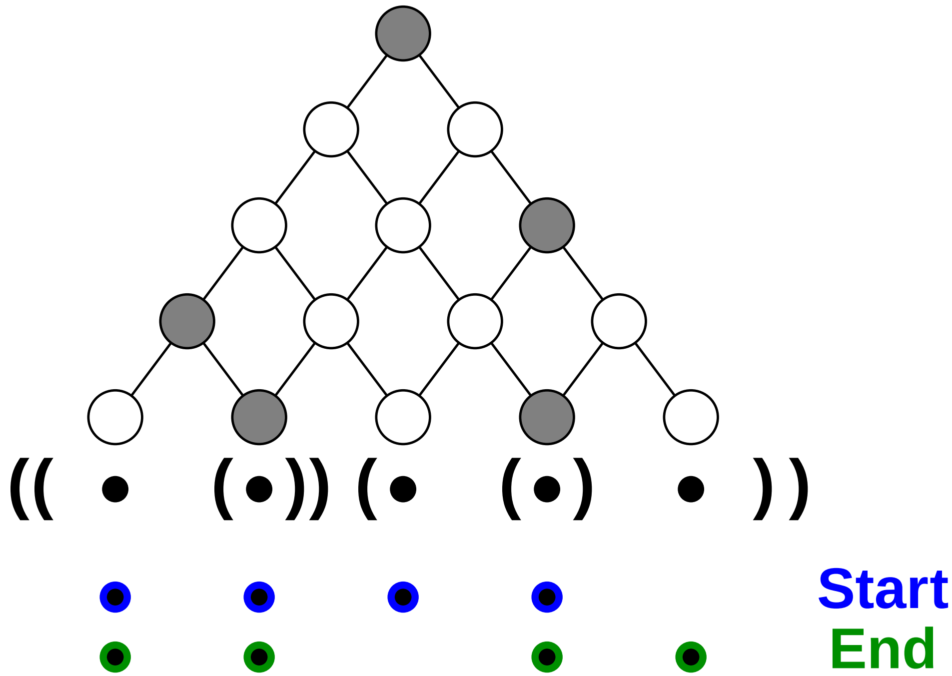
$$\mathrm{PhRec}(x) = \arg\max_{y \in \mathcal{Y}_{SE}} \sum_{(s,e)_k \in y} \mathrm{score}_k(s, e)$$

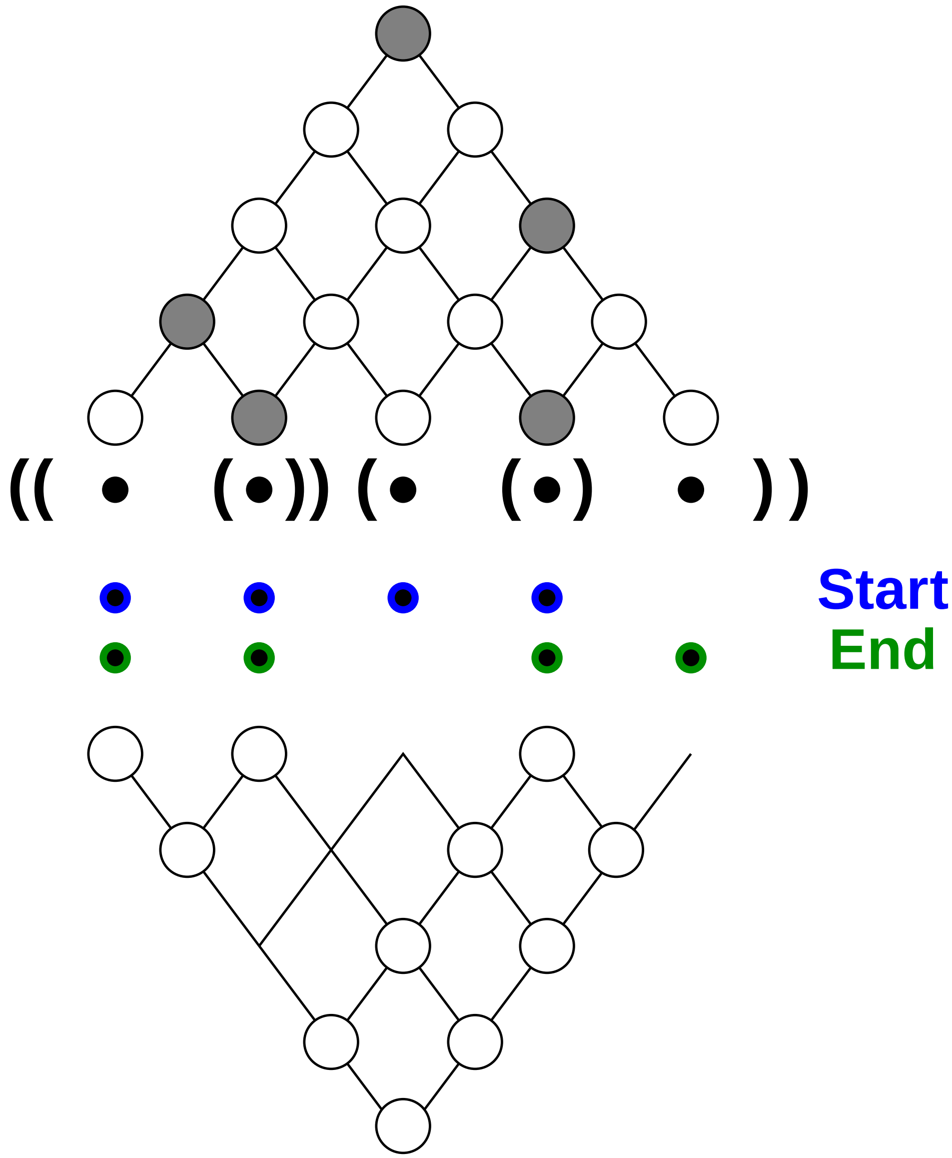
start and end binary classifiers perform filtering

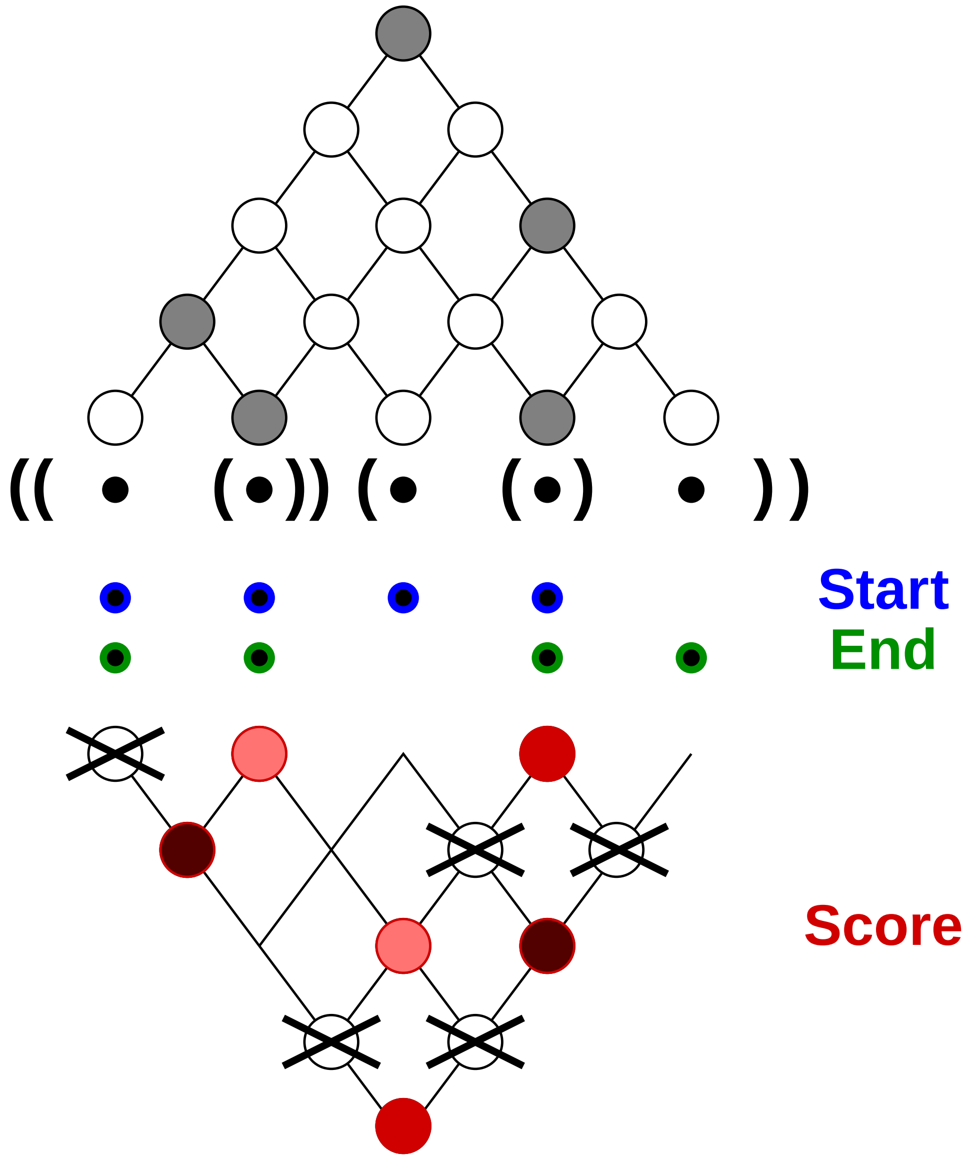
$$\mathcal{Y}_{SE} = \{y \in \mathcal{Y} \ \mid \ \forall (s, e)_k \in y \ \ \mathrm{start}_k(s) \wedge \mathrm{end}_k(e)\}$$

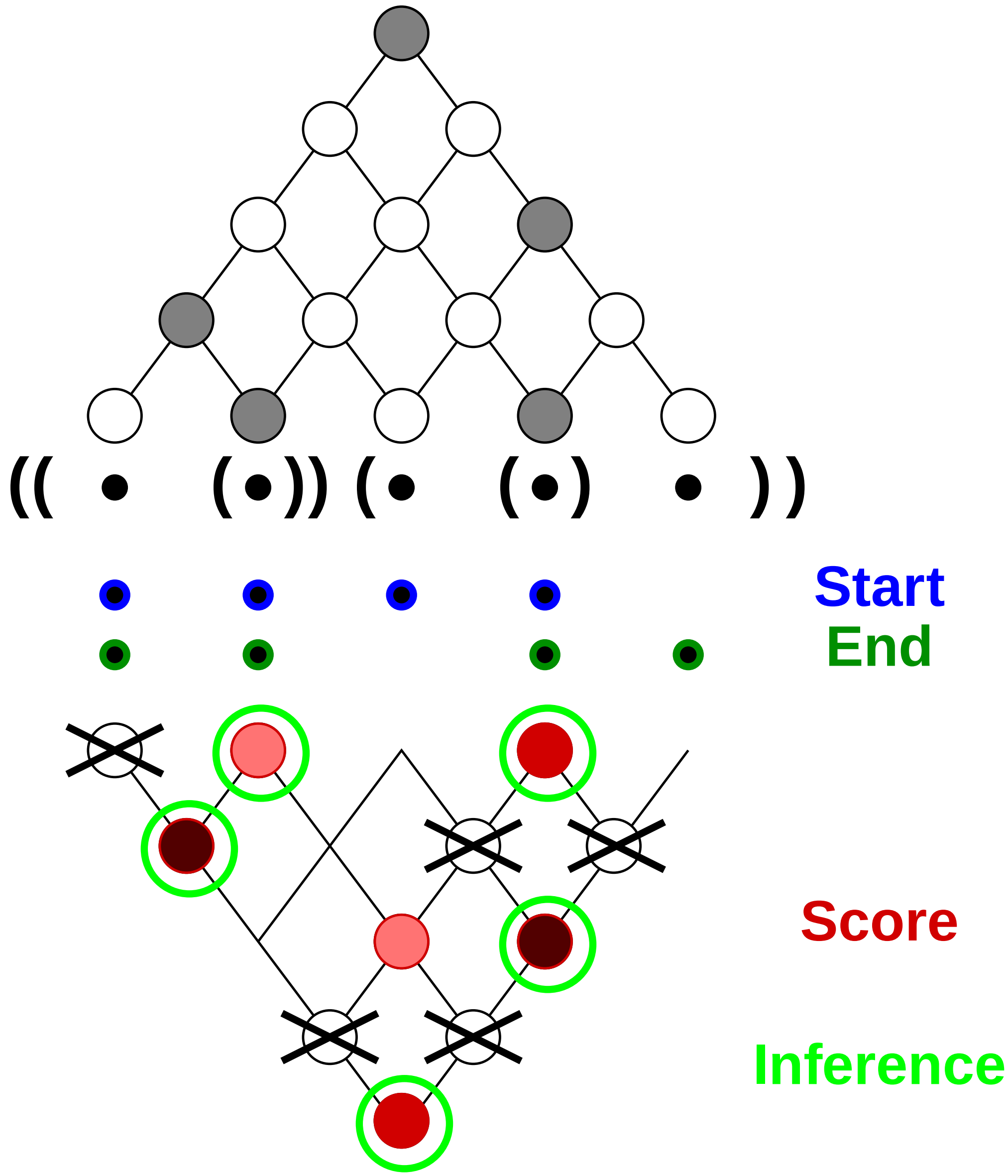



Start


((• (•)) (• (•) •))
Start
End


Start
End


((• (•)) (• (•) •))
Start
End
Score


((• (•)) (• (•) •))
Start
End
Score
Inference

Outline

- Introduction
- Phrase Recognition Model
- Global Learning Algorithm
- Experimental Evaluation
- Conclusions and Current Work

Learning Challenges

- Learn all functions $(\mathrm{start}_k, \mathrm{end}_k, \mathrm{score}_k)$ so as to maximize the F_1 measure on the recognition of phrases

Learning Challenges

- Learn all functions $(\mathrm{start}_k, \mathrm{end}_k, \mathrm{score}_k)$ so as to maximize the F_1 measure on the recognition of phrases
- Start-End:
  - ⋆ As filters, rather than default classifiers
  - ⋆ They define the input space to the score functions

Learning Challenges

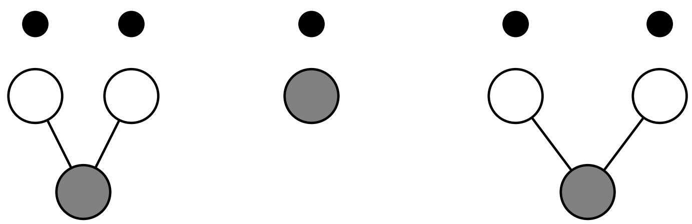
- Learn all functions $(\mathrm{start}_k, \mathrm{end}_k, \mathrm{score}_k)$ so as to maximize the F_1 measure on the recognition of phrases

- Start-End:
  - ⋆ As filters, rather than default classifiers
  - ⋆ They define the input space to the score functions

- Score functions:
  - ⋆ The space of negative examples is too big $\sim O(n^2)$
  - ⋆ We need to know about Start-End behavior
  - ⋆ As rankers, rather than default classifiers

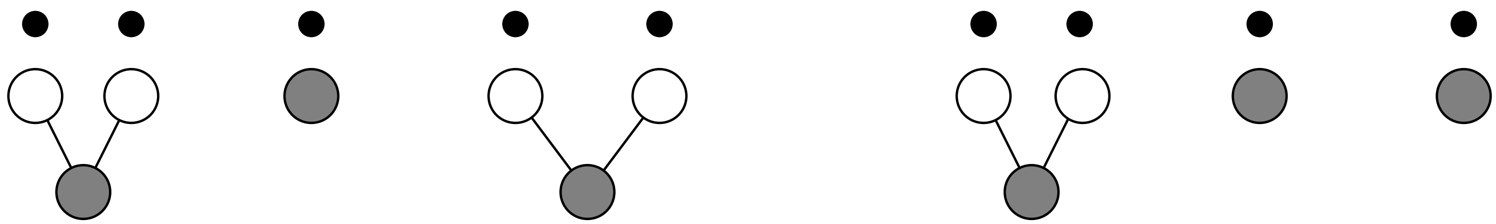
Motivation for the ranking

```
(The cat) (eats) (fresh fish) .
```

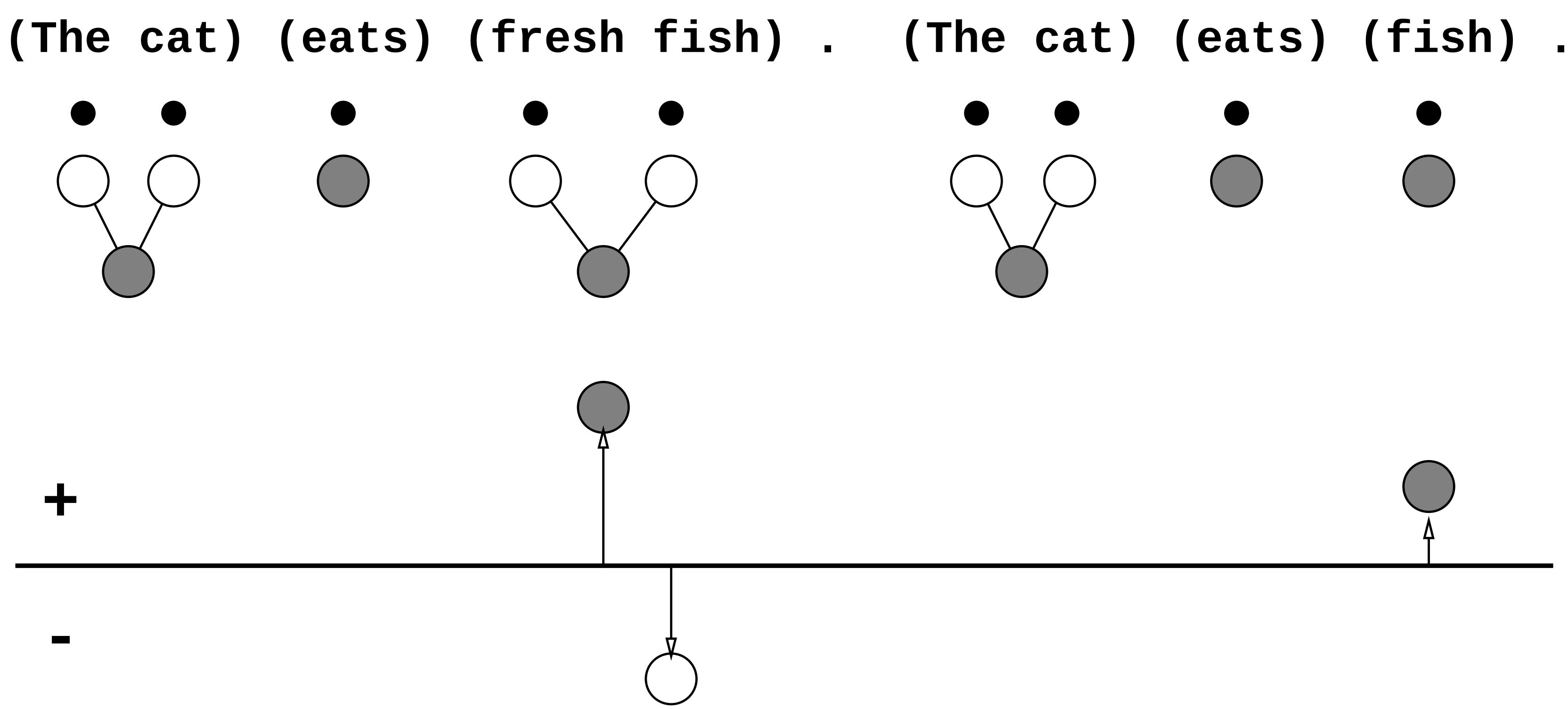


Motivation for the ranking

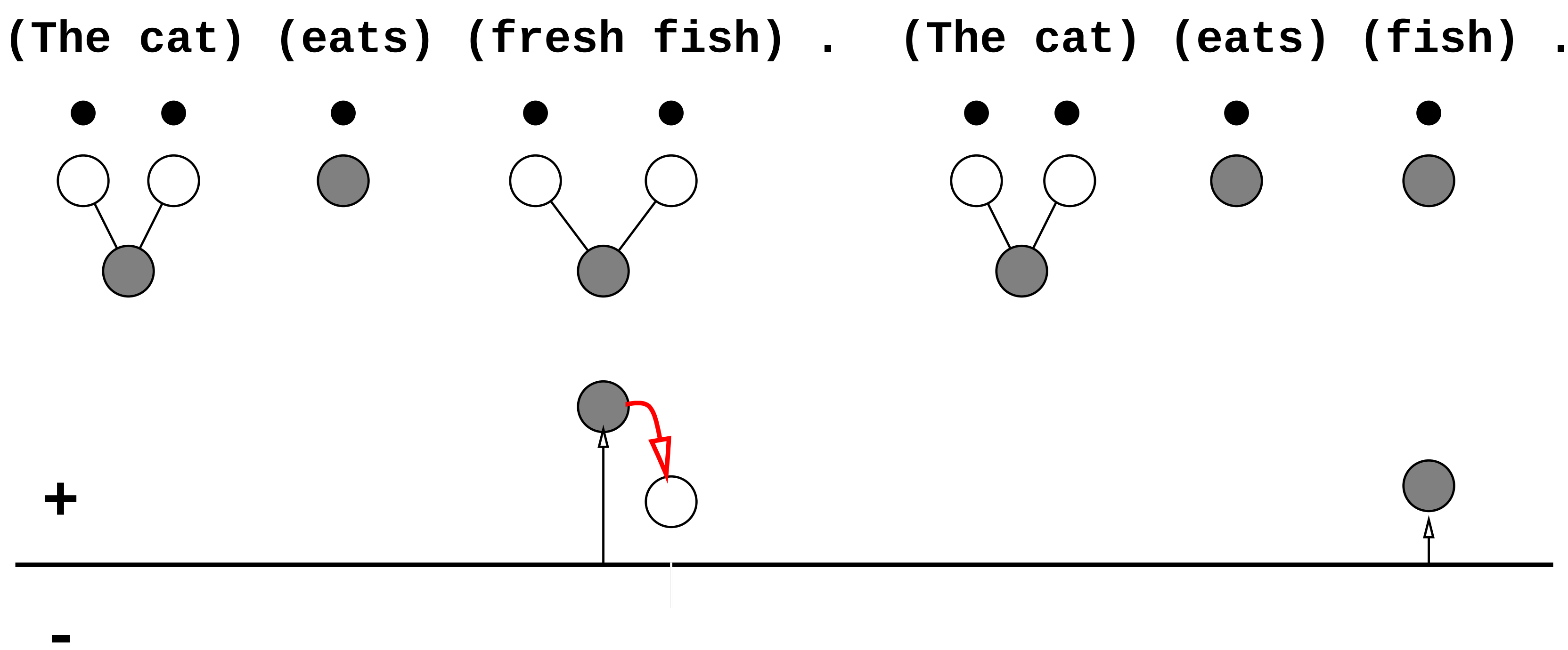
`(The cat) (eats) (fresh fish) .` `(The cat) (eats) (fish) .`



Motivation for the ranking



Motivation for the ranking



Perceptron-based Learning

- Linear discriminant function, $\mathrm{h}_{\mathbf{w}} : \mathbb{R}^n \longrightarrow \mathbb{R}$, parametrized by a weight vector $\mathbf{w}$
- Classification rule: $\mathrm{h}_{\mathbf{w}}(\mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x}) = \hat{y}$
- On-line error-driven training algorithm
- Additive updating rule: $\mathbf{w}_{t+1} = \mathbf{w}_t + y\mathbf{x}$

Perceptron-based Learning

- Linear discriminant function, $\mathrm{h}_{\mathbf{w}} : \mathbb{R}^n \longrightarrow \mathbb{R}$, parametrized by a weight vector **w**
- Classification rule: $\mathrm{h}_{\mathbf{w}}(\mathbf{x}) = sign(\mathbf{w} \cdot \mathbf{x}) = \hat{y}$
- On-line error-driven training algorithm
- Additive updating rule: $\mathbf{w}_{t+1} = \mathbf{w}_t + y\mathbf{x}$
- Representation function $\Phi : \mathcal{X} \longrightarrow \mathbb{R}^n$ to map sentence instances x into n–dimensional feature vectors

Perceptron Learning Algorithm

Input: $\{(x^1, y^1), \ldots, (x^m, y^m)\}$, x^i are sentences, y^i are solutions

Define: $W = \{\mathbf{w}_\mathrm{S}, \mathbf{w}_\mathrm{E}\} \cup \{\mathbf{w}_k | k \in \mathcal{K}\}$

Initialize: $\forall \mathbf{w} \in W \;\; \mathbf{w} = \mathbf{0}$;

for $t = 1 \ldots T$

  for $i = 1 \ldots m$

    $\hat{y} = \mathrm{PhRec}_W(x^i)$

    $\mathrm{learning\_feedback}(W, x^i, y^i, \hat{y})$

  end-for

end-for

Output: the vectors in W

Learning Feedback$_{(1)}$

- Phrases correctly identified: $\forall (s,e)_k \in\ y^* \cap \hat{y}$:
  - ⋆ Do nothing, since they are correct

Learning Feedback$_{(1)}$

- Phrases correctly identified: $\forall (s,e)_k \in\ y^* \cap \hat{y}$:
  - ⋆ Do nothing, since they are correct
- Missed phrases: $\forall (s,e)_k \in\ y^* \setminus \hat{y}$:
  - ⋆ Update misclassified boundary words:
    if $(\mathbf{w}_{\mathrm{S}} \cdot \Phi_{\mathrm{w}}(x_s) \leq 0)$ then $\mathbf{w}_{\mathrm{S}} = \mathbf{w}_{\mathrm{S}} + \Phi_{\mathrm{w}}(x_s)$
    if $(\mathbf{w}_{\mathrm{E}} \cdot \Phi_{\mathrm{w}}(x_e) \leq 0)$ then $\mathbf{w}_{\mathrm{E}} = \mathbf{w}_{\mathrm{E}} + \Phi_{\mathrm{w}}(x_e)$

Learning Feedback$_{(1)}$

- Phrases correctly identified: $\forall (s,e)_k \in\ y^* \cap \hat{y}$:
  - ⋆ Do nothing, since they are correct
- Missed phrases: $\forall (s,e)_k \in\ y^* \backslash \hat{y}$:
  - ⋆ Update misclassified boundary words:
    if $(\mathbf{w}_\mathrm{S} \cdot \Phi_\mathrm{w}(x_s) \leq 0)$ then $\mathbf{w}_\mathrm{S} = \mathbf{w}_\mathrm{S} + \Phi_\mathrm{w}(x_s)$
    if $(\mathbf{w}_\mathrm{E} \cdot \Phi_\mathrm{w}(x_e) \leq 0)$ then $\mathbf{w}_\mathrm{E} = \mathbf{w}_\mathrm{E} + \Phi_\mathrm{w}(x_e)$
  - ⋆ Update score function, if applied:
    if $(\mathbf{w}_\mathrm{S} \cdot \Phi_\mathrm{w}(x_s) > 0 \wedge \mathbf{w}_\mathrm{E} \cdot \Phi_\mathrm{w}(x_e) > 0)$ then
    $\mathbf{w}_k = \mathbf{w}_k + \Phi_\mathrm{p}(s,e)$

Learning Feedback$_{(2)}$

- Over-predicted phrases: $\forall (s,e)_k \in \ \hat{y} \backslash y^*$:
  - ⋆ Update score function: $\mathbf{w}_k = \mathbf{w}_k - \Phi_{\mathrm{p}}(s,e)$

Learning Feedback$_{(2)}$

- Over-predicted phrases: $\forall (s,e)_k \in \ \hat{y} \backslash y^*$:
  - ⋆ Update score function: $\mathbf{w}_k = \mathbf{w}_k - \Phi_{\mathrm{p}}(s,e)$
  - ⋆ Update words misclassified as S or E:
    if $(\mathrm{goldS}(s) = 0)$ then $\mathbf{w}_{\mathrm{S}} = \mathbf{w}_{\mathrm{S}} - \Phi_{\mathrm{w}}(x_s)$
    if $(\mathrm{goldE}(e) = 0)$ then $\mathbf{w}_{\mathrm{E}} = \mathbf{w}_{\mathrm{E}} - \Phi_{\mathrm{w}}(x_e)$

Learning Feedback$_{(2)}$

- Over-predicted phrases: $\forall (s,e)_k \in \ \hat{y} \backslash y^*$:
  - ⋆ Update score function: $\mathbf{w}_k = \mathbf{w}_k - \Phi_{\mathrm{p}}(s,e)$
  - ⋆ Update words misclassified as S or E:
    if $(\mathrm{goldS}(s) = 0)$ then $\mathbf{w}_{\mathrm{S}} = \mathbf{w}_{\mathrm{S}} - \Phi_{\mathrm{w}}(x_s)$
    if $(\mathrm{goldE}(e) = 0)$ then $\mathbf{w}_{\mathrm{E}} = \mathbf{w}_{\mathrm{E}} - \Phi_{\mathrm{w}}(x_e)$

- Note that we deliberately do not care about false positives, i.e., wrongly predicted *start* or *end* words which do not finally over-produce a phrase

Outline

- Introduction
- Phrase Recognition Model
- Global Learning Algorithm
- Experimental Evaluation
- Conclusions and Current Work

Experiments on NLP Problems

- CoNLL Benchmark Problems (public datasets):
  - ⋆ Syntactic Chunking (2000)
  - ⋆ Clause Identification (2001)
  - ⋆ Named Entity Recognition (2003)
- Features:
  - ⋆ Window-based features
  - ⋆ Phrase patterns
  - ⋆ Word forms, POS tags, chunk tags, affixes, orthography, etc.
  - ⋆ Filtering of features ocurring less than 3 times

Experiments on NLP Problems$_{(2)}$

- Some details about learning/evaluation:
  - ⋆ Training/developing/test data sets
  - ⋆ Voted perceptron algorithm
  - ⋆ Dual version using a degree 2 polynomial kernel
  - ⋆ Fixed number of epochs (15)
  - ⋆ ...more details in the paper

Results$_{(1)}$

		development			**test**		
	T	P	R	F_1	P	R	F_1
Chunks	10	-	-	-	94.2%	93.3%	**93.74**
Clauses	11	89.8%	84.1%	86.8	88.0%	81.0%	**84.36**
NERC	12	89.6%	88.2%	88.9	83.9%	83.4%	**83.68**

- **Chunks**:
  - ⋆ Best result at competition time
  - ⋆ Third best result ever published on this data set
  - ⋆ (Kudoh & Matsumoto, 01): F_1=93.91
  - ⋆ (Zhang et al., 02): F_1=94.17

Results$_{(2)}$

		development			**test**		
	T	P	R	F_1	P	R	F_1
Chunks	10	-	-	-	94.2%	93.3%	**93.74**
Clauses	11	89.8%	84.1%	86.8	88.0%	81.0%	**84.36**
NERC	12	89.6%	88.2%	88.9	83.9%	83.4%	**83.68**

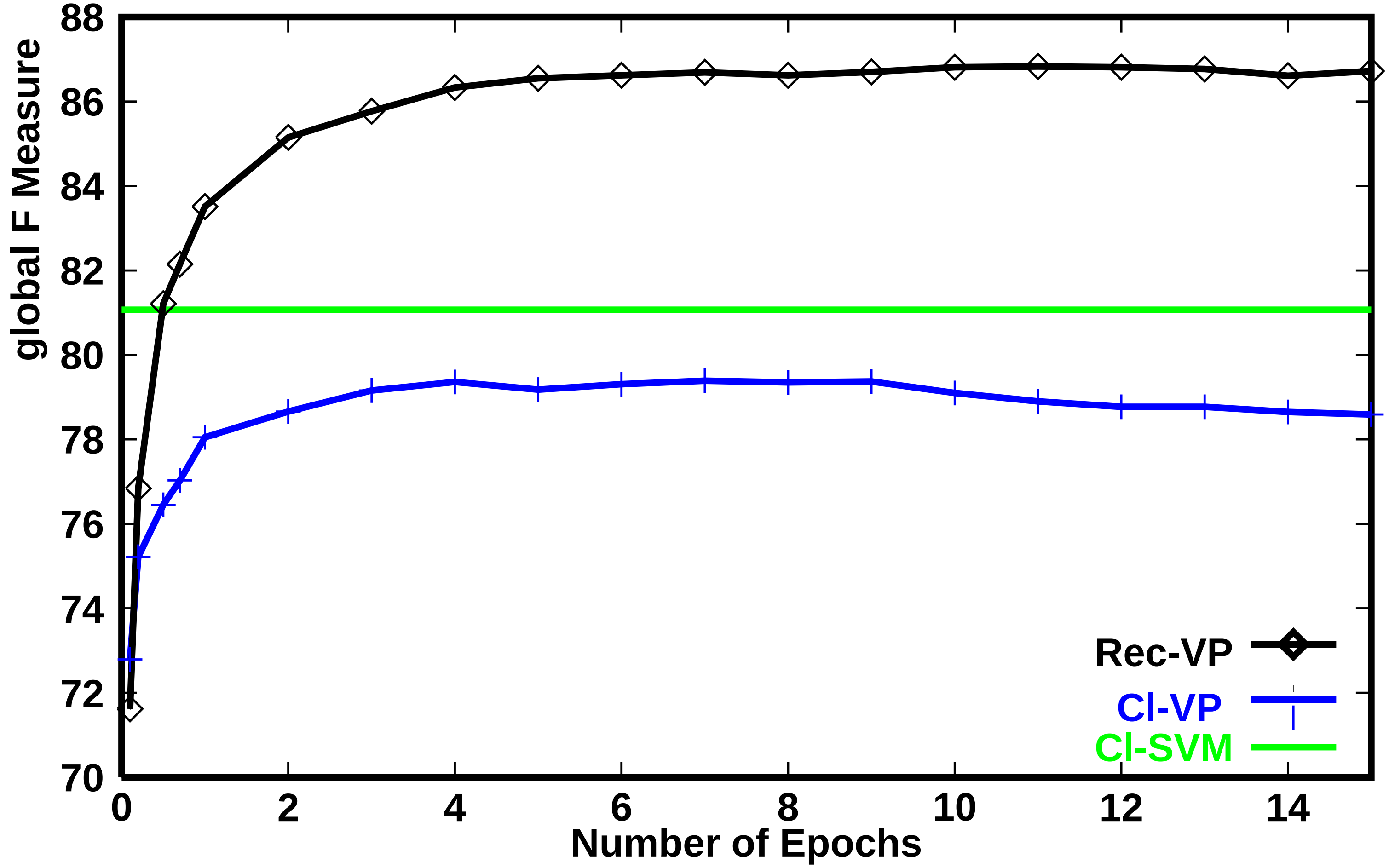
- **Clauses**:
  - ⋆ Best result ever published on this data set
  - ⋆ (Carreras et al., 2002): F_1=83.71

Results$_{(3)}$

		development			**test**		
	T	P	R	F_1	P	R	F_1
Chunks	10	-	-	-	94.2%	93.3%	**93.74**
Clauses	11	89.8%	84.1%	86.8	88.0%	81.0%	**84.36**
NERC	12	89.6%	88.2%	88.9	83.9%	83.4%	**83.68**

- **NERC**:
  - ⋆ Lower results but competitive
  - ⋆ NE recognition depends more on the features (also external knowledge) than on the structure

Does Global Learning Work Better?



Conclusions

- We have presented a general 2-layer perceptron-based learning architecture for phrase recognition problems, and an online learning algorithm to train all the perceptrons together

Conclusions

- We have presented a general 2-layer perceptron-based learning architecture for phrase recognition problems, and an online learning algorithm to train all the perceptrons together

- Some good properties:
  - ⋆ Good results on several NLP problems
  - ⋆ The learning feedback takes into account the global solution
  - ⋆ Training the functions together is better than training them separately
  - ⋆ On-line fashion: deals with negative examples in a natural way
  - ⋆ Simplicity and flexibility of the model
  - ⋆ Rich features can be developed at phrase level

Current/Future Work

- Convergence proofs for the training algorithm and theoretical bounds on generalization: coming soon!
- Further study of the interaction between layers during training
- Solving several NLP tasks at the same time: POS tagging + chunking; chunking + clausing; full parsing; etc.

Thank you very much for your attention!