Hidden–Variable Models for Discriminative Reranking

Terry Koo and Michael Collins

{maestro|mcollins}@csail.mit.edu

CSAIL

MIT COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE LABORATORY

Overview of reranking

The reranking approach

Use a baseline model to get the *N*-best candidates

“Rerank” the candidates using a more complex model

Parse reranking

Collins (2000): 88.2% $\Rightarrow$ 89.8%

Charniak and Johnson (2005): 89.7% $\Rightarrow$ 91.0%

Talk by Brooke Cowan in 7B: 83.6% $\Rightarrow$ 85.1%

Also applied to

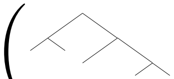
MT (Och and Ney, 2002; Shen et al., 2004)

NL Generation (Walker et al., 2001)

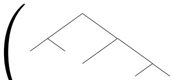
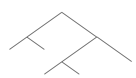
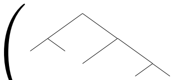
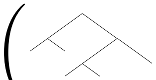
Representing NLP structures

Proper representation is critical to success

Hand–crafted feature vector representations

$\Phi($  $) = \{0, 1, 2, 0, 0, 3, 0, 1\}$

Features defined through kernels

$K($  $,$  $) = \Phi($  $) \cdot \Phi($  $)$

This talk: A new approach using hidden variables

Two facets of lexical items

Different lexical items can have similar meanings, e.g. *president* and *chairman*

Clustering: *president*, *chairman* $\in$ NounCluster$_4$

A single lexical item can have different meanings, e.g. [river] *bank* vs [financial] *bank*

Refinement: *bank*$_1$, *bank*$_2$ $\in$ *bank*

Model clusterings and refinements as hidden variables that support the reranking task

Highlights of the approach

Conditional log–linear model with hidden variables

Dynamic programming is used for training and decoding

Clustering and refinement done automatically using a discriminative criterion

Overview of talk

- Motivation
- Design
  - General form of the model
  - Training and decoding efficiently
  - Creating specific instantiations
- Results
- Discussion
- Conclusion

The parse reranking framework

Sentences s_i for $1 \leq i \leq n$

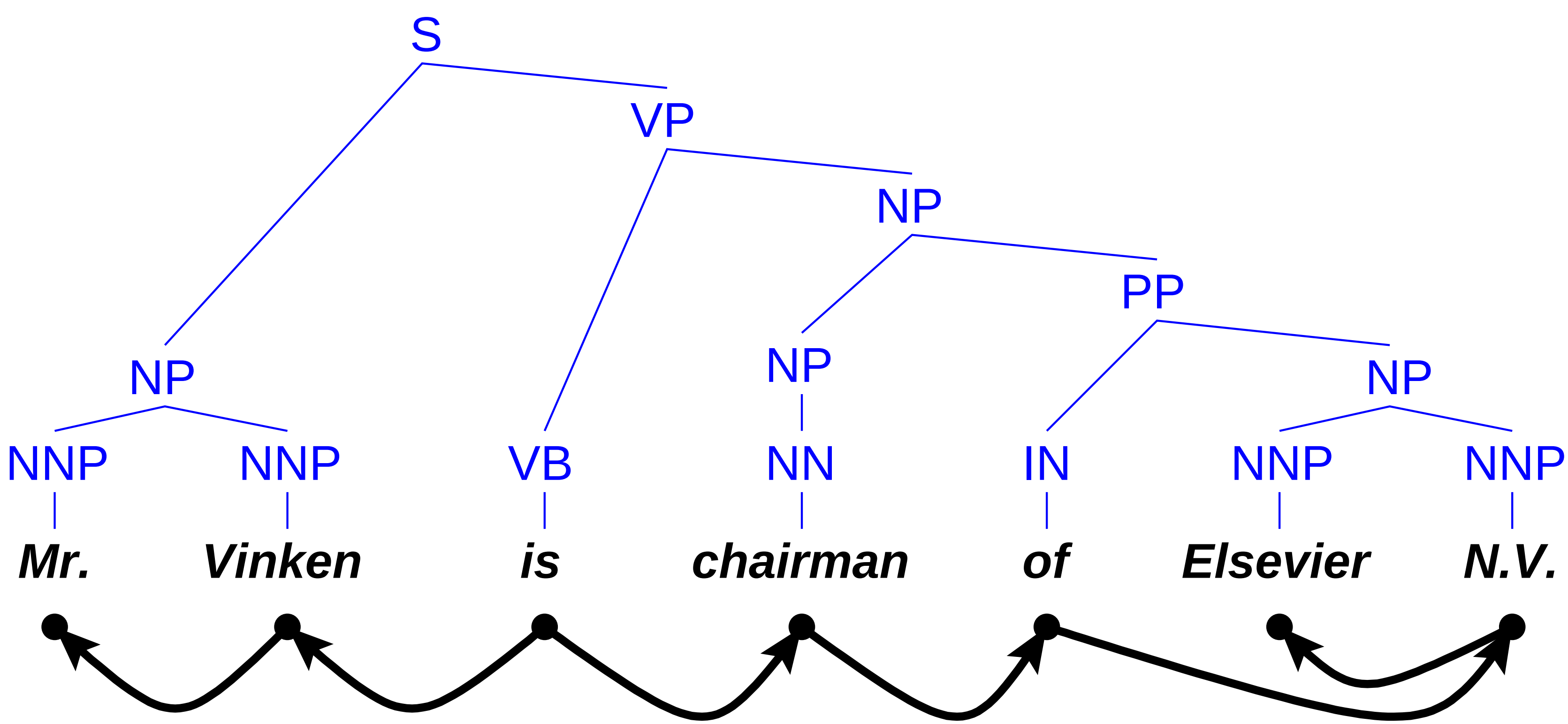
- s_1: ***Pierre Vinken , 61 years old , will join ...***
- s_2: ***Mr. Vinken is chairman of Elsevier N.V. ...***
- s_3: ***Big Board Chairman John Phelan said yesterday ...***

Each s_i has candidate parses $t_{i,j}$ for $1 \leq j \leq n_i$

- $t_{i,1}$ **is the best candidate parse for** s_i

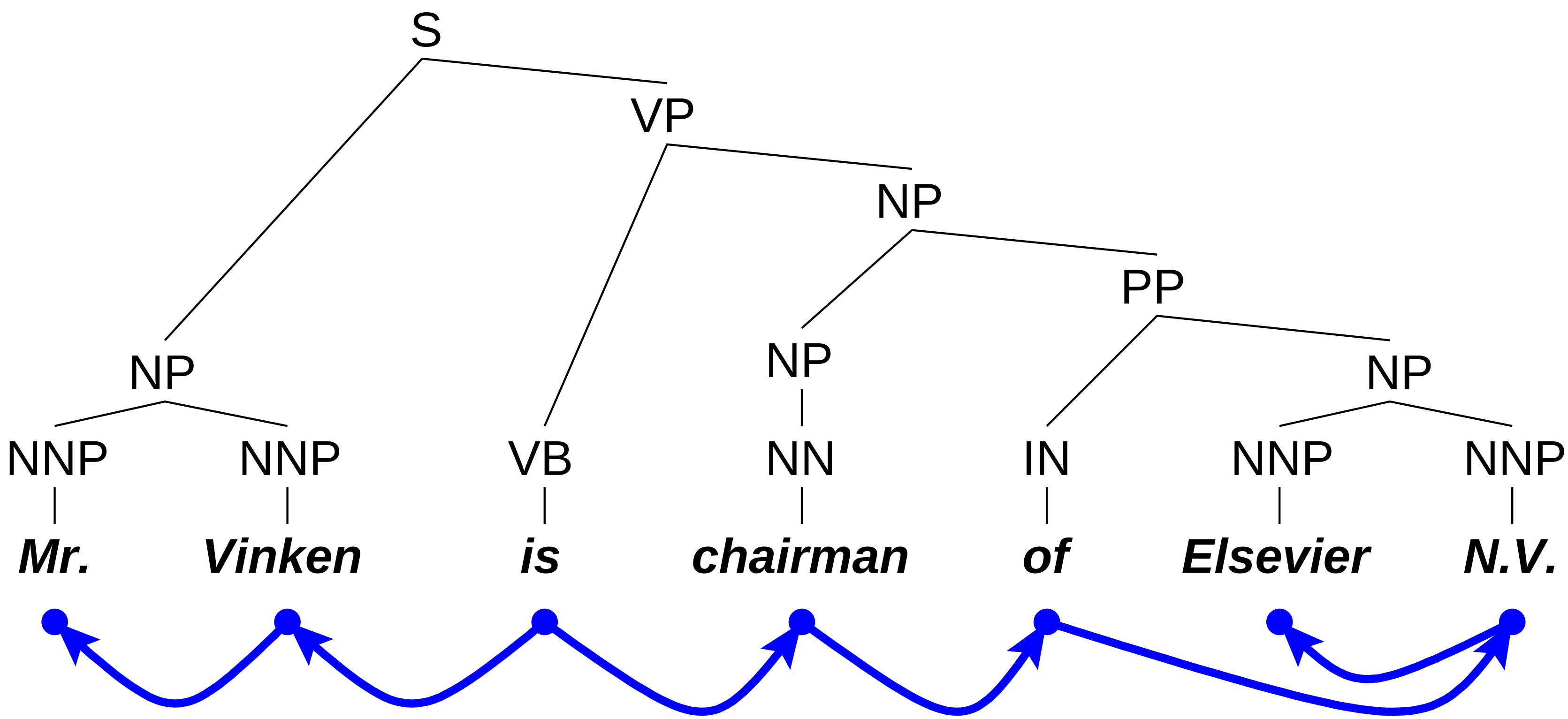
The parse reranking framework

$t_{i,j}$ has phrase structure and dependency tree



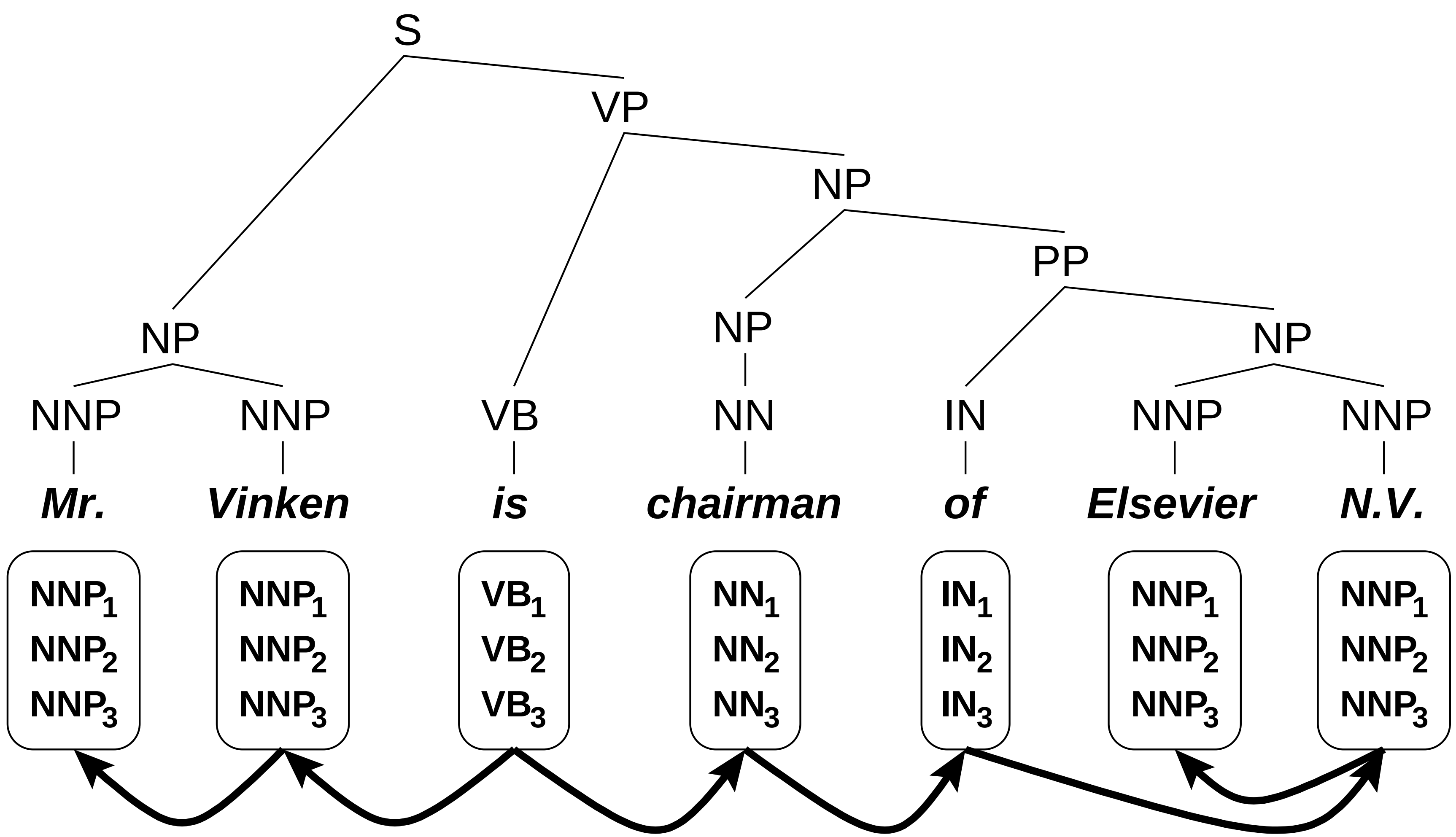
The parse reranking framework

$t_{i,j}$ has phrase structure and dependency tree



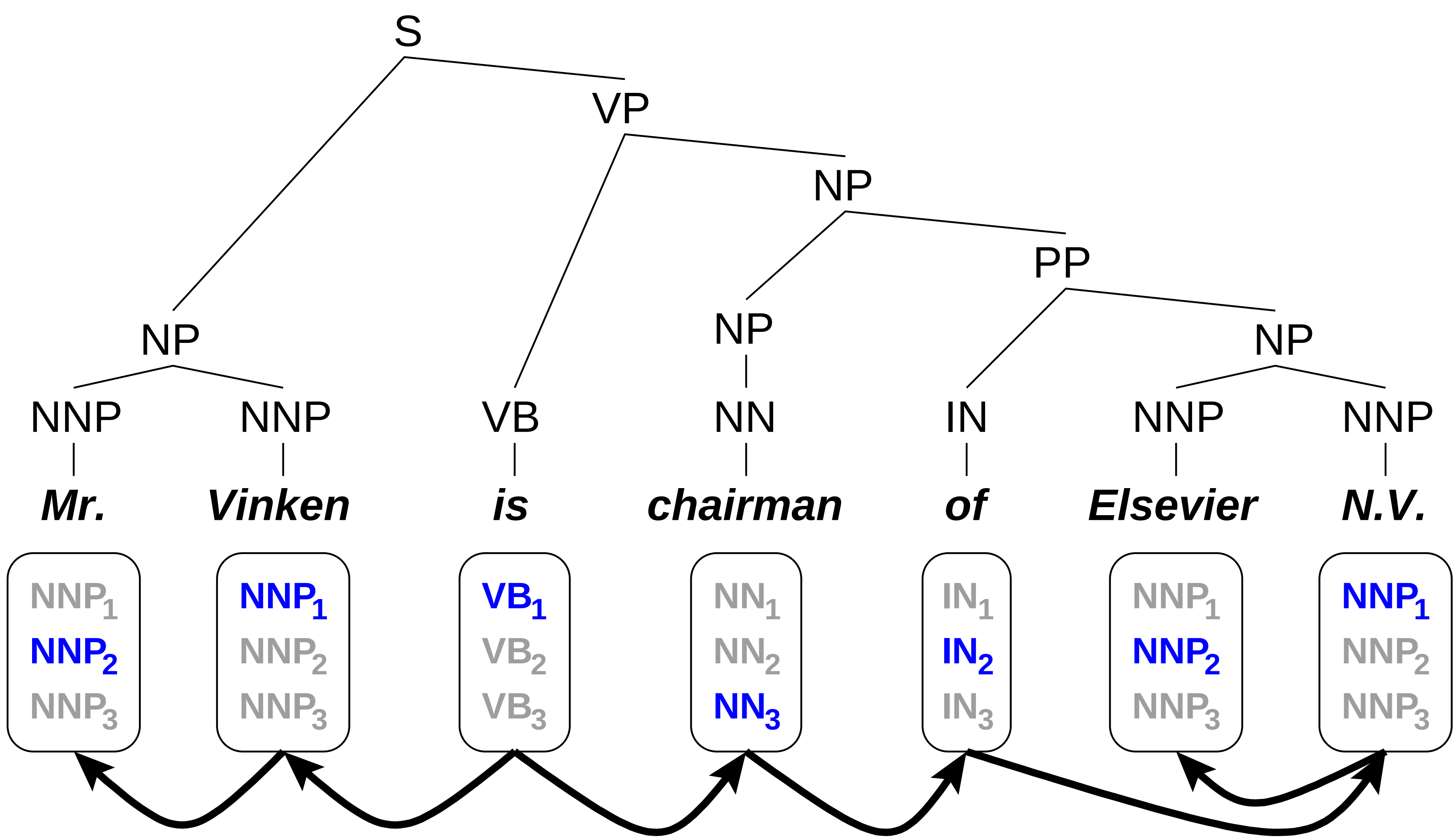
Adding hidden variables

Hidden–value domains $H_w(t_{i,j})$ for $1 \leq w \leq \text{len}(s_i)$



Adding hidden variables

Assignment $h \in H_1(t_{i,j}) \times \ldots \times H_{\mathrm{len}(s_i)}(t_{i,j})$



Marginalized probability model

$\Phi(t_{i,j}, h)$ produces a descriptive vector of feature occurrence counts, e.g.

$\Phi_2(t_{i,j}, h)$ = Count(*chairman* has hidden value NN_1)

$\Phi_{13}(t_{i,j}, h)$ = Count(NNP_2 is a direct object of VB_1)

$\Phi_{19}(t_{i,j}, h)$ = Count(NN_1 coordinates with NN_2)

Marginalized probability model

Log–linear distribution over ($t_{i,j}$, h) with parameters Θ:

$$p(t_{i,j}, h \mid s_i, \Theta) = \frac{e^{\Phi(t_{i,j}, h)\cdot\Theta}}{\sum_{j',h'} e^{\Phi(t_{i,j'}, h')\cdot\Theta}}$$

Marginalize over assignments h:

$$p(t_{i,j} \mid s_i, \Theta) = \sum_{h} p(t_{i,j}, h \mid s_i, \Theta)$$

Optimizing the parameters

Define loss as negative log-likelihood

$$L(\Theta) = -\sum_{i=1}^{n} \log p(t_{i,1} \mid s_i, \Theta)$$

Minimize $L(\Theta)$ through gradient descent

$$\frac{\partial L}{\partial \Theta} = -\sum_{i}\sum_{h} p(h \mid t_{i,1}, s_i, \Theta)\Phi(t_{i,1}, h)$$
$$+ \sum_{i,j} p(t_{i,j} \mid s_i, \Theta) \sum_{h} p(h \mid t_{i,j}, s_i, \Theta)\Phi(t_{i,j}, h)$$

Overview of talk

- Motivation
- Design
  - General form of the model
  - Training and decoding efficiently
  - Creating specific instantiations
- Results
- Discussion
- Conclusion

Problems with efficiency

$|H_1(t_{i,j}) \times \ldots \times H_{\text{len}(s_i)}(t_{i,j})|$ grows exponentially, so training the model is intractable:

$$\frac{\partial L}{\partial \Theta} = -\sum_i \sum_h p(h \mid t_{i,1}, s_i, \Theta)\Phi(t_{i,1}, h)$$
$$+ \sum_{i,j} p(t_{i,j} \mid s_i, \Theta) \sum_h p(h \mid t_{i,j}, s_i, \Theta)\Phi(t_{i,j}, h)$$

Decoding the model is also intractable:

$$p(t_{i,j} \mid s_i, \Theta) = \sum_h p(t_{i,j}, h \mid s_i, \Theta)$$

Problems with efficiency

$|H_1(t_{i,j}) \times \ldots \times H_{\mathrm{len}(s_i)}(t_{i,j})|$ grows exponentially, so training the model is intractable:

$$\frac{\partial L}{\partial \Theta} = -\sum_i \sum_h p(h \mid t_{i,1}, s_i, \Theta)\Phi(t_{i,1}, h) + \sum_{i,j} p(t_{i,j} \mid s_i, \Theta) \sum_h p(h \mid t_{i,j}, s_i, \Theta)\Phi(t_{i,j}, h)$$

Decoding the model is also intractable:

$$p(t_{i,j} \mid s_i, \Theta) = \sum_h p(t_{i,j}, h \mid s_i, \Theta)$$

Locality constraint on features

Features have pairwise local scope on hidden variables

Features still have global scope on non-hidden information

Φ can be factored into local feature vectors, allowing dynamic programming

Local feature vectors

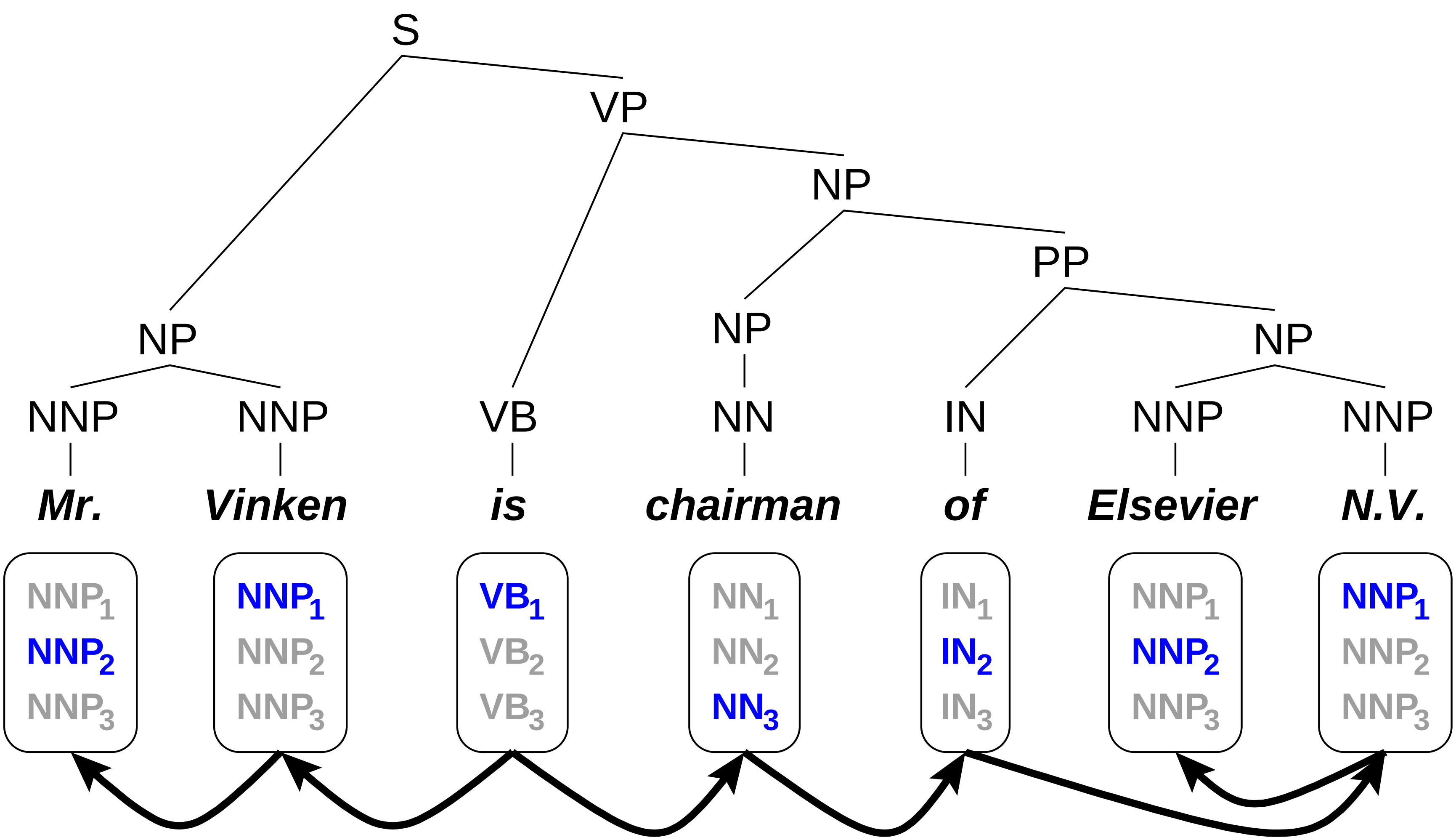
Define two kinds of local feature vector ϕ:

Single-variable $\phi(t_{i,j}, w, h_w)$ look at a single hidden variable

Pairwise $\phi(t_{i,j}, u, v, h_u, h_v)$ look at two hidden variables in a dependency relationship

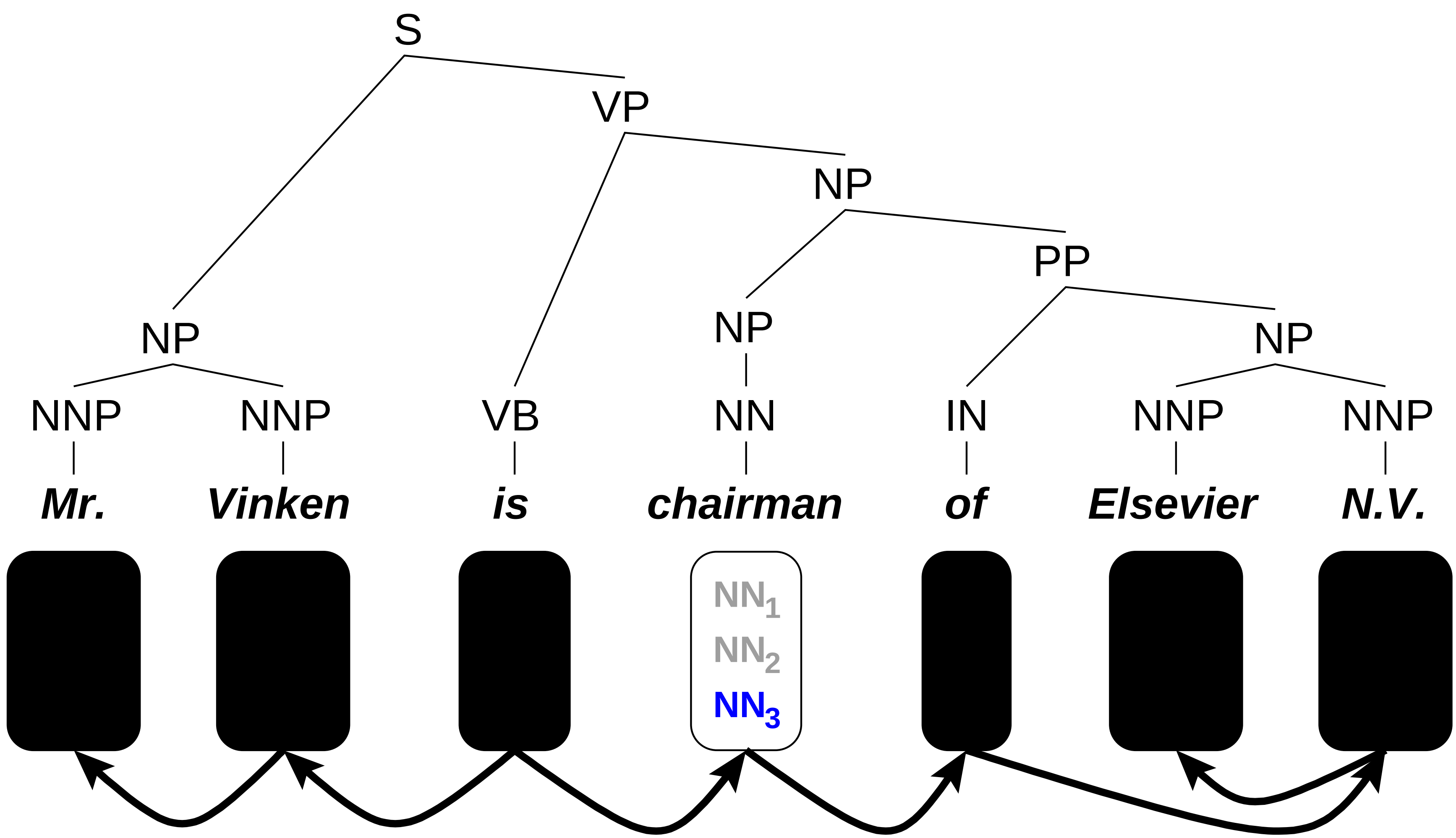
Local feature vectors

$\Phi(t_{i,j}, h)$ looks at every hidden variable



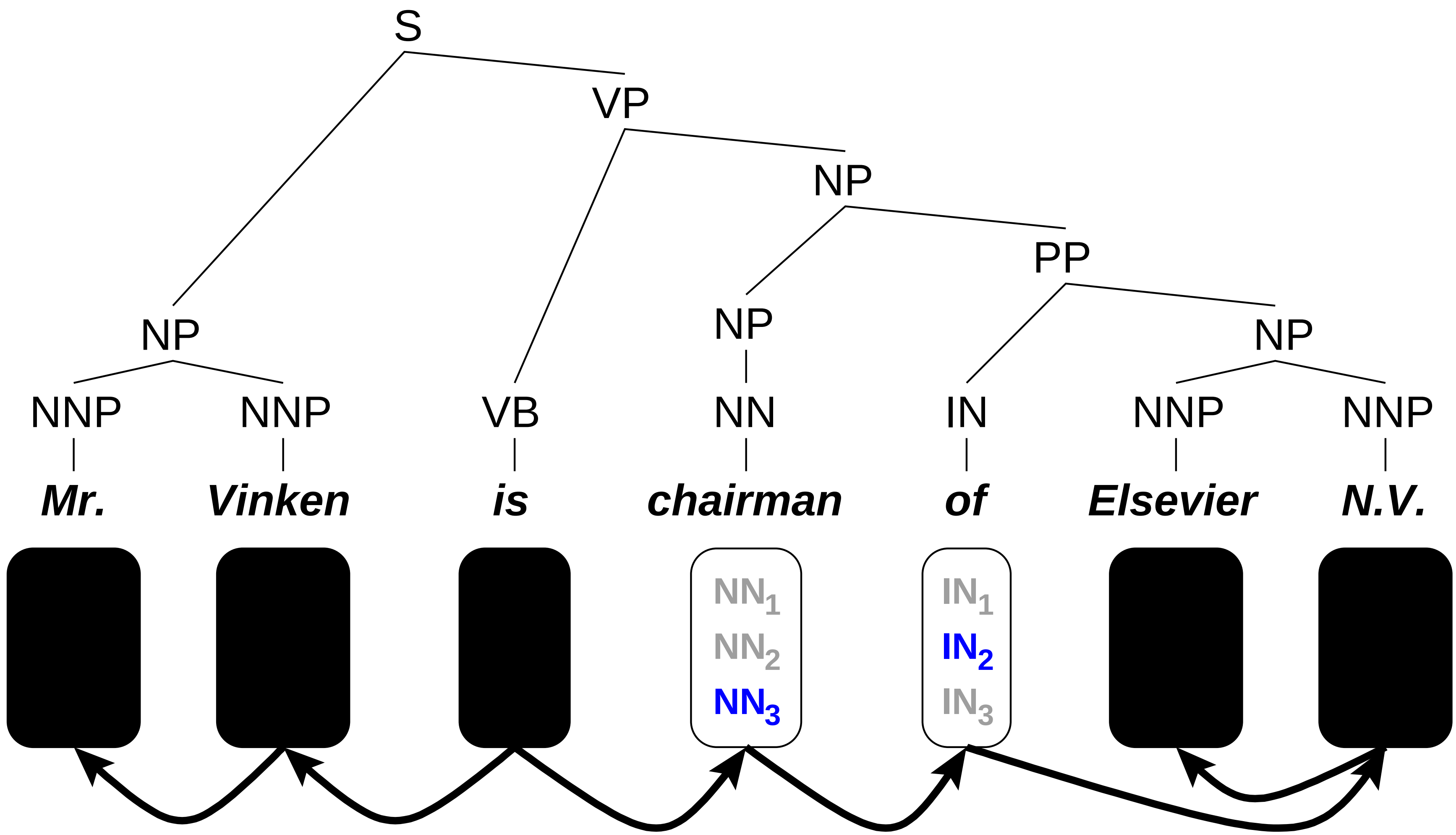
Local feature vectors

$\phi(t_{i,j}$, *chairman*, $\text{NN}_3)$ only sees NN_3



Local feature vectors

$\phi(t_{i,j}, \textit{chairman}, \textit{of}, NN_3, IN_2)$ sees NN_3 and IN_2



Local feature vectors

Rewrite global Φ as a sum over local ϕ

$$\Phi(t_{i,j}, h) = \sum_{w \in t_{i,j}} \phi(t_{i,j}, w, h_w) + \sum_{(u,v) \in D(t_{i,j})} \phi(t_{i,j}, u, v, h_u, h_v)$$

Local feature vectors

Rewrite global Φ as a sum over local ϕ

$$\Phi(t_{i,j}, h) = \sum_{w \in t_{i,j}} \phi(t_{i,j}, w, h_w)$$

$$+ \sum_{(u,v) \in D(t_{i,j})} \phi(t_{i,j}, u, v, h_u, h_v)$$

Applying belief propagation

New restrictions enable dynamic–programming approaches, e.g. belief propagation

BP generalizes the forward–backward algorithm from a chain to a tree

Runtime $O(\text{len}(s_i)H^2)$, $H = \max |H_w(t_{i,j})|$

BP efficiently computes

$$\sum_h p(t_{i,j}, h \mid s_i, \Theta)$$

$$\sum_h p(h \mid t_{i,j}, s_i, \Theta)\Phi(t_{i,j}, h)$$

Overview of talk

- Motivation
- Design
  - General form of the model
  - Training and decoding efficiently
  - Creating specific instantiations
- Results
- Discussion
- Conclusion

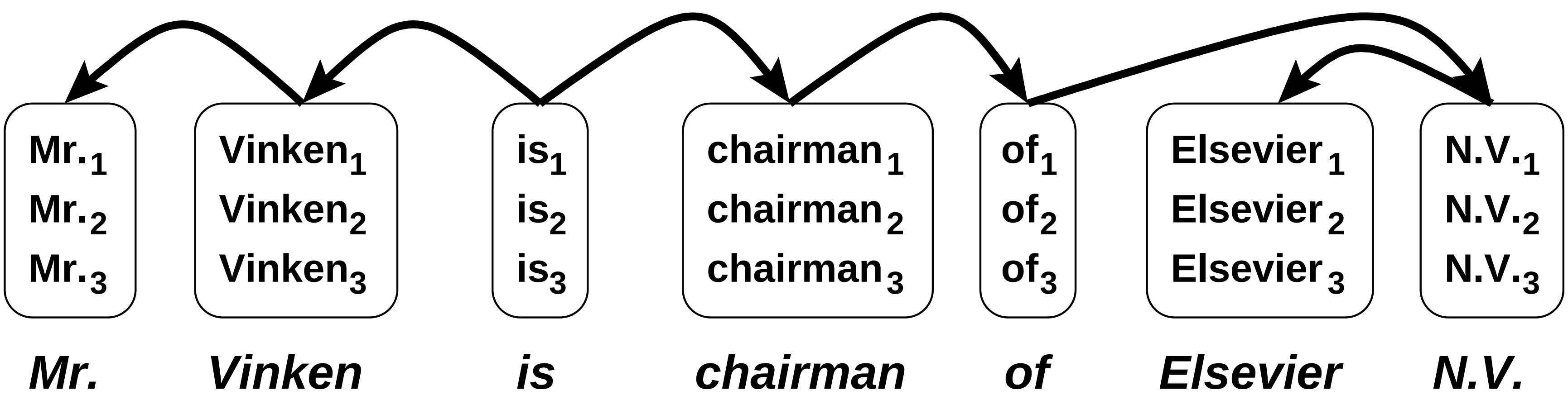
Two areas for choice in the model

Definition of the hidden–value domains $H_w(t_{i,j})$

Definition of the feature vectors ϕ

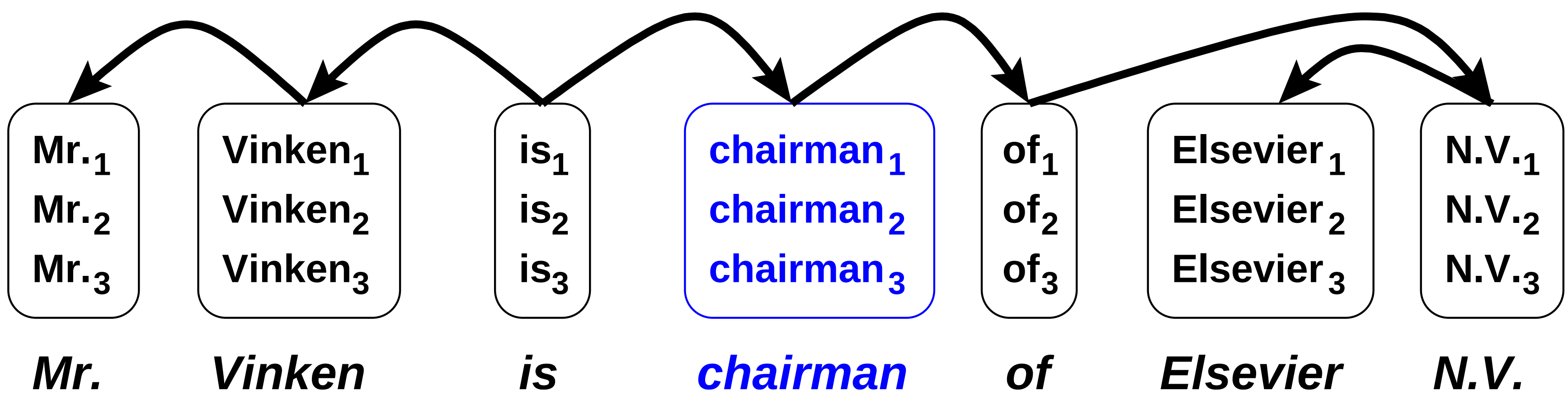
Hidden–value domains

Lexical domains allow word refinement



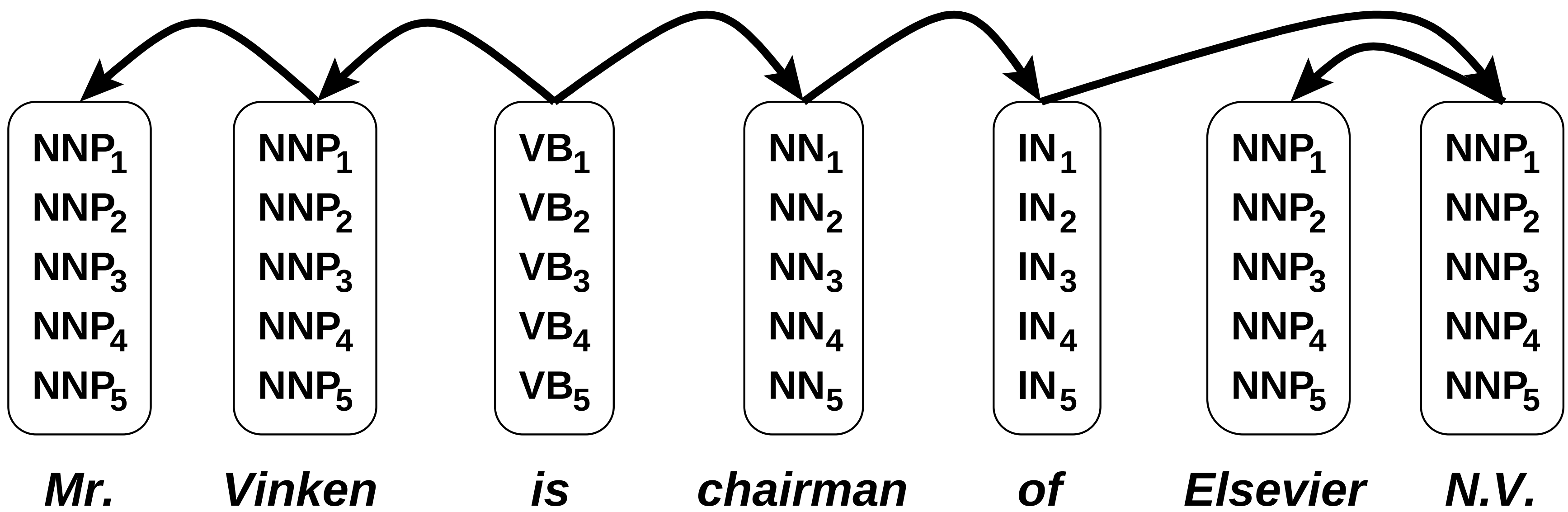
Hidden–value domains

Lexical domains allow word refinement



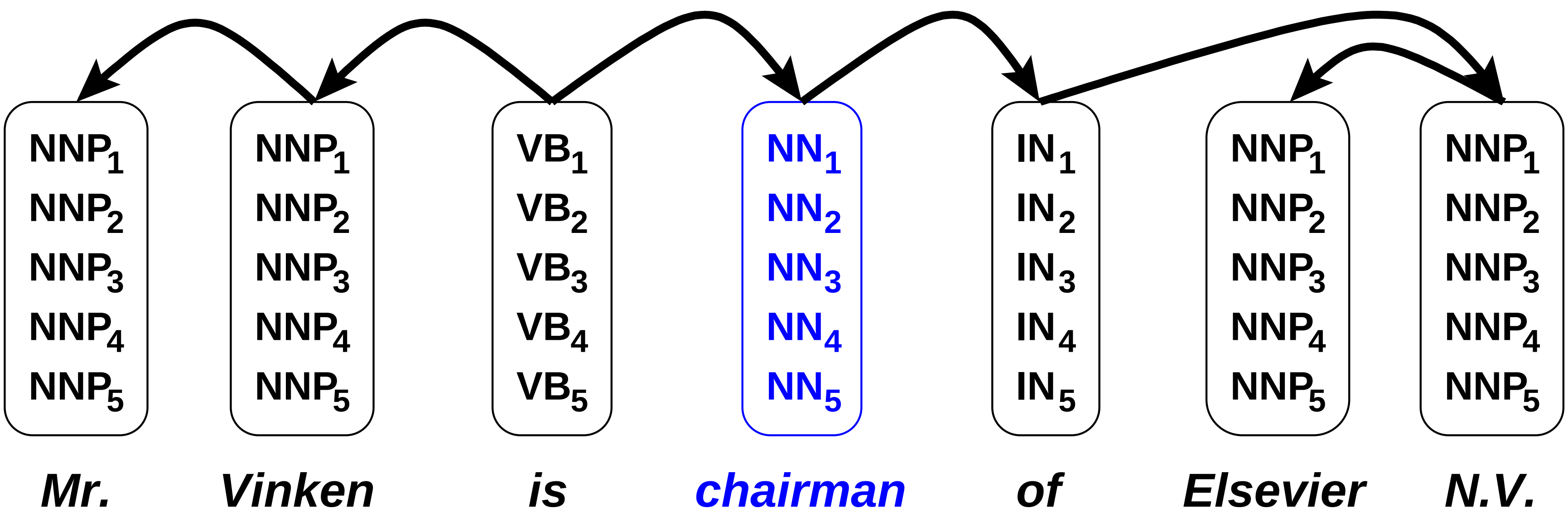
Hidden–value domains

Part-of-speech domains allow word clustering



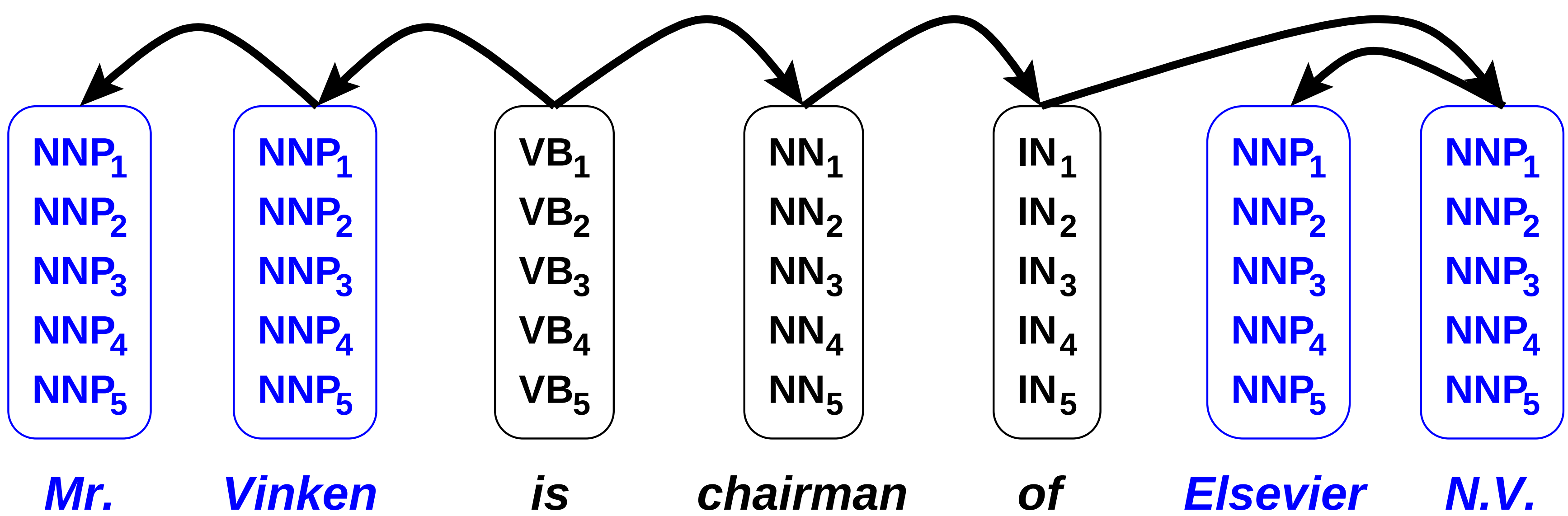
Hidden–value domains

Part-of-speech domains allow word clustering



Hidden–value domains

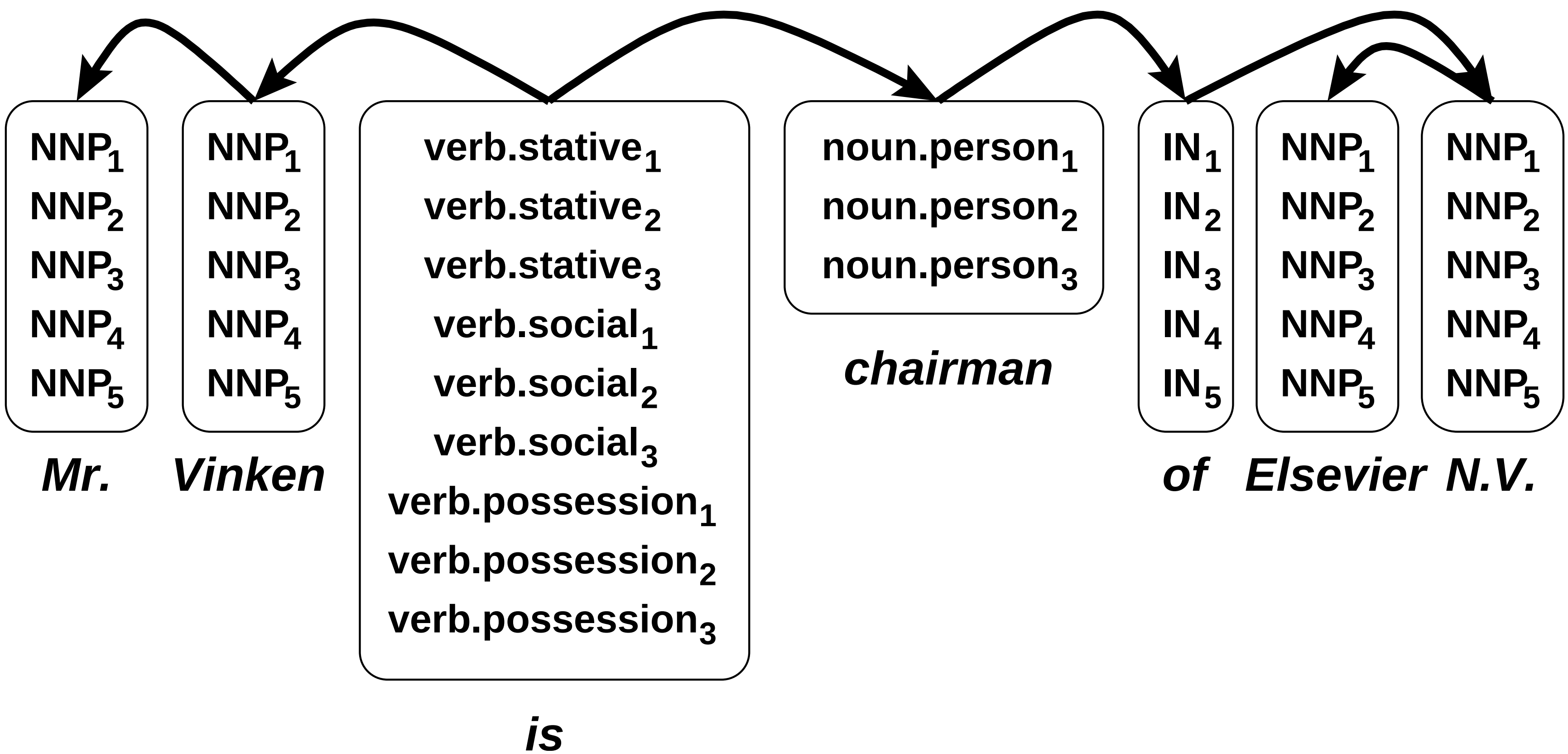
Part-of-speech domains allow word clustering



Hidden–value domains

Supersense domains model WordNet ontology

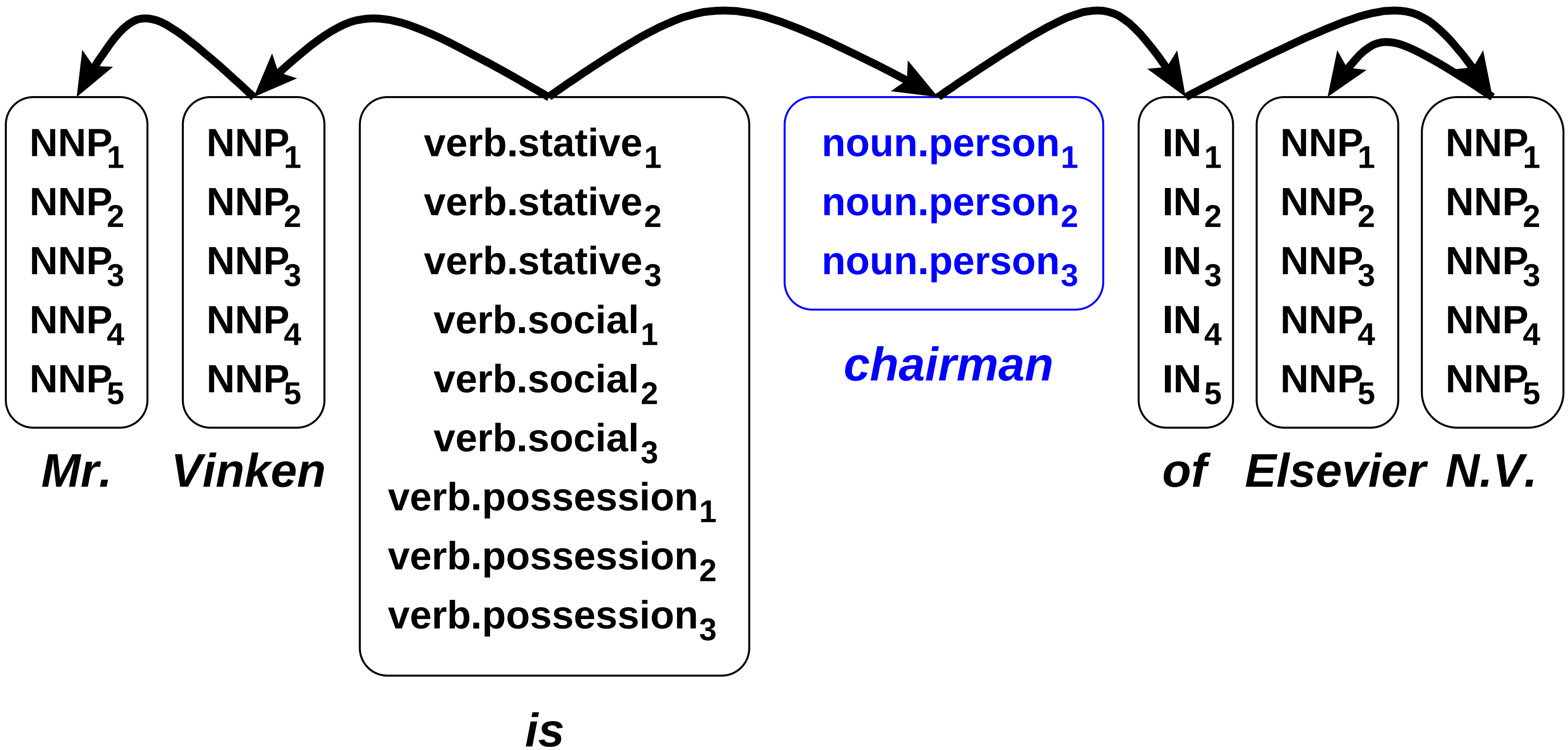
(Ciaramita and Johnson, 2003; Miller et at., 1993)



Hidden–value domains

Supersense domains model WordNet ontology

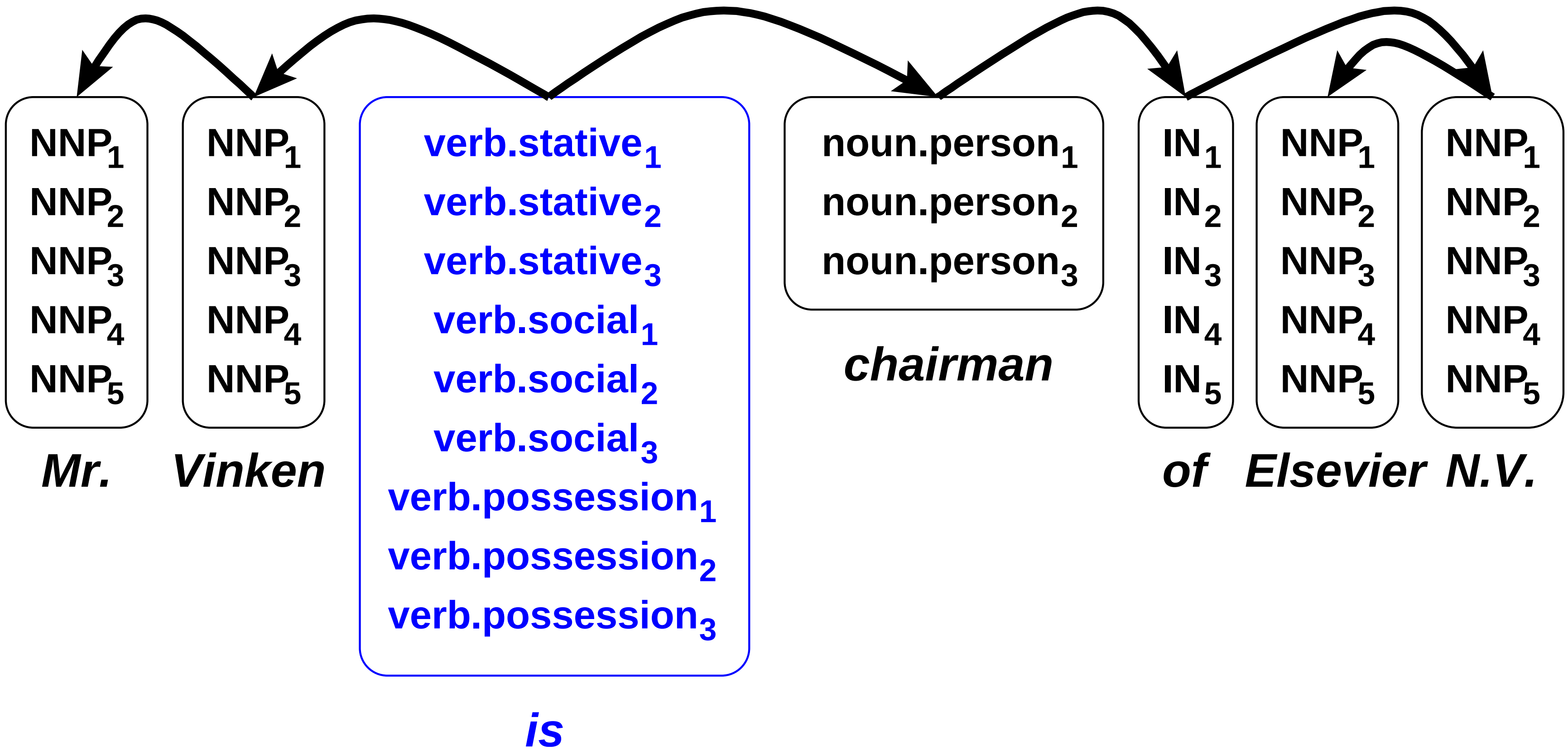
(Ciaramita and Johnson, 2003; Miller et at., 1993)



Hidden–value domains

Supersense domains model WordNet ontology

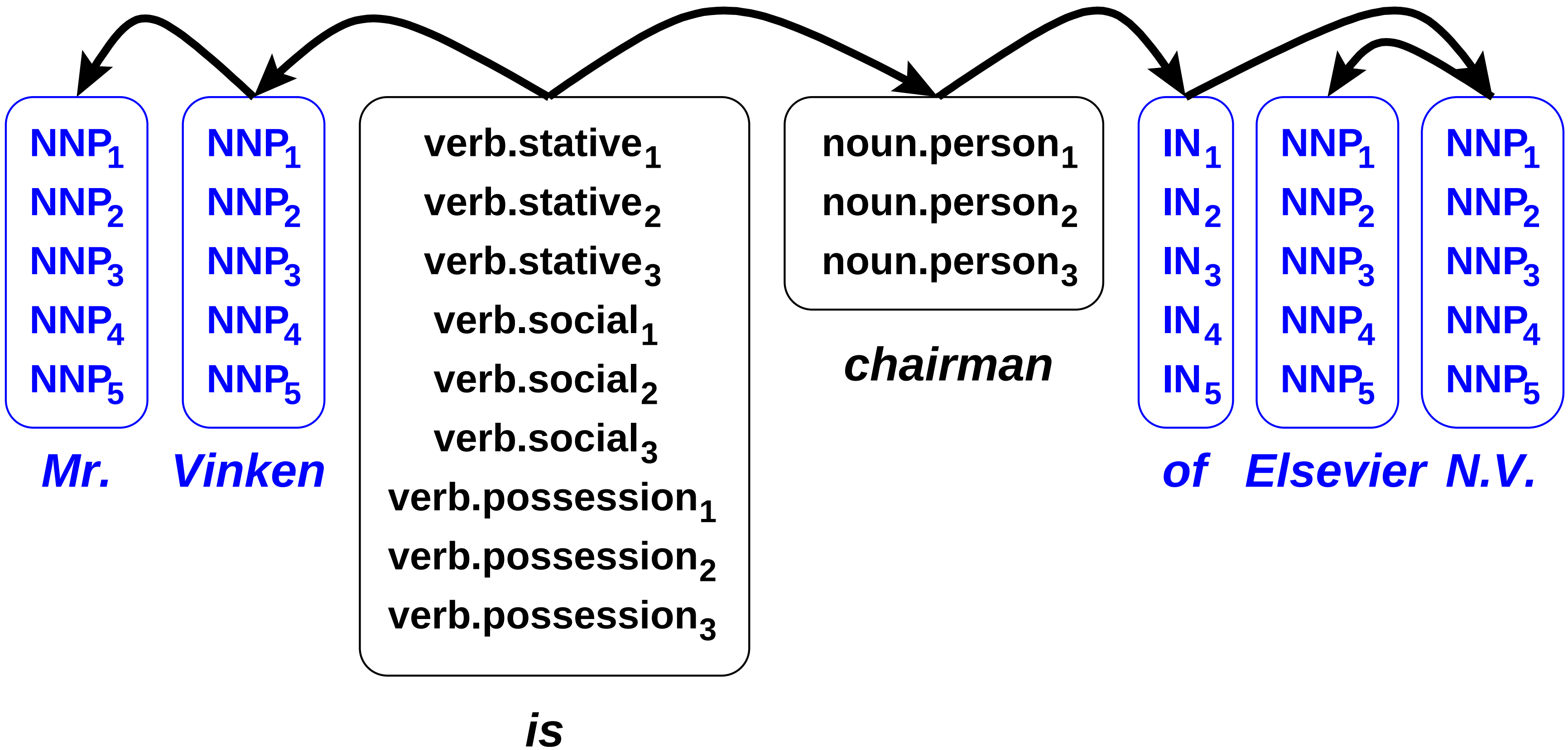
(Ciaramita and Johnson, 2003; Miller et at., 1993)



Hidden–value domains

Supersense domains model WordNet ontology

(Ciaramita and Johnson, 2003; Miller et at., 1993)

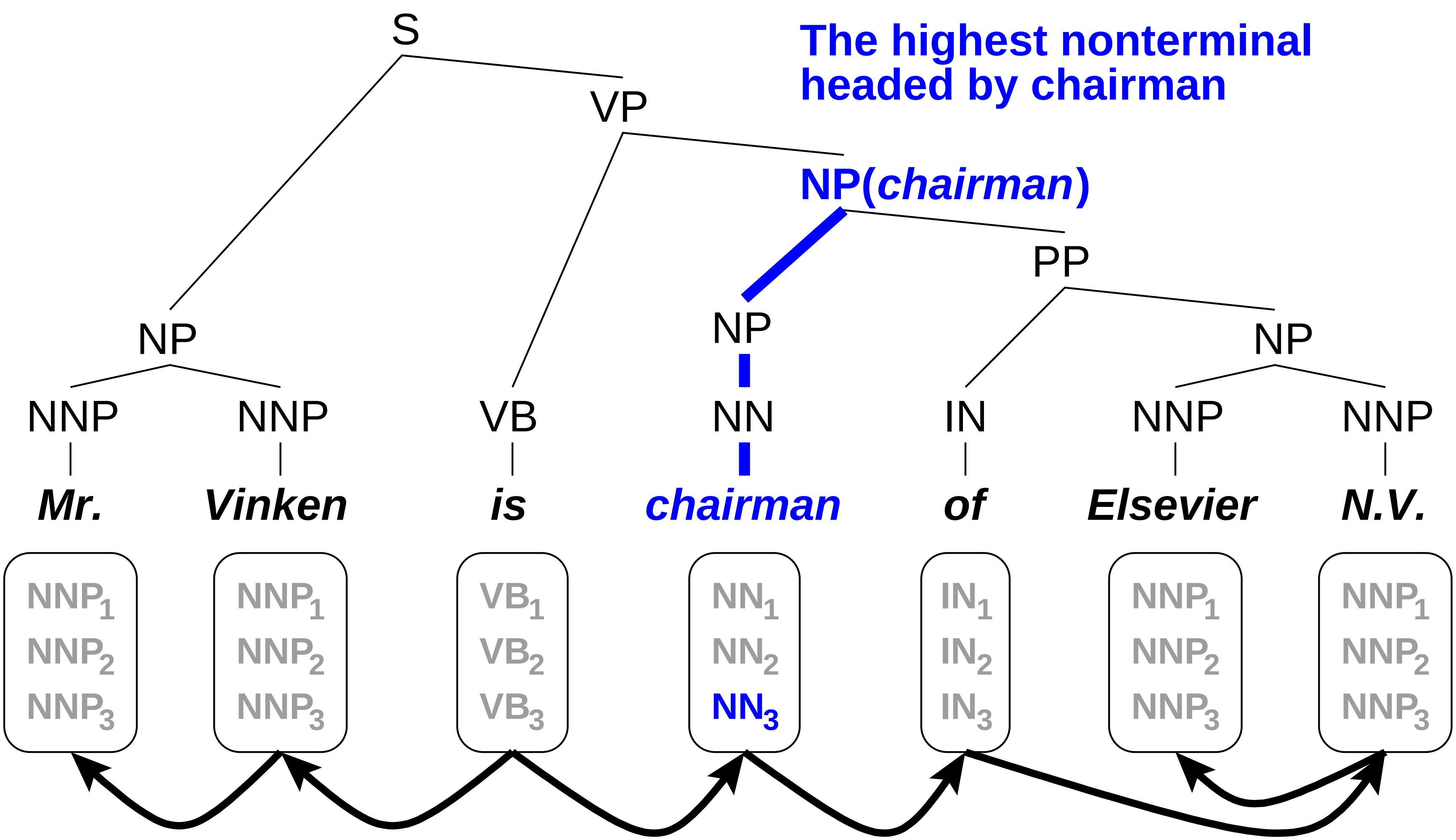


Hidden–value domains

Hidden–value domains that didn't work well

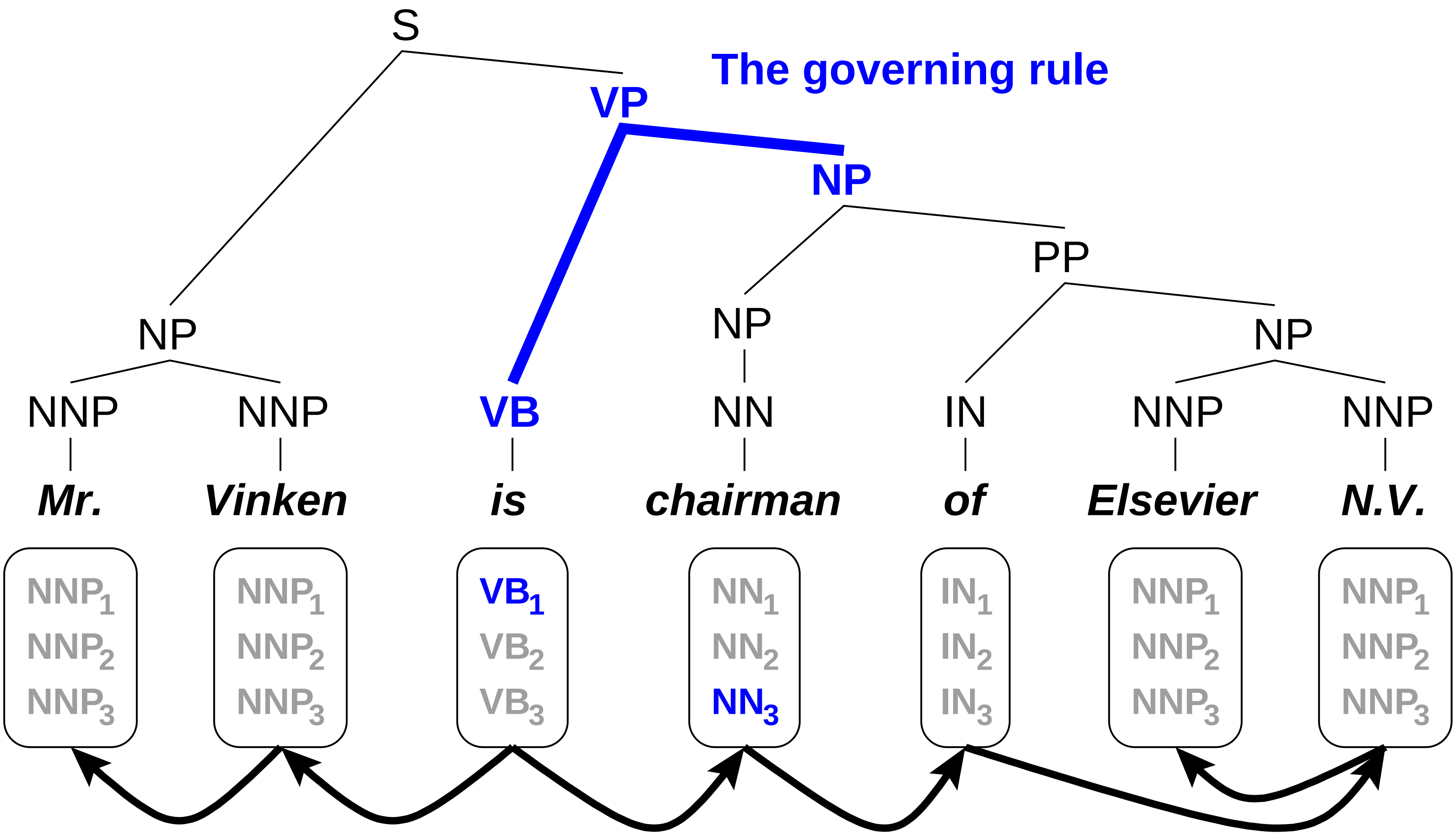
- Word clustering without part-of-speech subdivisions
- WordNet hyper/hyponym ontology
- Domains containing mixed values

Examples of features



(NN_3, Word=*chairman*, Highest Nonterminal=NP) $\in \phi(t_{i,j}$, *chairman*, NN_3)

Examples of features



$(\mathbf{VB_1}, \mathbf{NN_3}, \text{Rule=}\mathbf{VP} \rightarrow \mathbf{VB\ NP}) \in \phi(t_{i,j}, \textit{is}, \textit{chairman}, \mathbf{VB_1}, \mathbf{NN_3})$

Overview of talk

Motivation

Design

Results

Discussion

Conclusion

Experimental Setup

N-best lists generated by Collins parser, $n_i \approx 30$

Training set:	WSJ sections 2–21
Development set:	WSJ section 0
Test set:	WSJ sections 22-24

CSAIL

Final test models

Two baseline models

- The Collins (1999) base parser
- The Collins (2000) reranker

Two mixed models

- *MIX* combined clustering, refinement, and WordNet
- *MIX*+ augments *MIX* with features of Collins reranker

Results on Sections 22–24

	LR	LP	F_1
Collins parser	88.19	88.60	88.39
MIX	89.41	89.87	89.64
Collins reranker	89.46	90.07	89.76
MIX+	89.78	90.29	90.03

All comparisons except *MIX* vs. Collins reranker are significant at $p \leq 0.01$ using the sign test

Results on Sections 22–24

	LR	LP	F_1
Collins parser	88.19	88.60	88.39
MIX	89.41	89.87	89.64
Collins reranker	89.46	90.07	89.76
MIX+	89.78	90.29	90.03

All comparisons except *MIX* vs. Collins reranker are significant at $p \leq 0.01$ using the sign test

Results on Sections 22–24

	LR	LP	F_1
Collins parser	88.19	88.60	88.39
MIX	89.41	89.87	89.64
Collins reranker	89.46	90.07	89.76
MIX+	89.78	90.29	90.03

All comparisons except *MIX* vs. Collins reranker are significant at $p \leq 0.01$ using the sign test

Overview of talk

- Motivation
- Design
- Results
- Discussion
- Conclusion

Previous work

Parsing approaches that use hidden variables

- Riezler et al. (2002)
- Clark and Curran (2004)
- Matsuzaki et al. (2005)

Differences with our approach

- Use of reranking
- Definition of hidden variables
- Use of belief propagation

Using packed representations

Candidates $t_{i,j}$ represented as a packed forest

- Compact representation of *many* parse trees

Packed representation forces local scope

- Features would become locally scoped on non-hidden information
- Decoding becomes NP-hard, must approximate with Viterbi (cf. Matsuzaki et al., 2005):

$$\underset{t_{i,j}}{\operatorname{argmax}}\ p(t_{i,j} \mid s_i, \Theta) \approx \underset{t_{i,j},h}{\operatorname{argmax}}\ p(t_{i,j}, h \mid s_i, \Theta)$$

Empirical analysis of hidden values

The model makes hidden–value assignments on the basis of the reranking criterion

i.e. maximize $\sum \log p(t_{i,1} \mid s_i, \Theta)$

The empirical distribution of assignments shows linguistically reasonable trends

Overview of talk

Motivation

Design

Results

Discussion

Conclusion

Concluding remarks

The hidden–variable model defines a new representation for NLP structures

Conditional log–linear model with hidden variables

BP enables efficient and exact training and decoding

Significant improvement over Collins (2000)

CSAIL

Example

I [will [give/VB_2 an example]]

I expected [to [give/VB_1 an example]]

I expected [to/TO_4 [give an example]]

You expected [me [to/$TO_{1,5}$ [give an example]]]