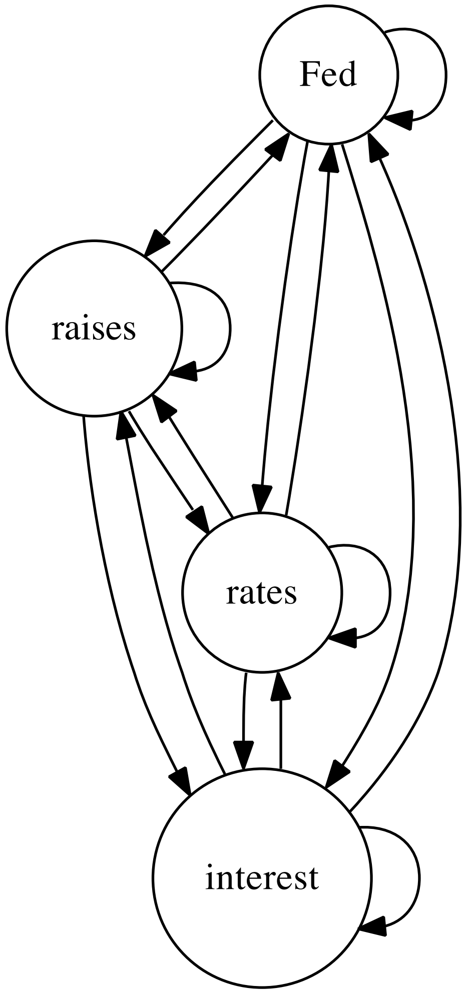
Hidden Markov Models: Maxing and Summing

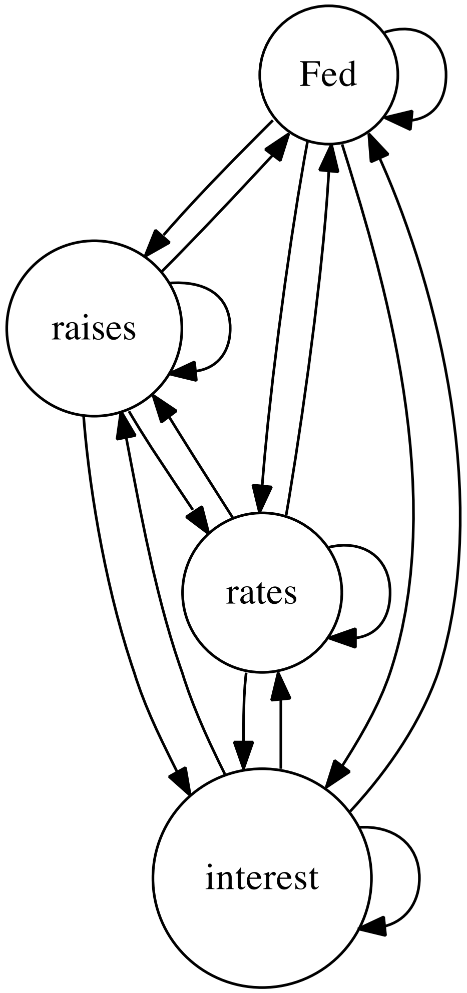
Introduction to Natural Language Processing
Computer Science 585—Fall 2009
University of Massachusetts Amherst

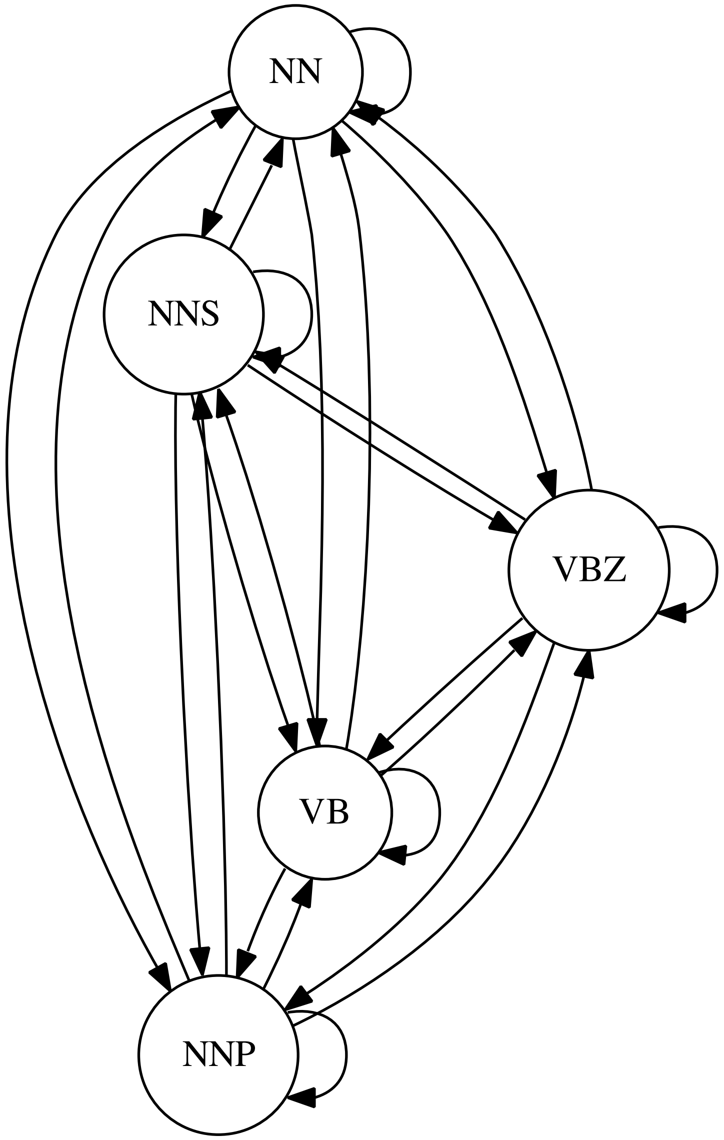
David Smith

Markov vs. Hidden Markov Models

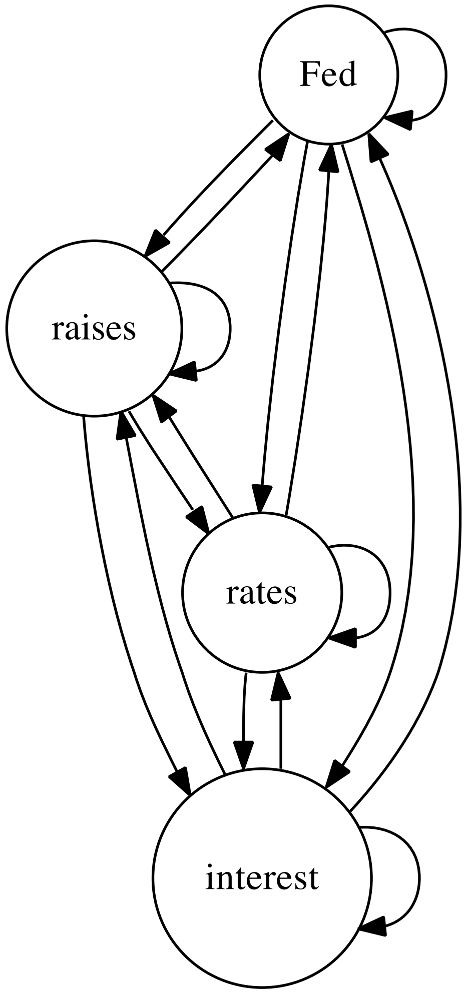


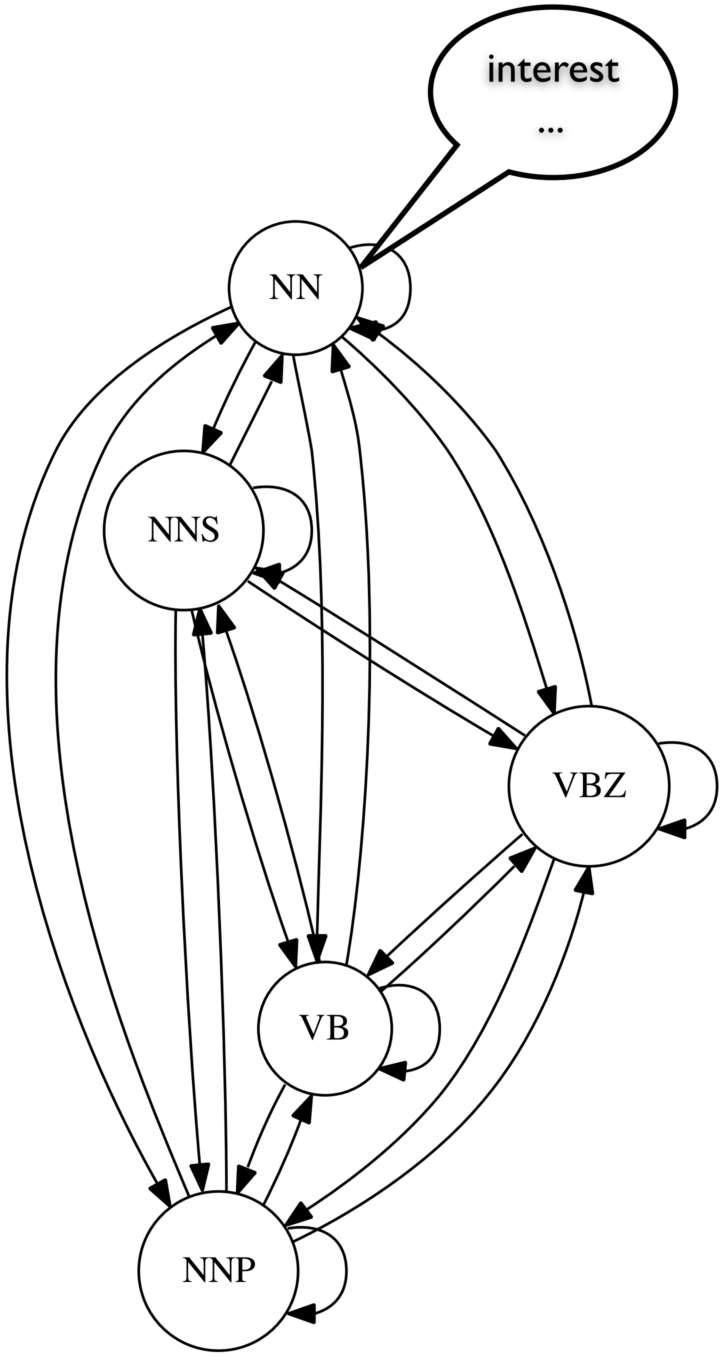
Markov vs. Hidden Markov Models



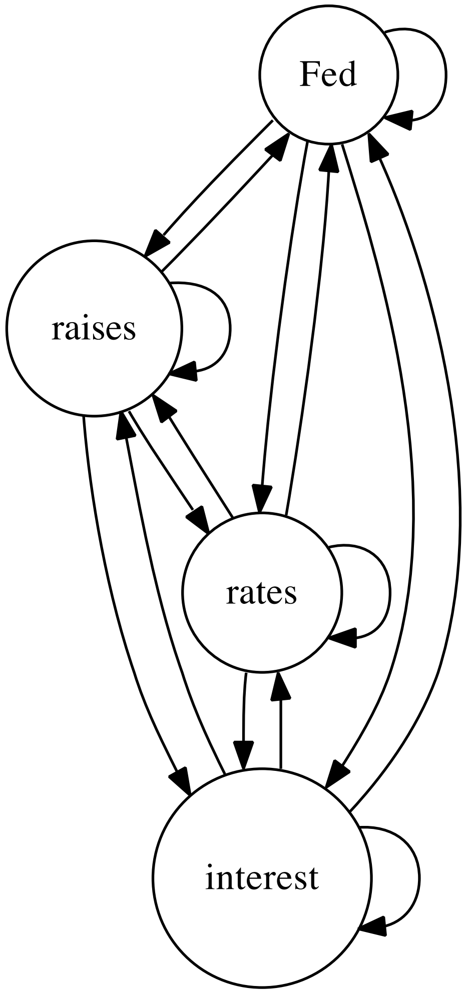


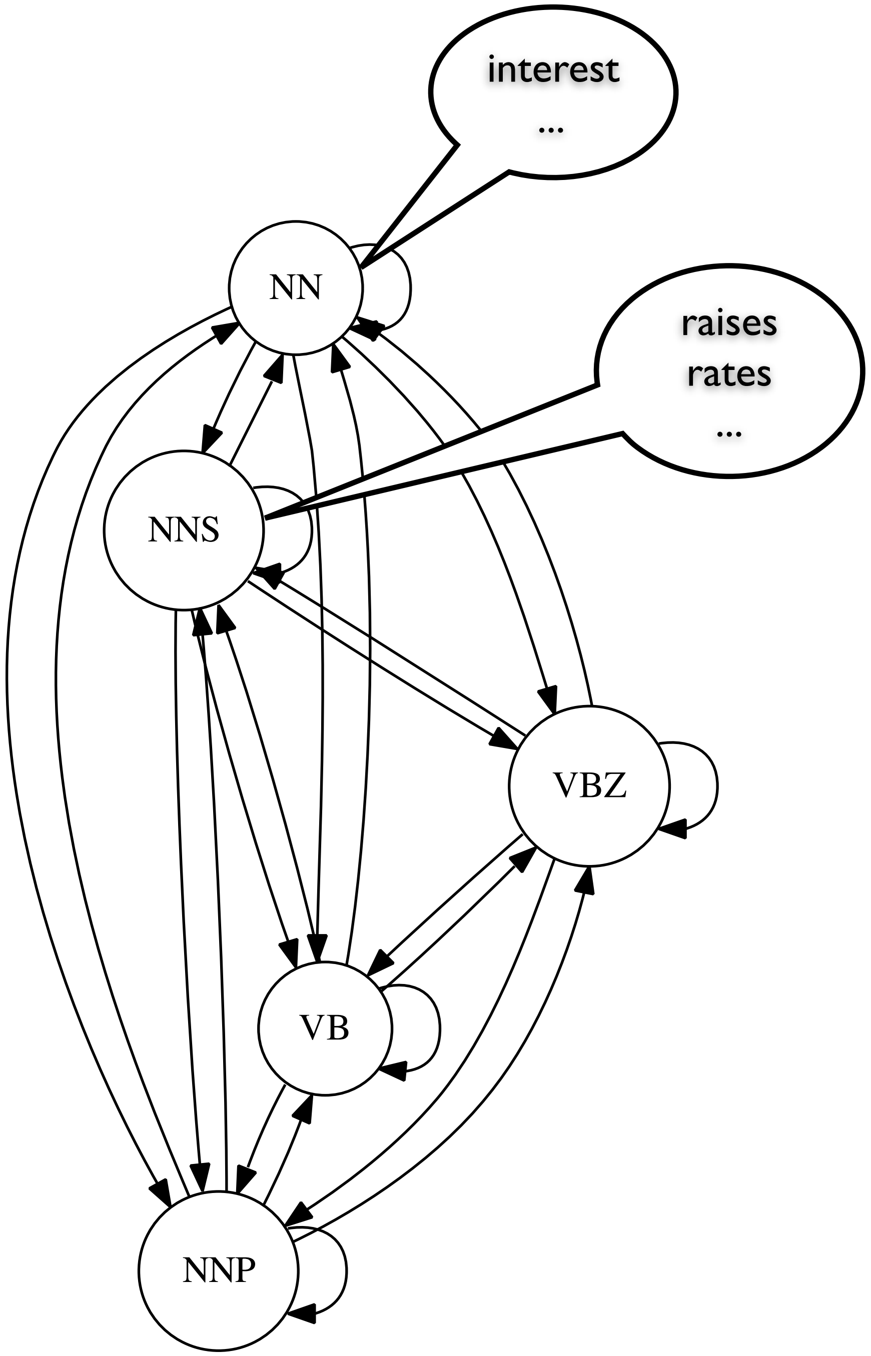
Markov vs. Hidden Markov Models



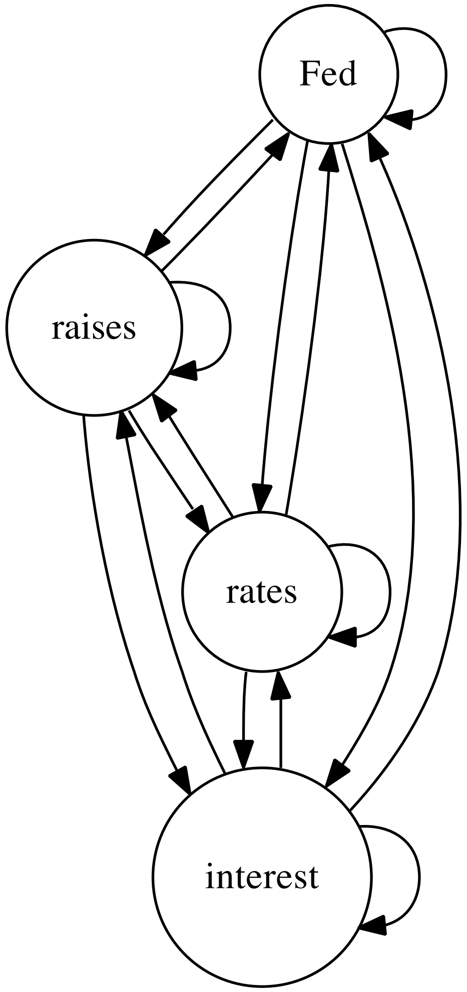


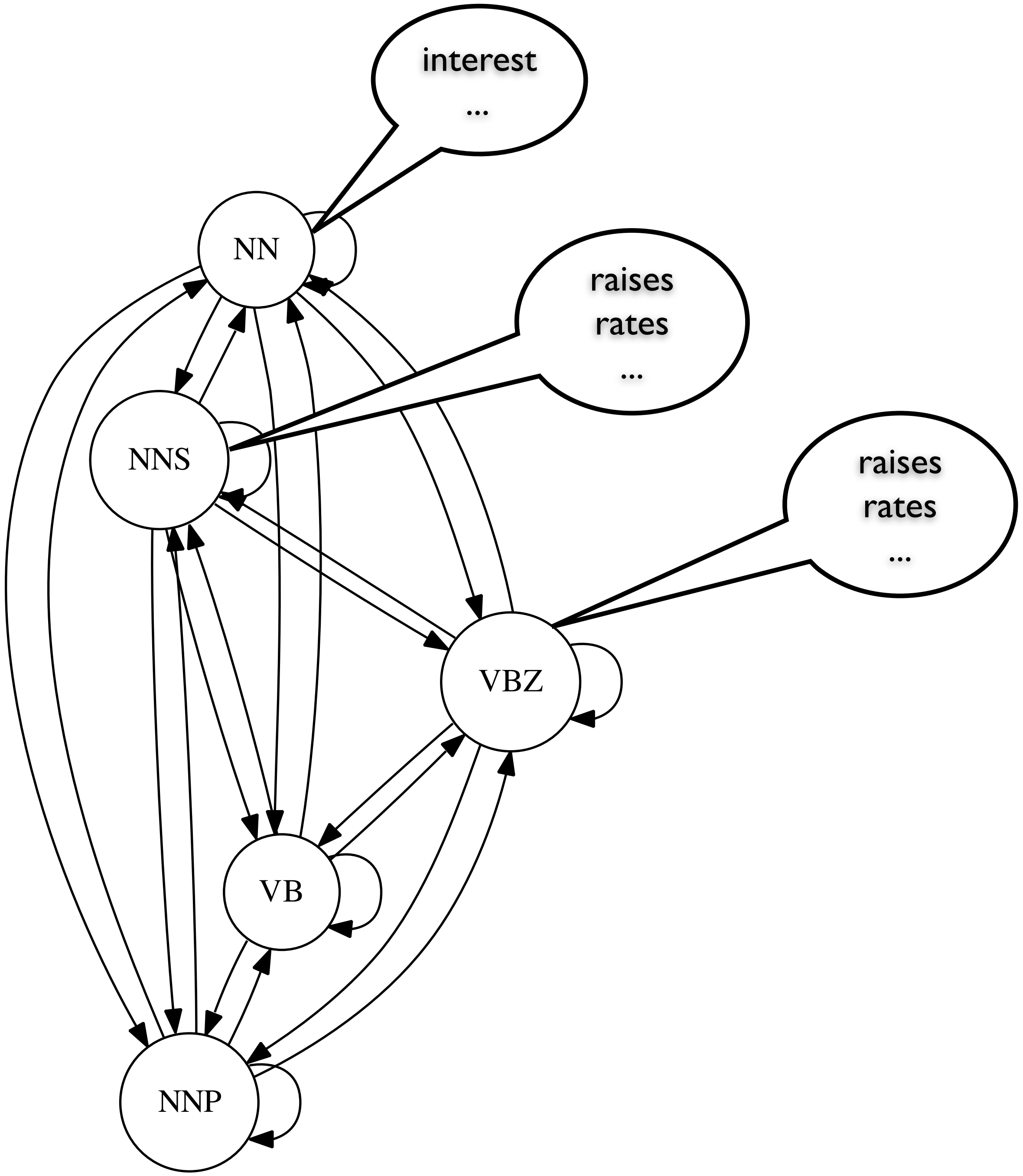
Markov vs. Hidden Markov Models



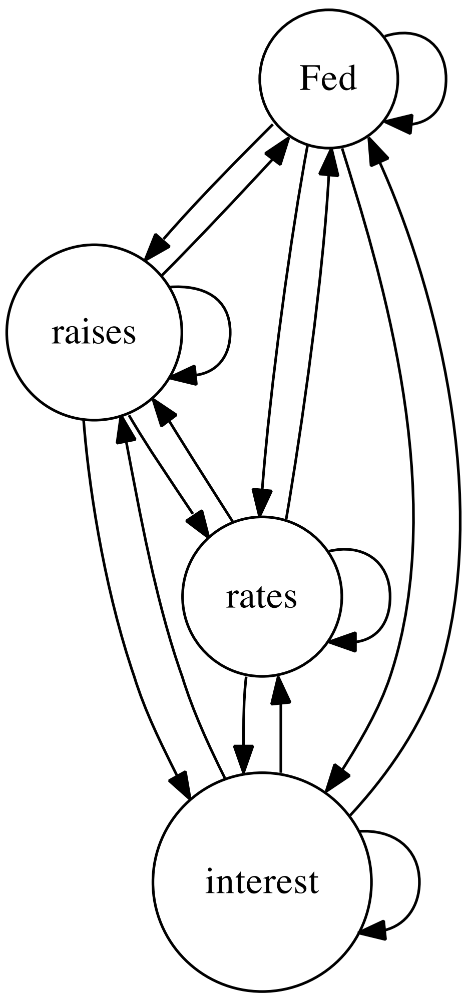


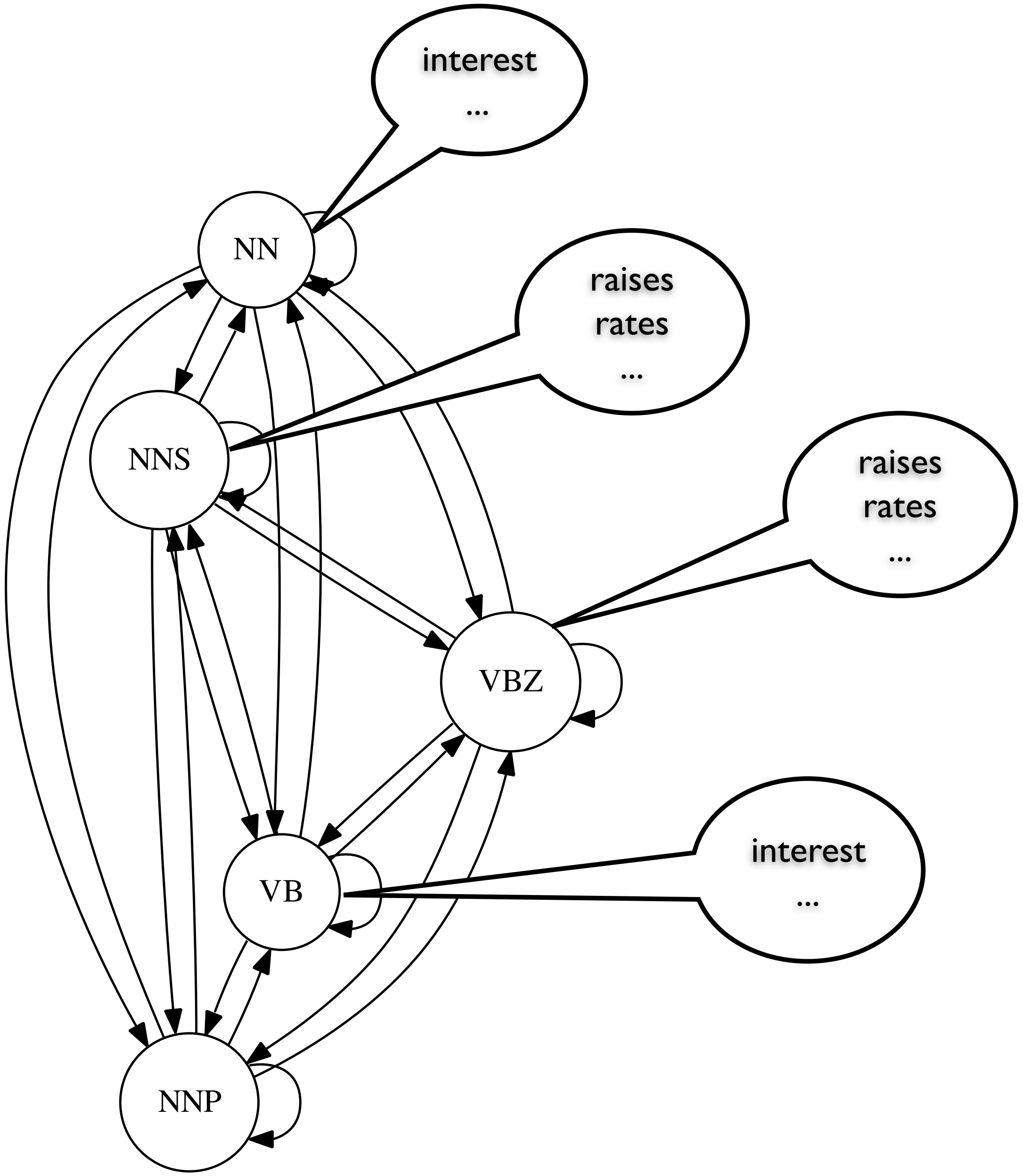
Markov vs. Hidden Markov Models



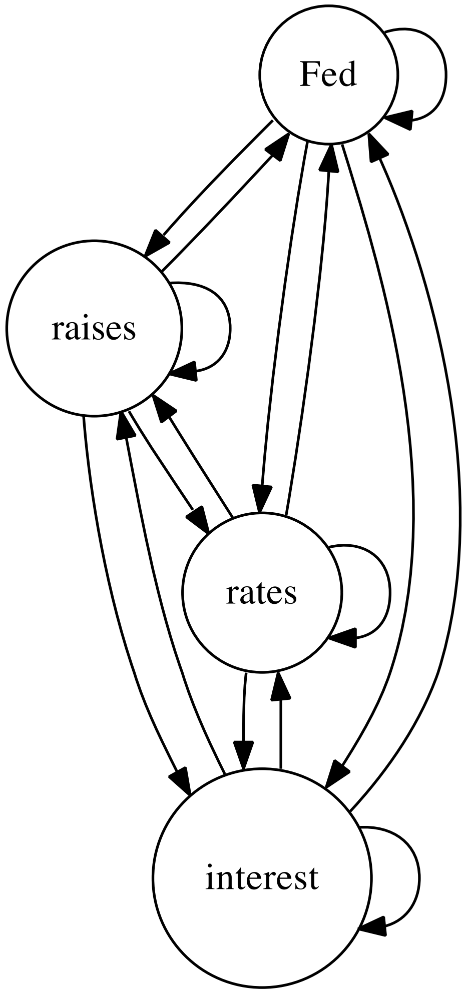


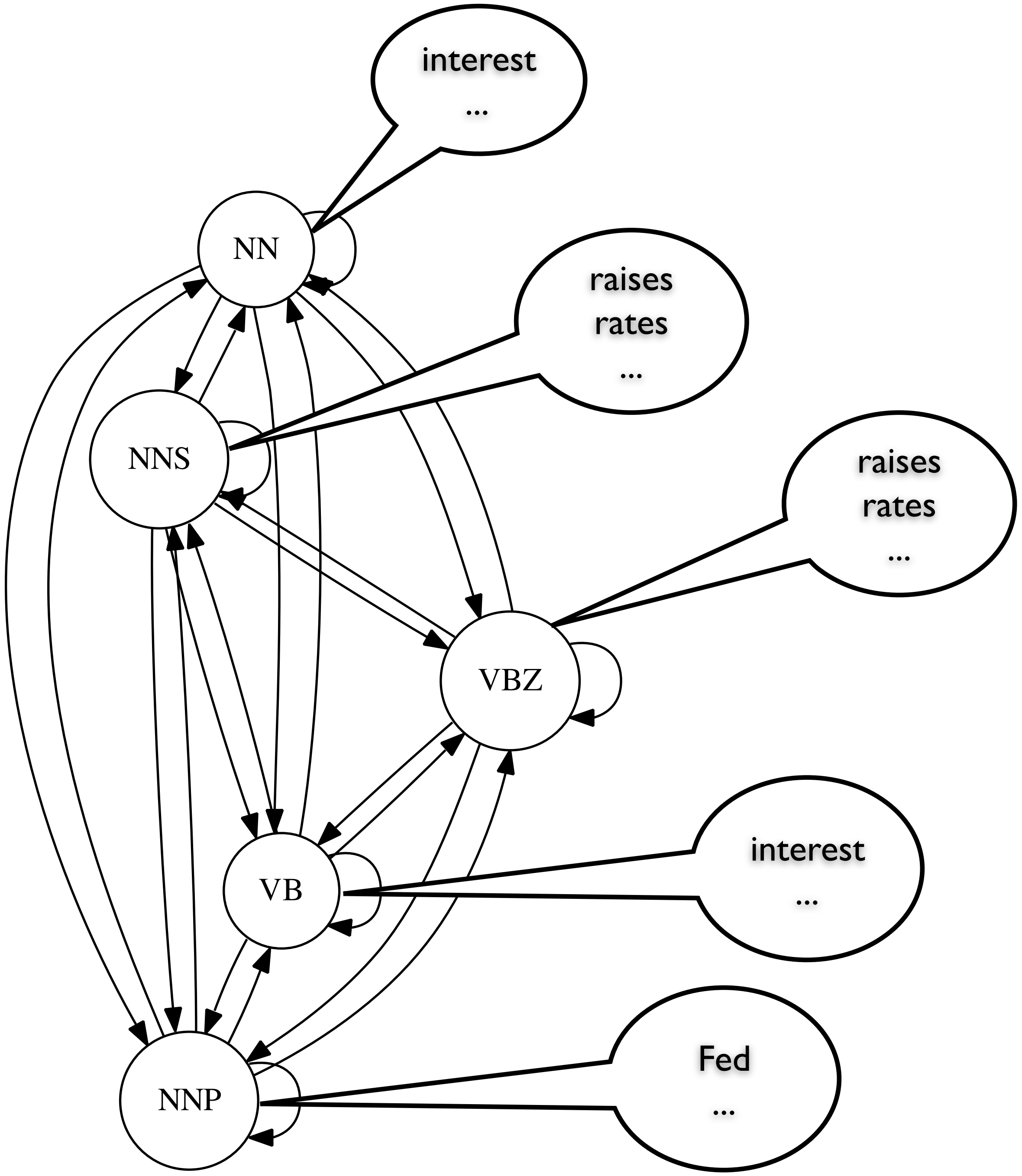
Markov vs. Hidden Markov Models



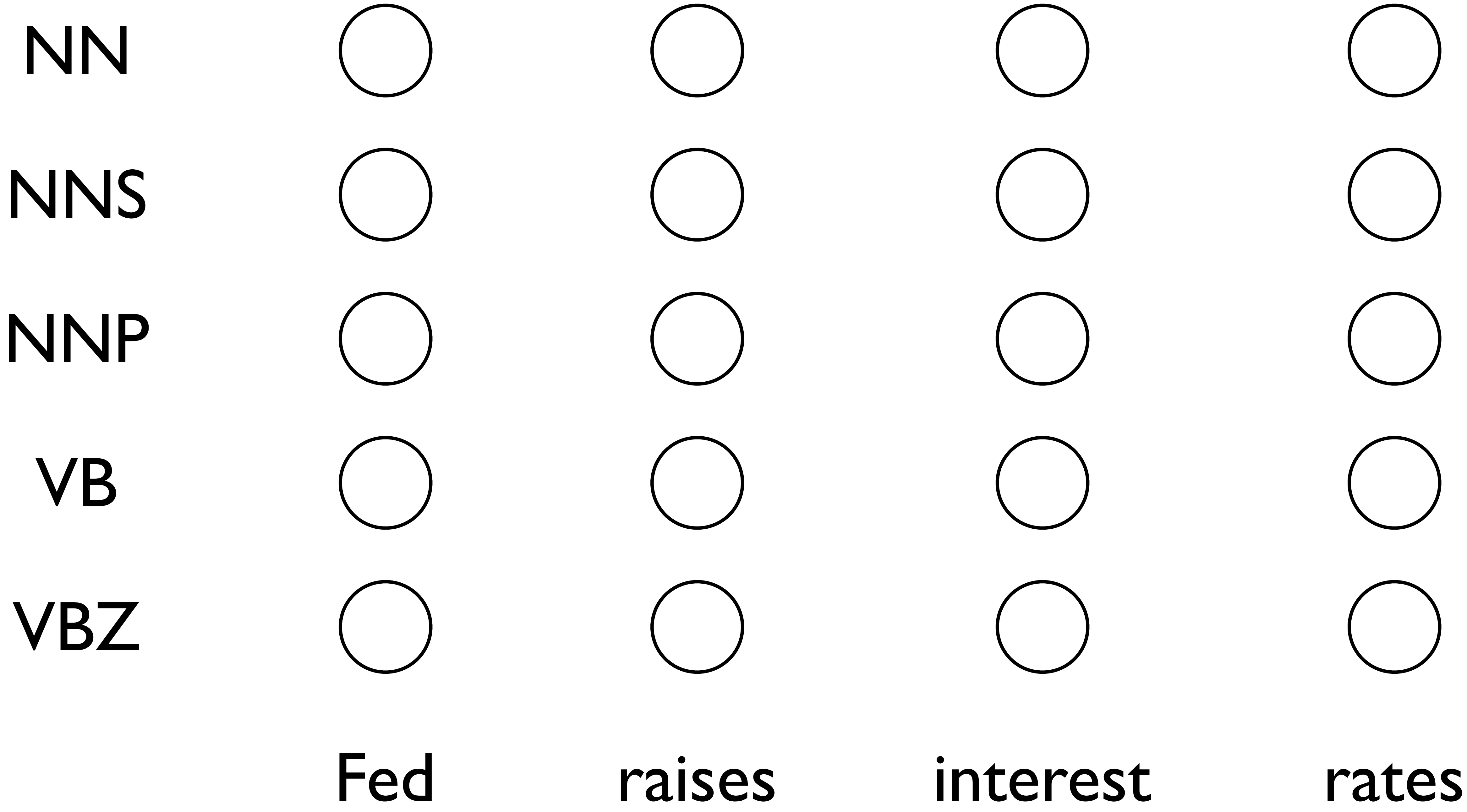


Markov vs. Hidden Markov Models





Unrolled into a Trellis



NN

NNS

NNP

VB

VBZ

Fed raises interest rates

HMM Inference Problems

- Given an observation sequence, find the most likely state sequence (tagging)
- Compute the probability of observations when state sequence is hidden (language modeling)
- Given observations and (optionally) a their corresponding states, find parameters that maximize the probability probability of the observations (parameter estimation)

Tagging

Given an observation sequence, find the most likely state sequence.

$$\arg\max_X P(X \mid O, \mu) = \arg\max_X \frac{P(X, O \mid \mu)}{P(O \mid \mu)} = \arg\max_X P(X, O \mid \mu)$$

$$\arg \max_{x_1, x_2, \ldots x_T} P(x_1, x_2, \ldots, x_T, O \mid \mu)$$

Last time: Use dynamic programming to find highest-probability sequence (i.e. best path, like Dijsktra's algorithm)

Language Modeling

Compute the probability of observations when state sequence is hidden.

$$P(X, O \mid \mu) = P(O \mid X, \mu) P(X \mid \mu)$$

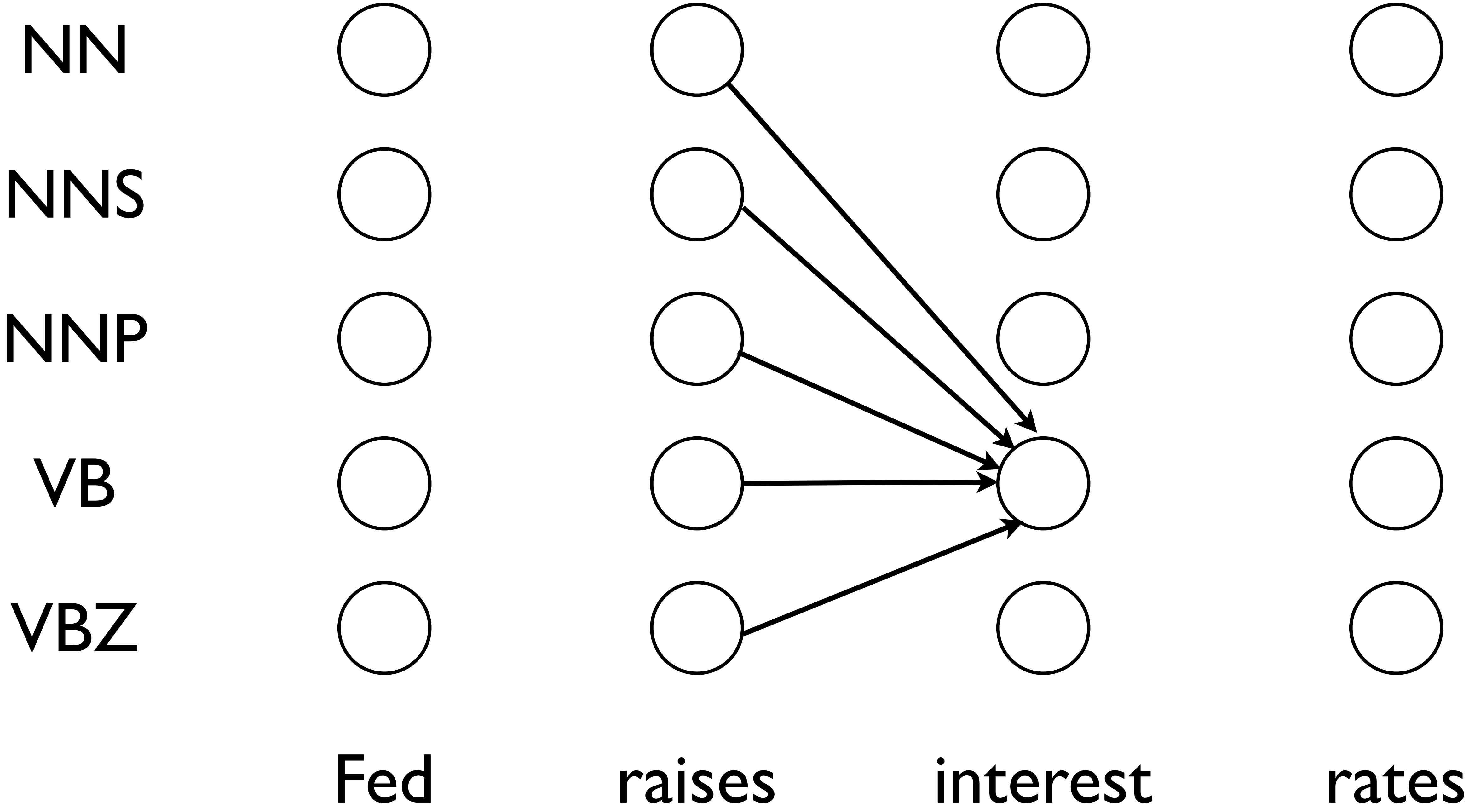
Therefore

$$P(O \mid \mu) = \sum_{X} P(O \mid X, \mu) P(X \mid \mu)$$

$$\sum_{x_1, x_2, \ldots x_T} P(x_1, x_2, \ldots, x_T, O \mid \mu)$$

Suspiciously similar to $\max_{x_1, x_2, \ldots x_T} P(x_1, x_2, \ldots, x_T, O \mid \mu)$

Viterbi Algorithm (Tagging)



NN

NNS

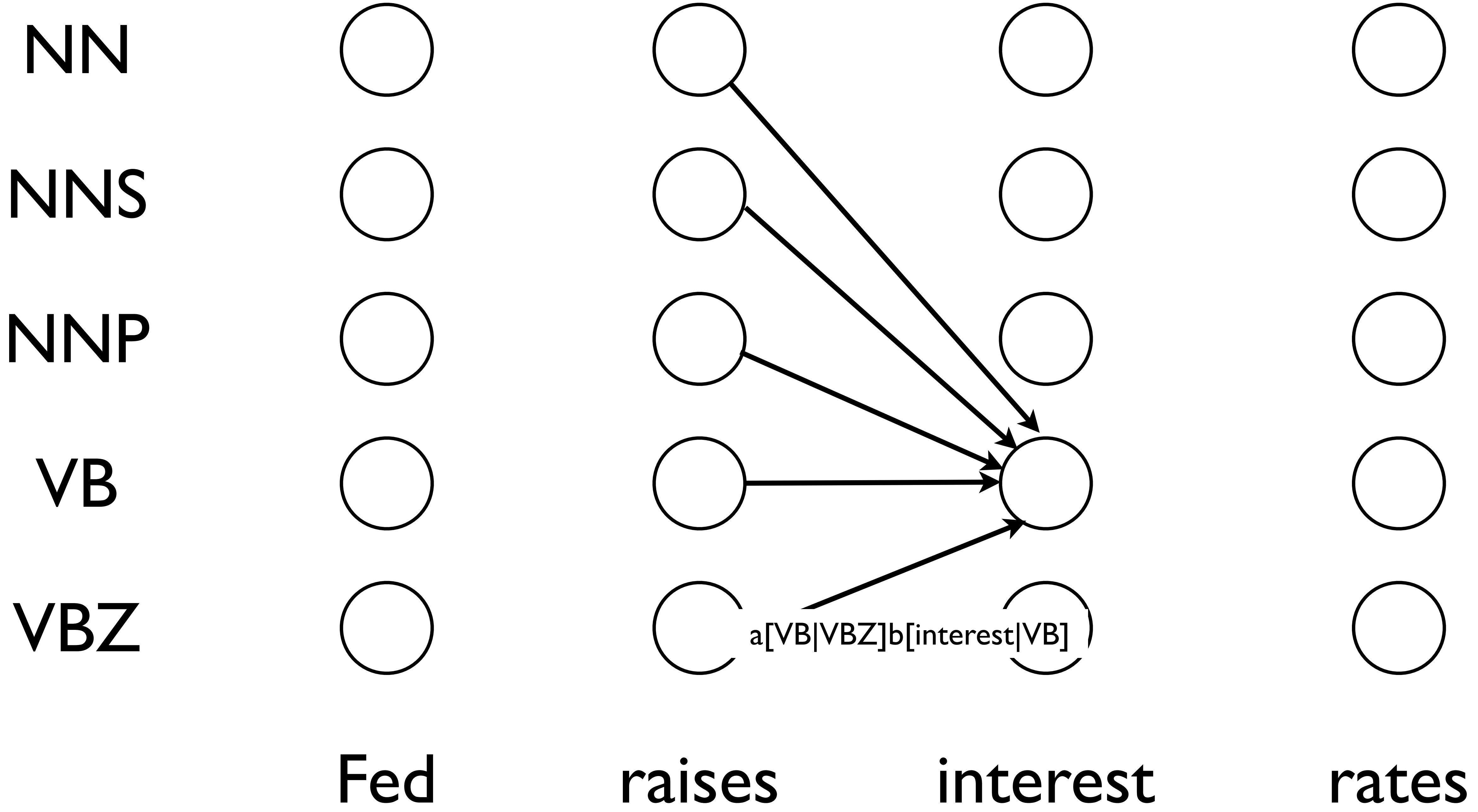
NNP

VB

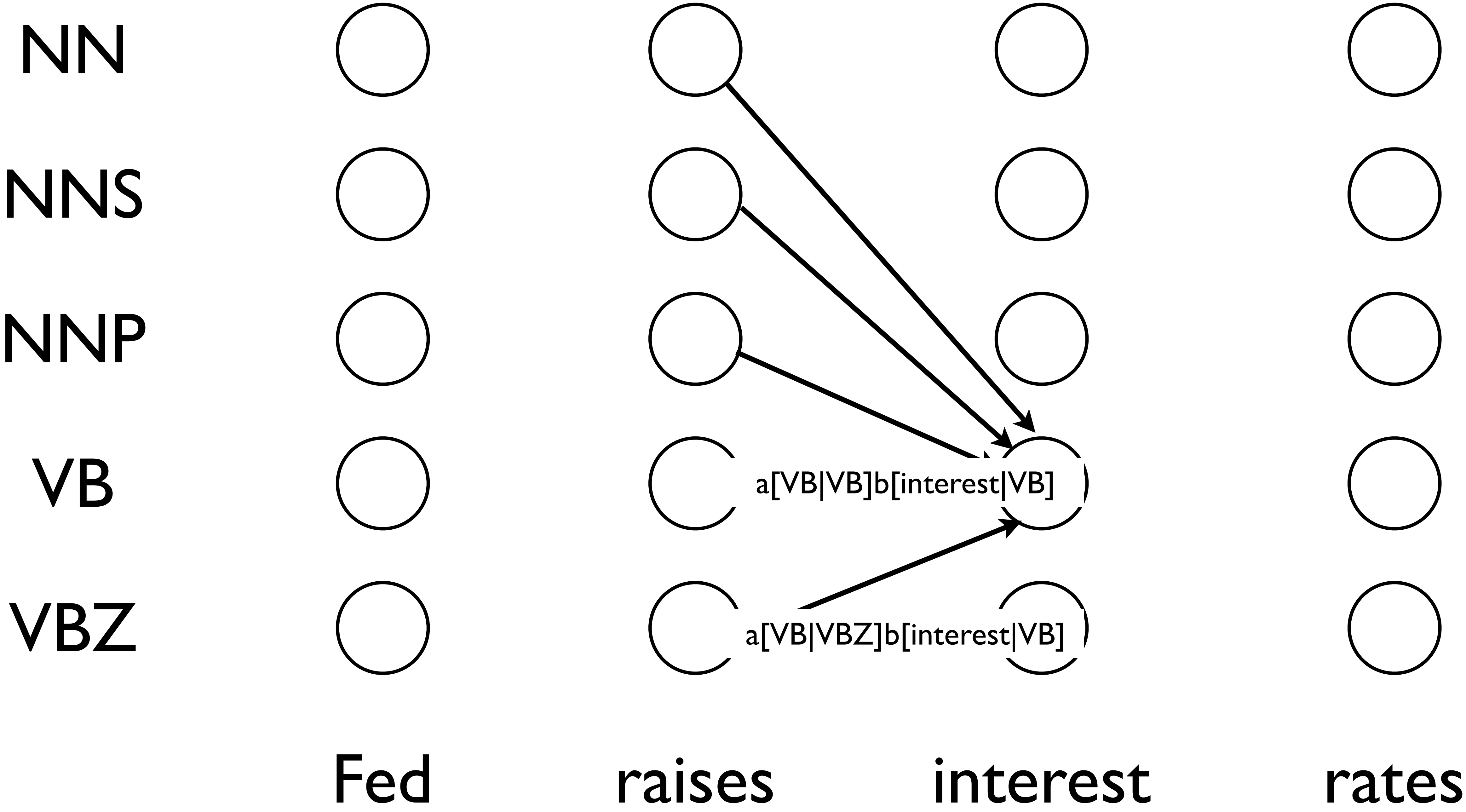
VBZ

Fed raises interest rates

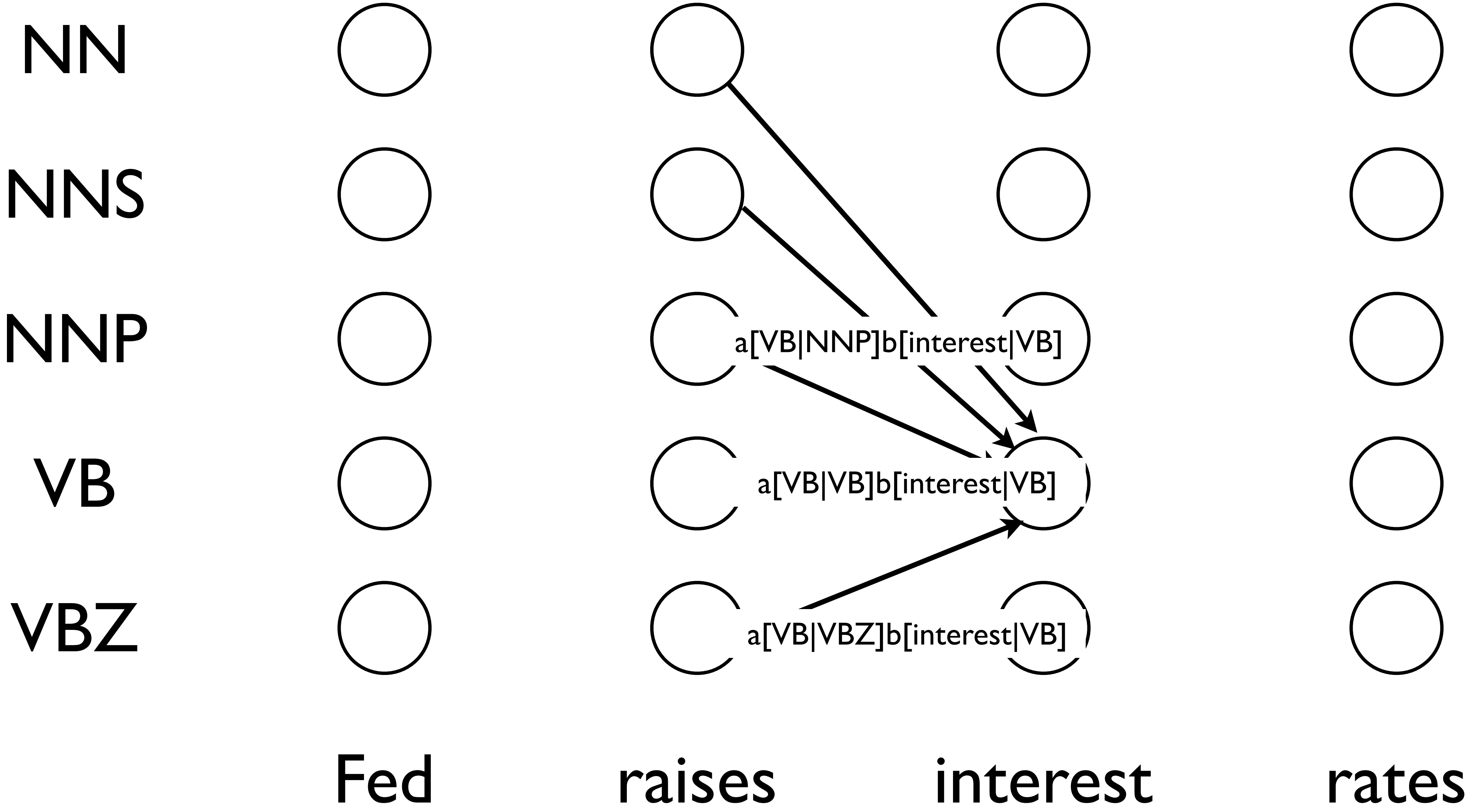
Viterbi Algorithm (Tagging)



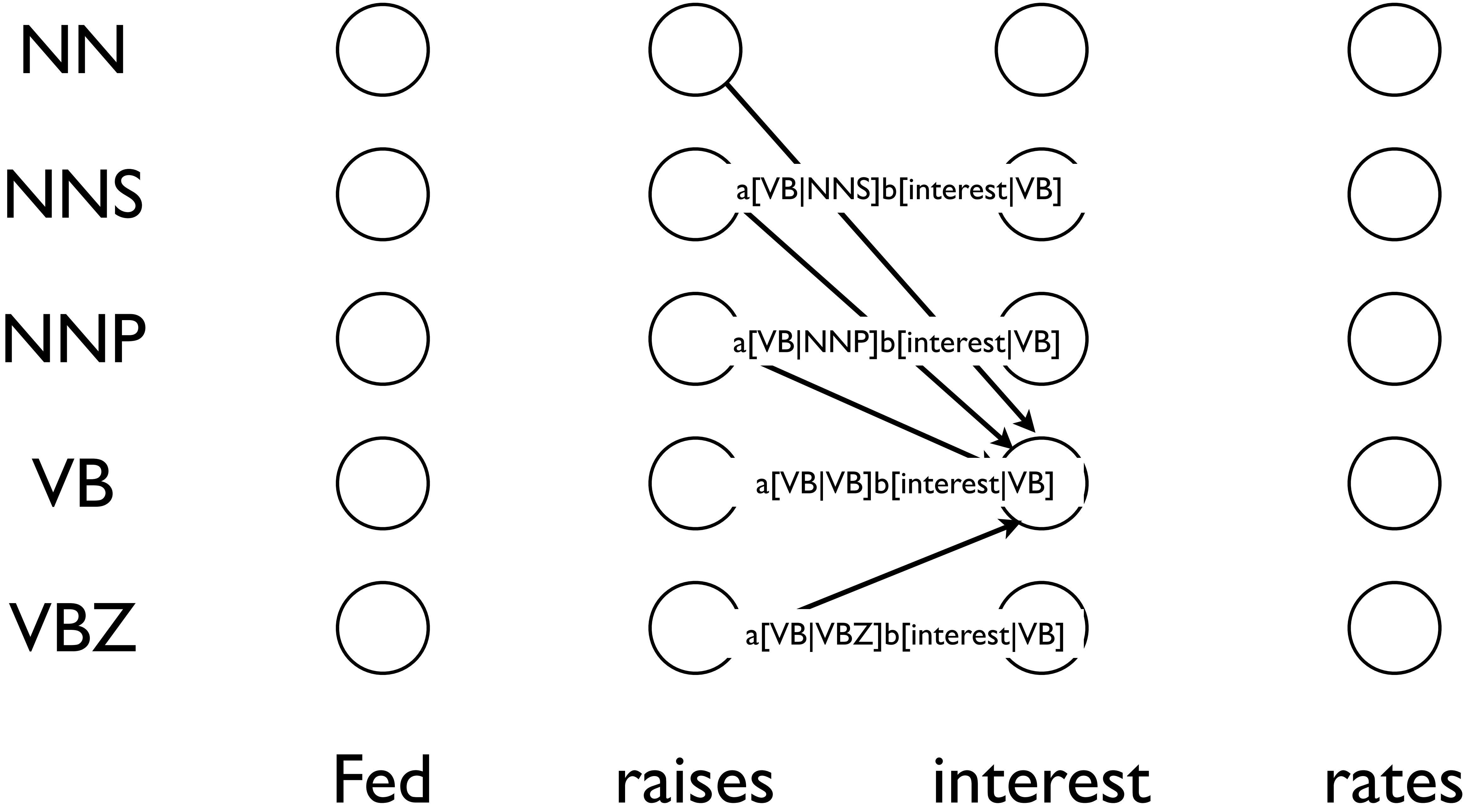
Viterbi Algorithm (Tagging)



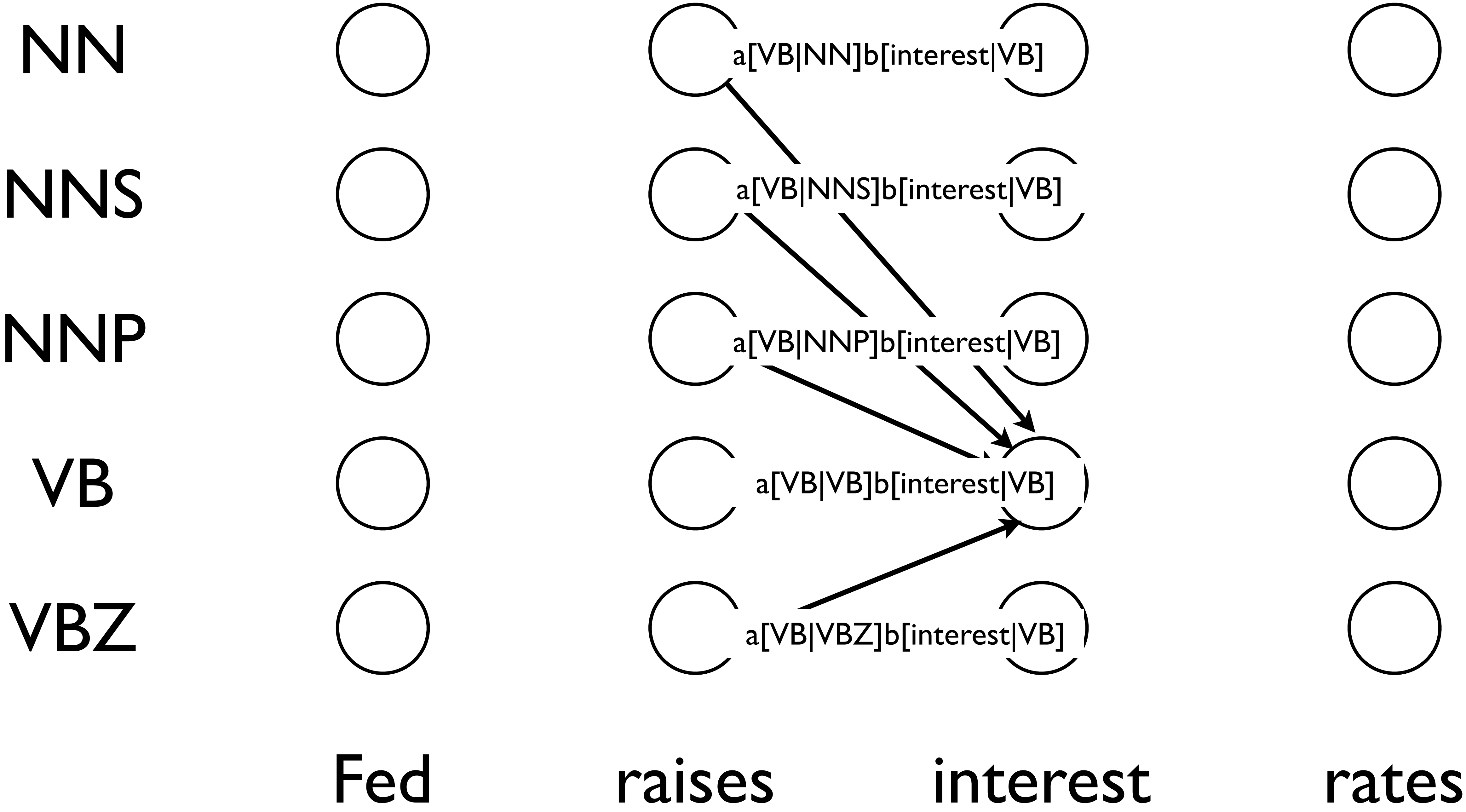
Viterbi Algorithm (Tagging)



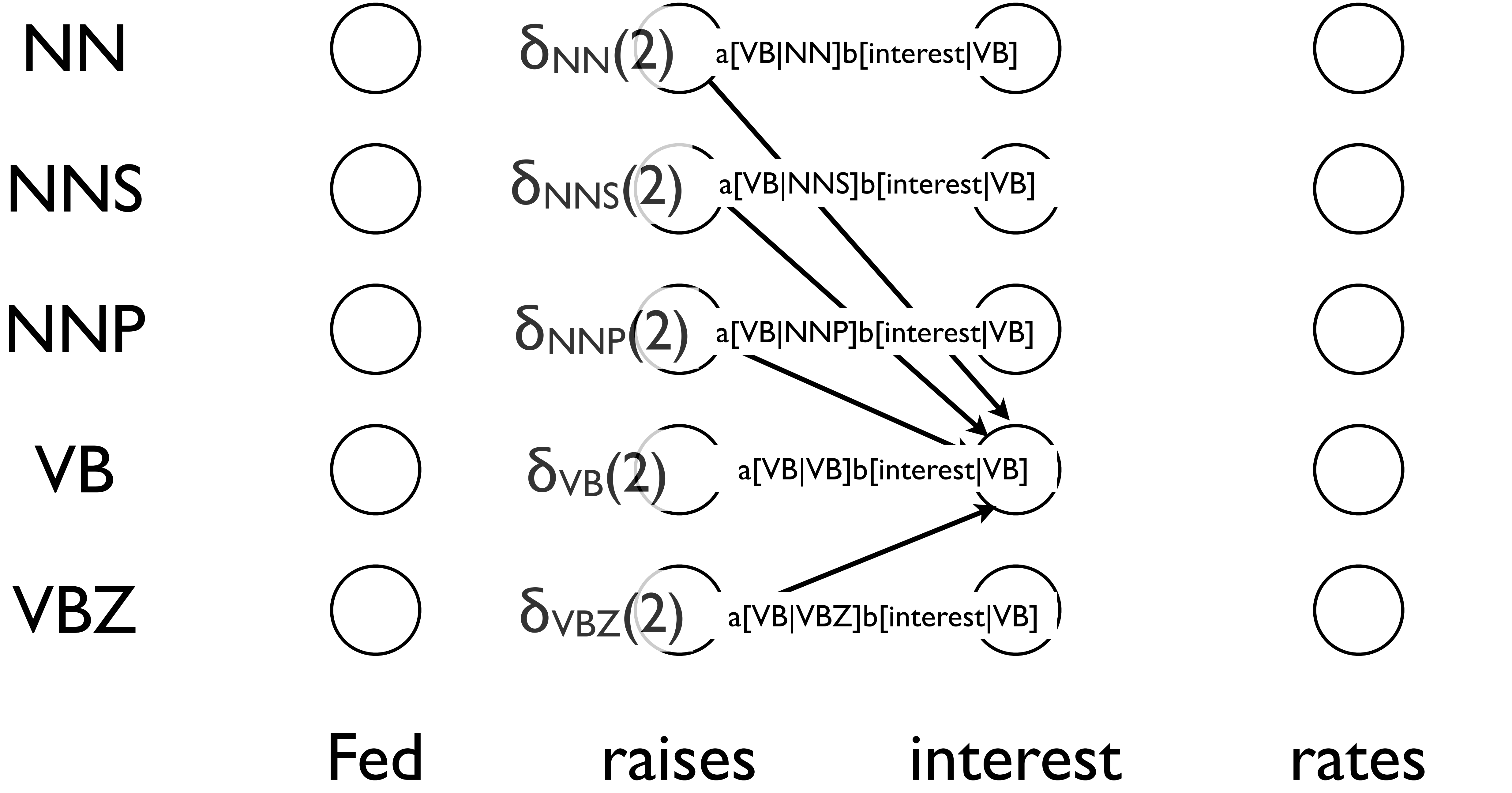
Viterbi Algorithm (Tagging)



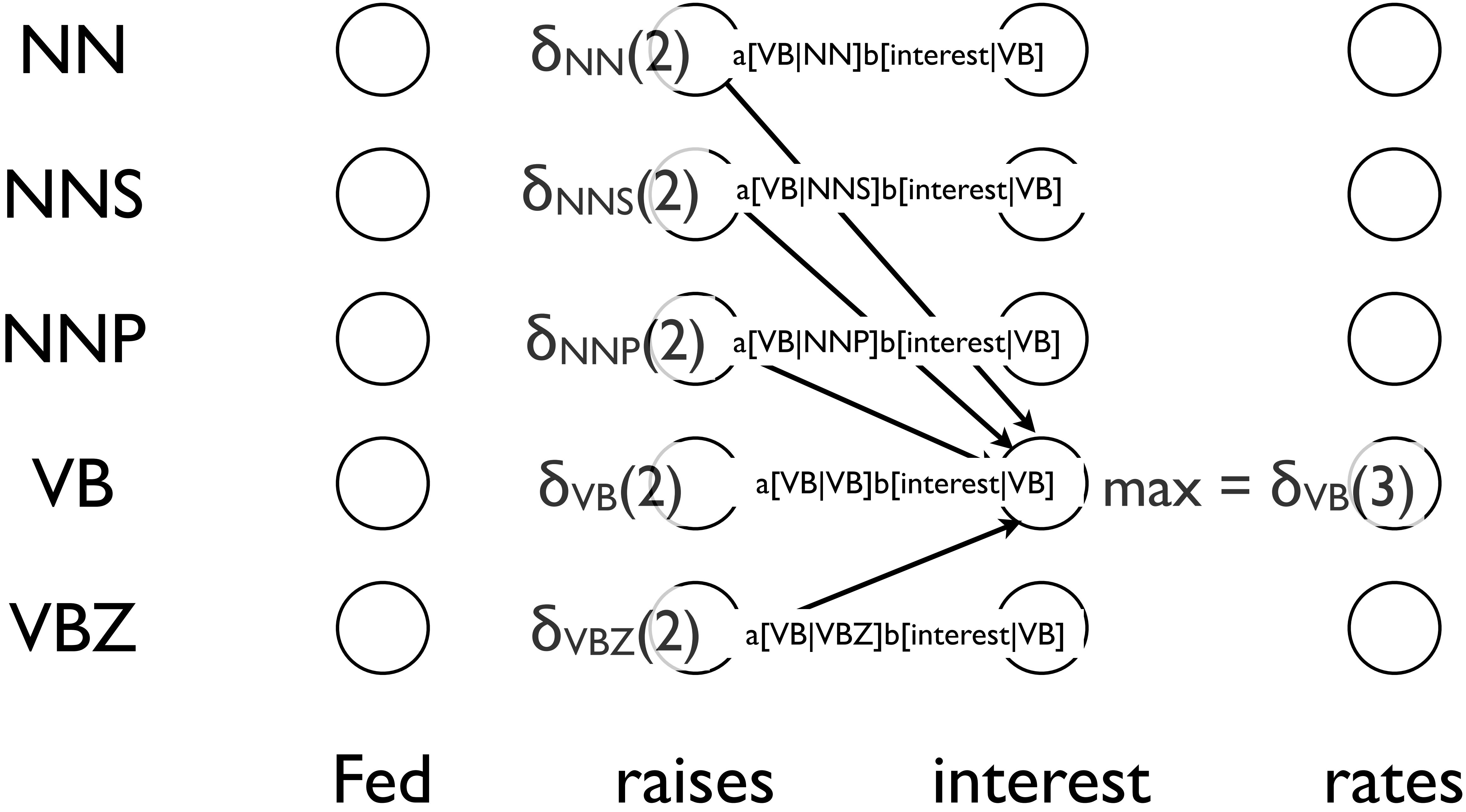
Viterbi Algorithm (Tagging)



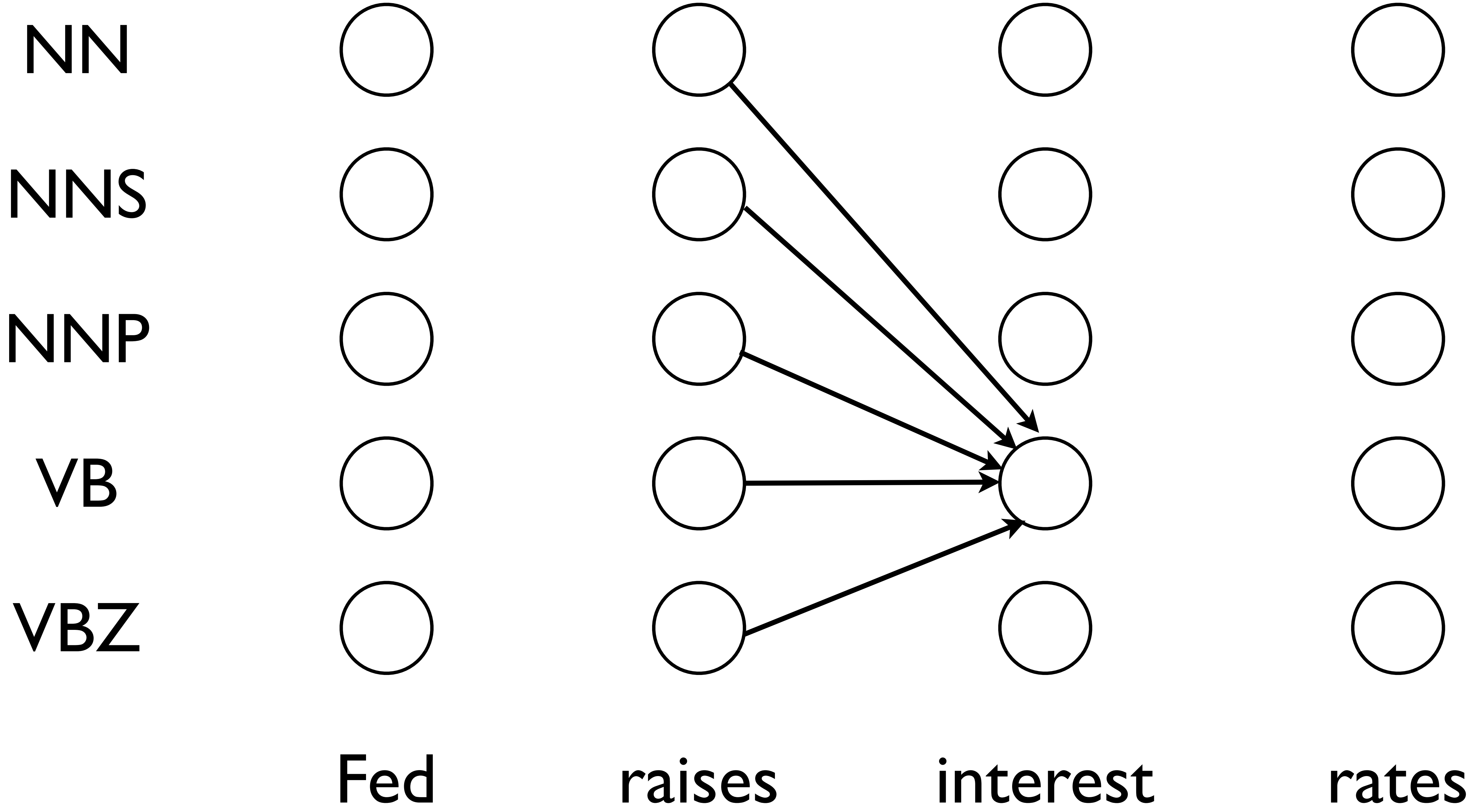
Viterbi Algorithm (Tagging)



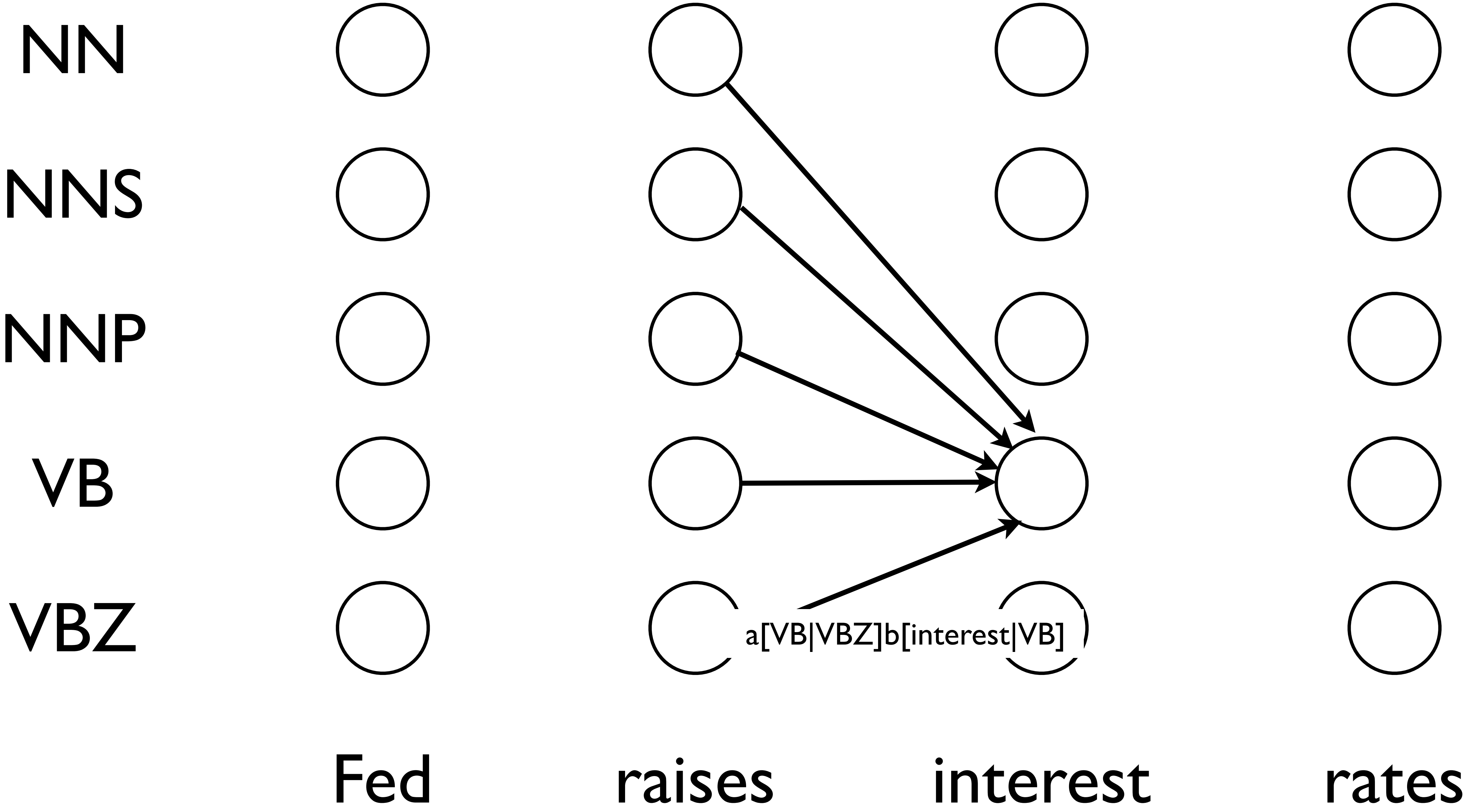
Viterbi Algorithm (Tagging)



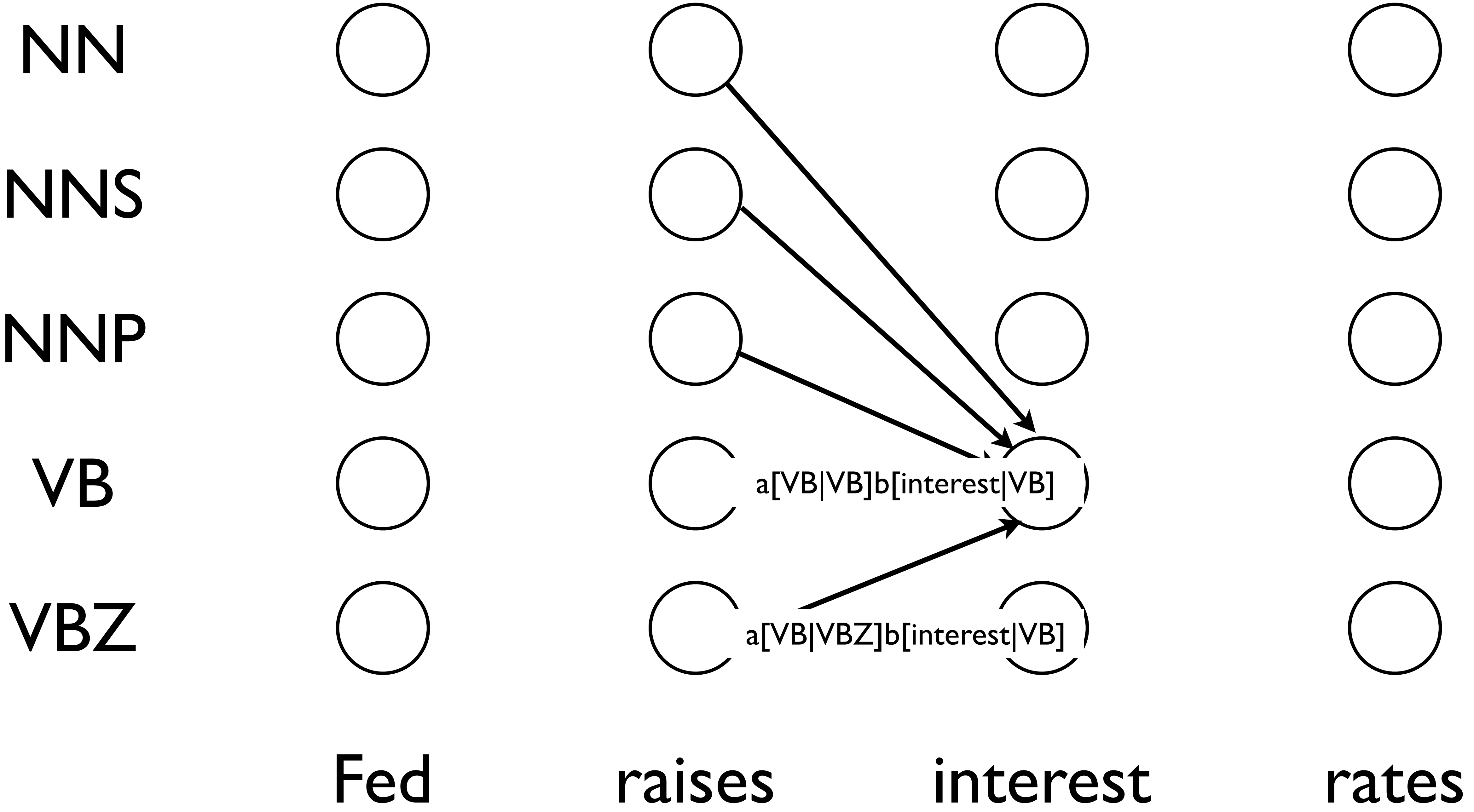
Forward Algorithm (LM)



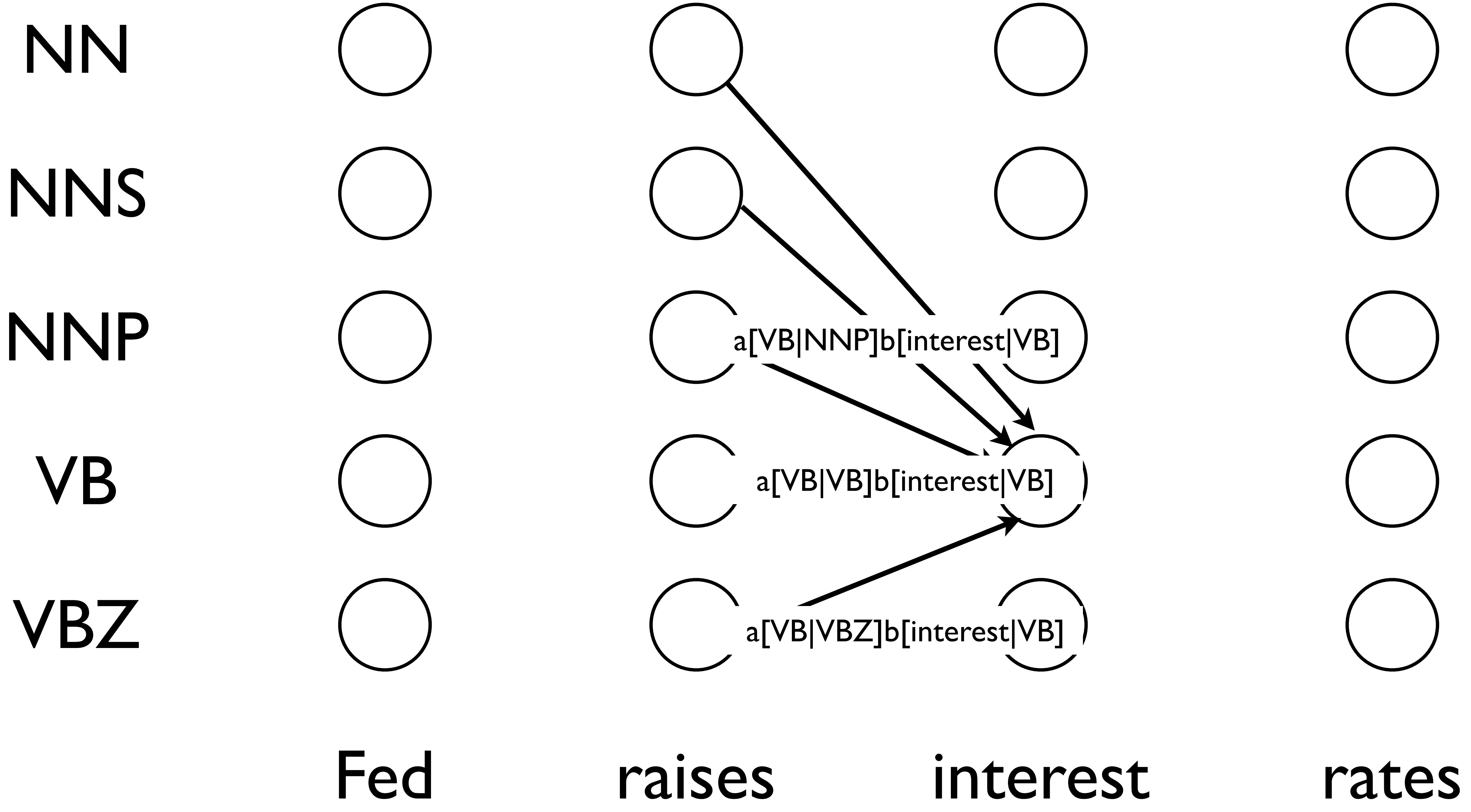
Forward Algorithm (LM)



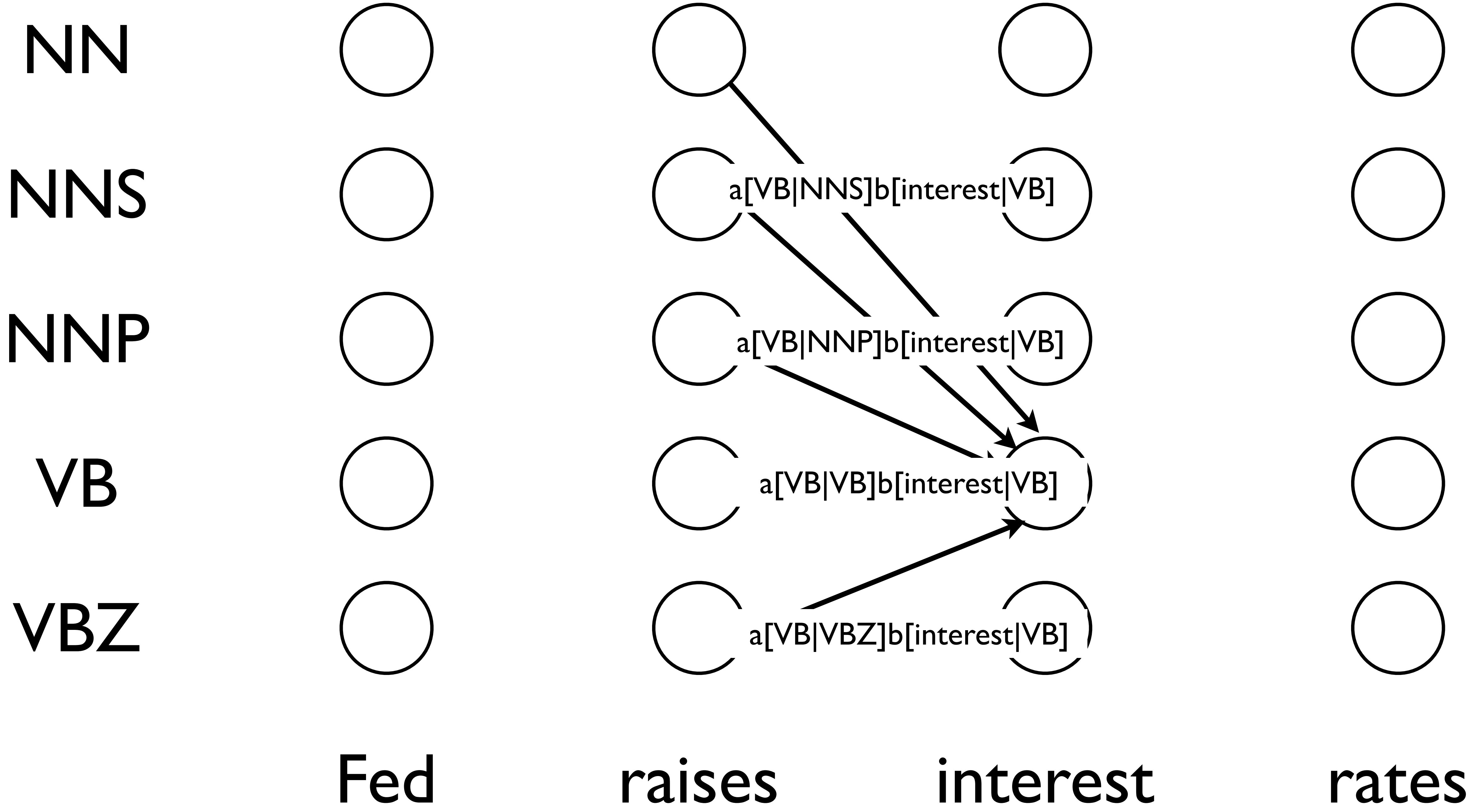
Forward Algorithm (LM)



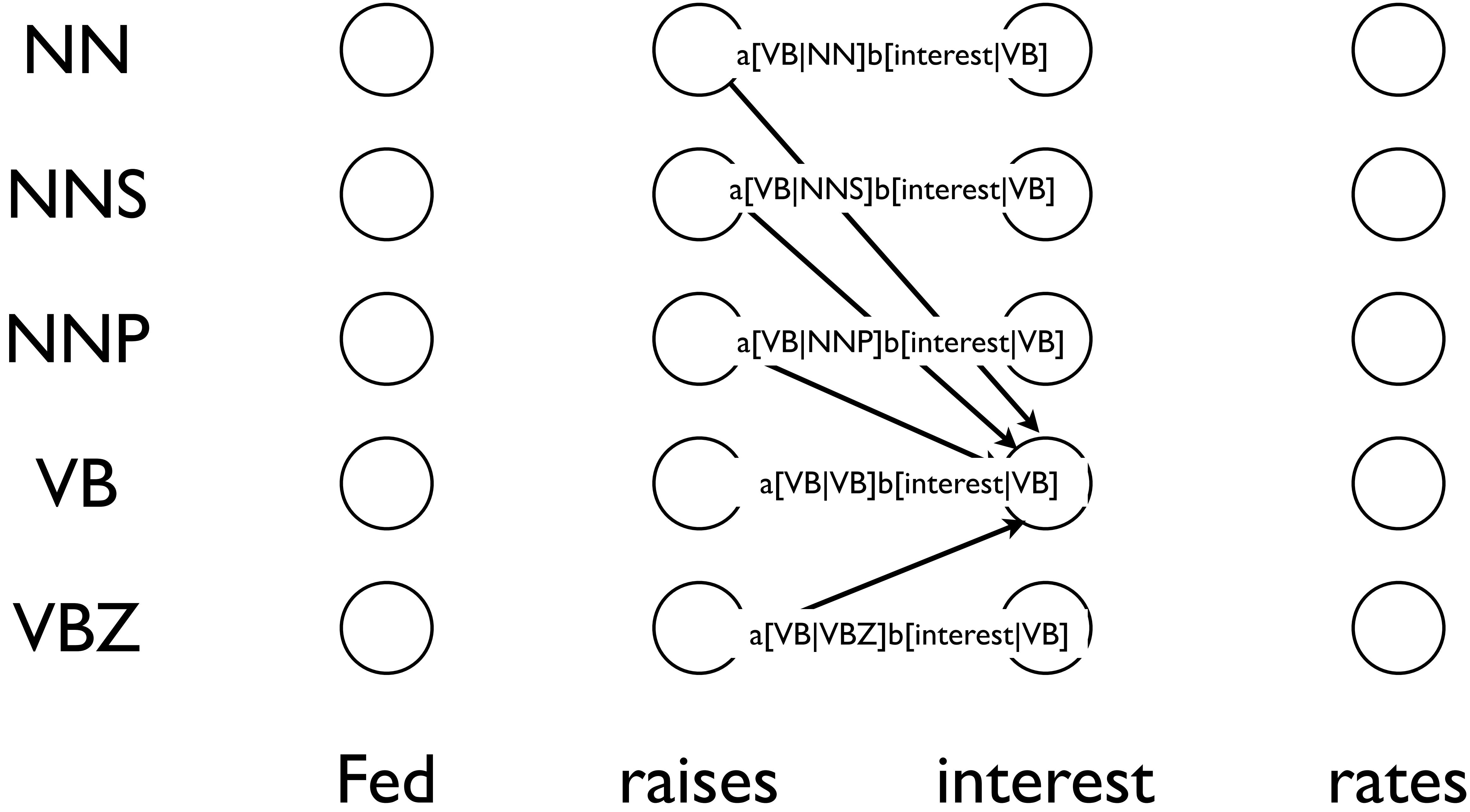
Forward Algorithm (LM)



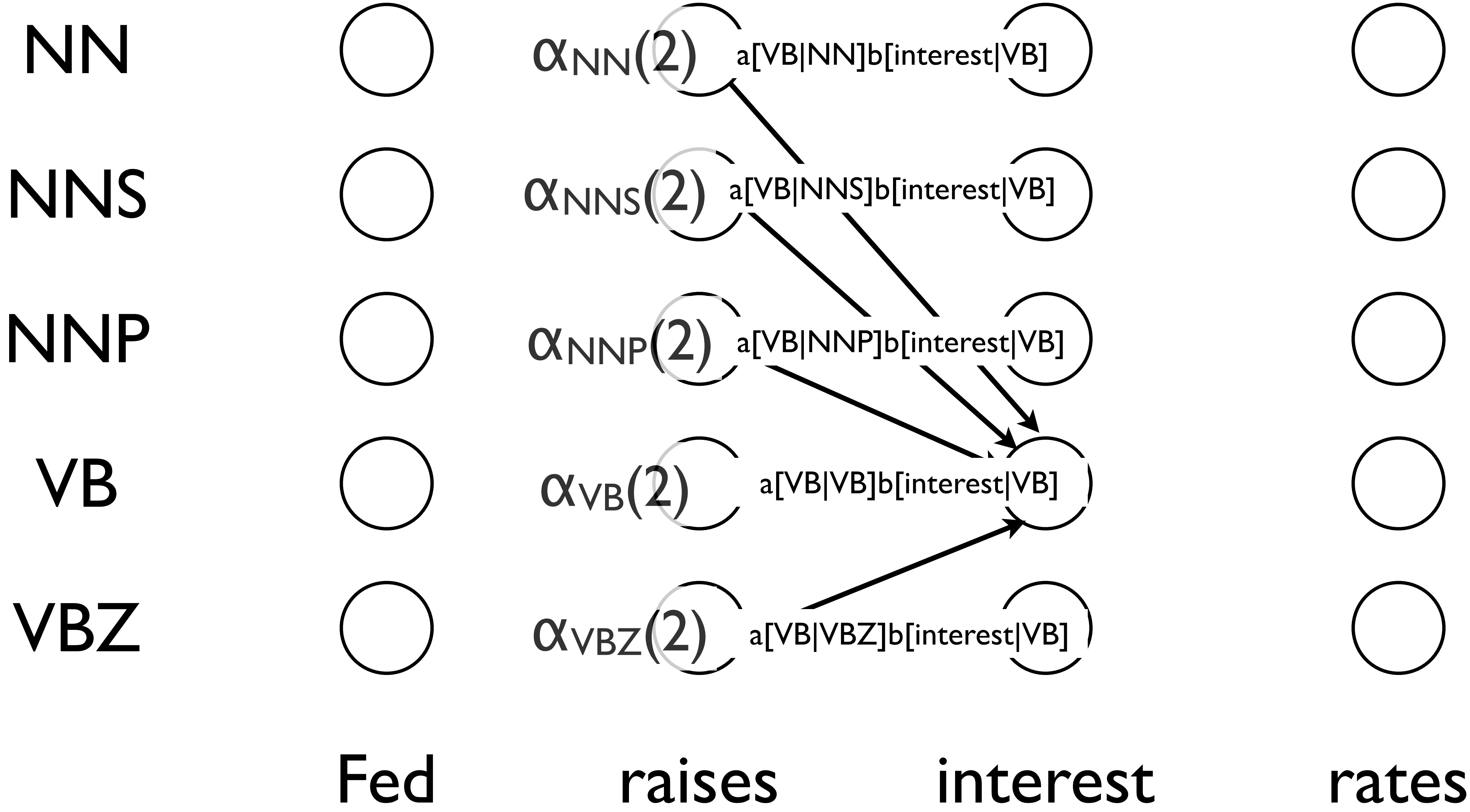
Forward Algorithm (LM)



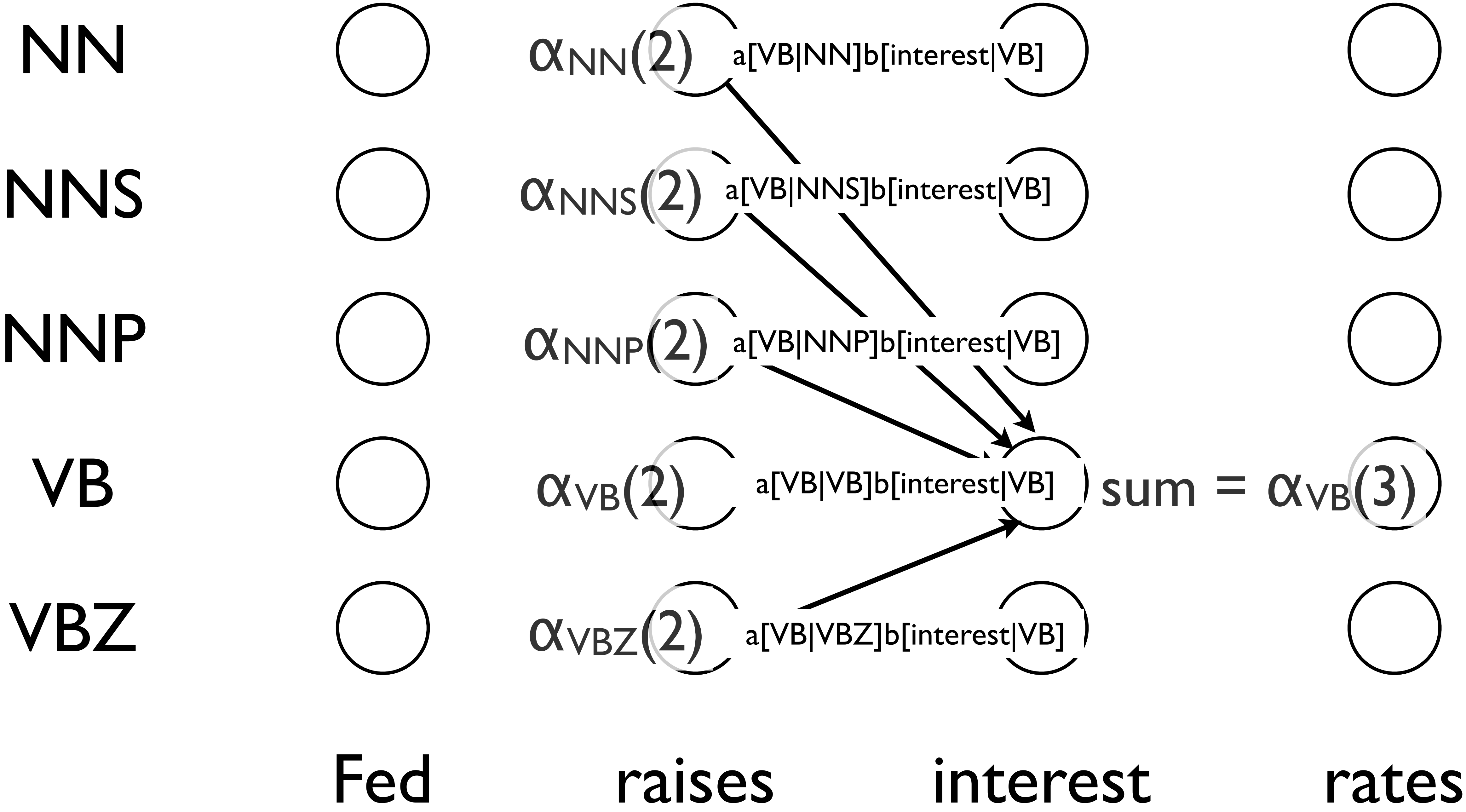
Forward Algorithm (LM)



Forward Algorithm (LM)



Forward Algorithm (LM)



What Do These Greek Letters Mean?

$$\delta_j(t) = \max_{x_1 \cdots x_{t-1}} P(x_1 \cdots x_{t-1}, o_1 \cdots o_{t-1}, x_t = j \mid \mu)$$

$$\begin{aligned}\alpha_j(t) &= \sum_{x_1 \cdots x_{t-1}} P(x_1 \cdots x_{t-1}, o_1 \cdots o_{t-1}, x_t = j \mid \mu) \\ &= P(o_1 \cdots o_{t-1}, x_t = j \mid \mu)\end{aligned}$$

What Do These Greek Letters Mean?

Probability of the best path from the beginning to word *t* such that word *t* has tag *j*

$$\delta_j(t) = \max_{x_1 \cdots x_{t-1}} P(x_1 \cdots x_{t-1}, o_1 \cdots o_{t-1}, x_t = j \mid \mu)$$

$$\begin{aligned}\alpha_j(t) &= \sum_{x_1 \cdots x_{t-1}} P(x_1 \cdots x_{t-1}, o_1 \cdots o_{t-1}, x_t = j \mid \mu) \\ &= P(o_1 \cdots o_{t-1}, x_t = j \mid \mu)\end{aligned}$$

What Do These Greek Letters Mean?

Probability of the best path from the beginning to word *t* such that word *t* has tag *j*

$$\delta_j(t) = \max_{x_1 \cdots x_{t-1}} P(x_1 \cdots x_{t-1}, o_1 \cdots o_{t-1}, x_t = j \mid \mu)$$

Probability of all paths from the beginning to word *t* such that word *t* has tag *j*

$$\begin{aligned} \alpha_j(t) &= \sum_{x_1 \cdots x_{t-1}} P(x_1 \cdots x_{t-1}, o_1 \cdots o_{t-1}, x_t = j \mid \mu) \\ &= P(o_1 \cdots o_{t-1}, x_t = j \mid \mu) \end{aligned}$$

What Do These Greek Letters Mean?

Probability of the best path from the beginning to word *t* such that word *t* has tag *j*

$$\delta_j(t) = \max_{x_1 \cdots x_{t-1}} P(x_1 \cdots x_{t-1}, o_1 \cdots o_{t-1}, x_t = j \mid \mu)$$

Probability of all paths from the beginning to word *t* such that word *t* has tag *j*

$$\begin{aligned} \alpha_j(t) &= \sum_{x_1 \cdots x_{t-1}} P(x_1 \cdots x_{t-1}, o_1 \cdots o_{t-1}, x_t = j \mid \mu) \\ &= P(o_1 \cdots o_{t-1}, x_t = j \mid \mu) \end{aligned}$$

NOT
the probability of tag *j*
at time *t*

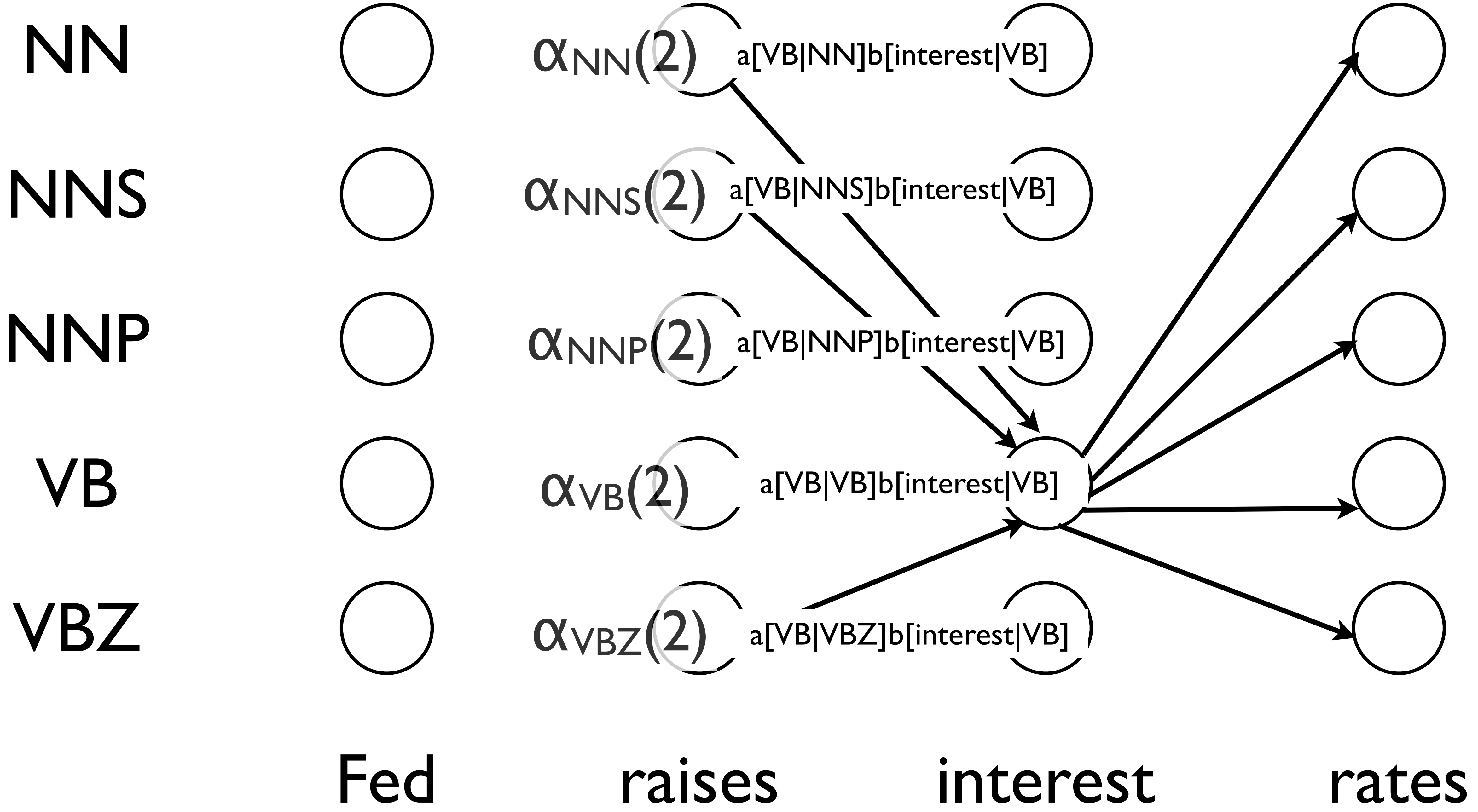
HMM Language Modeling

- Probability of observations, summed over all possible ways of tagging that observation: $$\sum_i \alpha_i(T)$$
- This is the sum of all path probabilities in the trellis

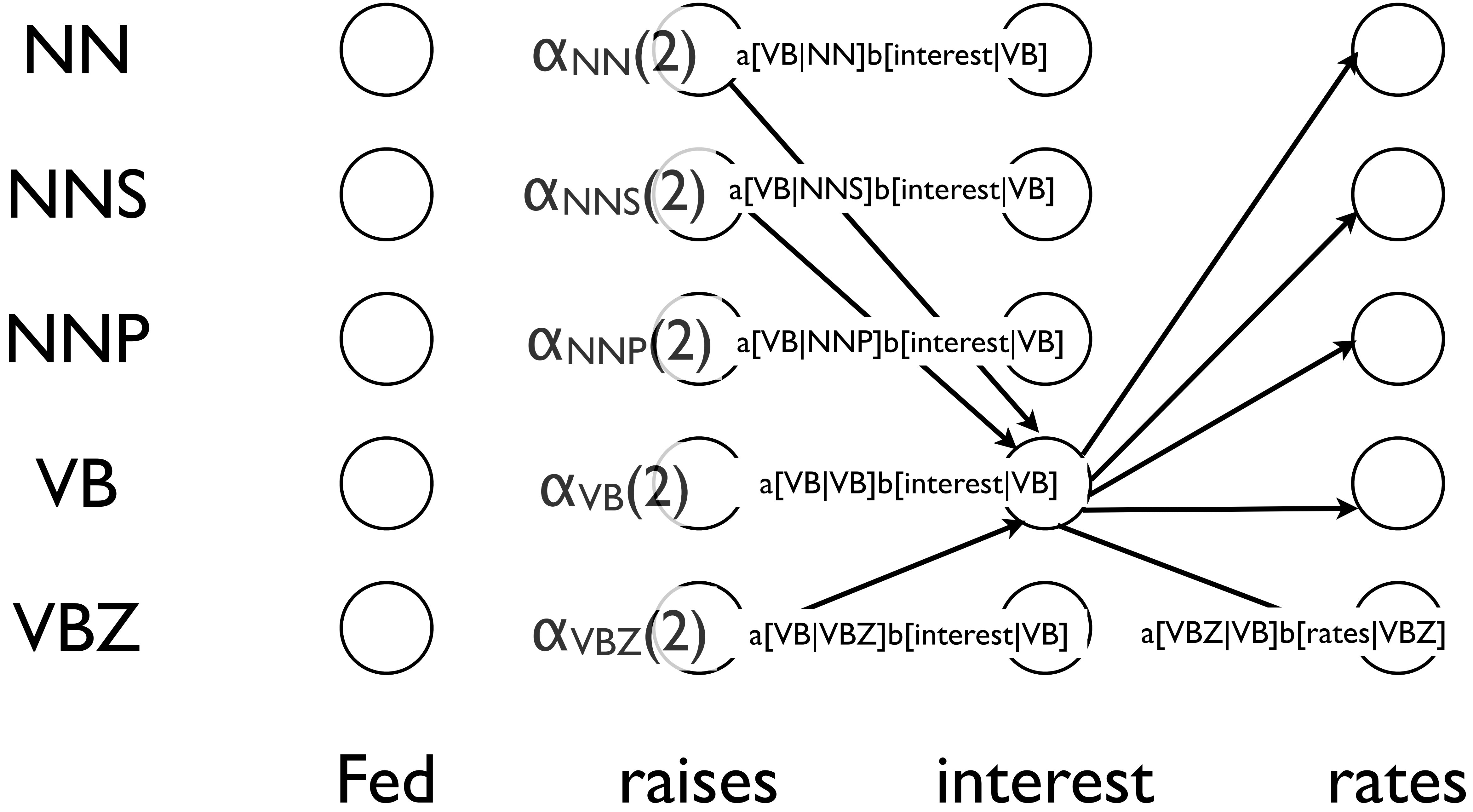
HMM Parameter Estimation

- Supervised
  - Train on tagged text, test on plain text
  - Maximum likelihood (can be smoothed):
    - a[VBZ | NN] = C(NN,VBZ) / C(NN)
    - b[rates |VBZ] = C(VBZ,rates) / C(VBZ)
- Unsupervised
  - Train and test on plain text
  - What can we do?

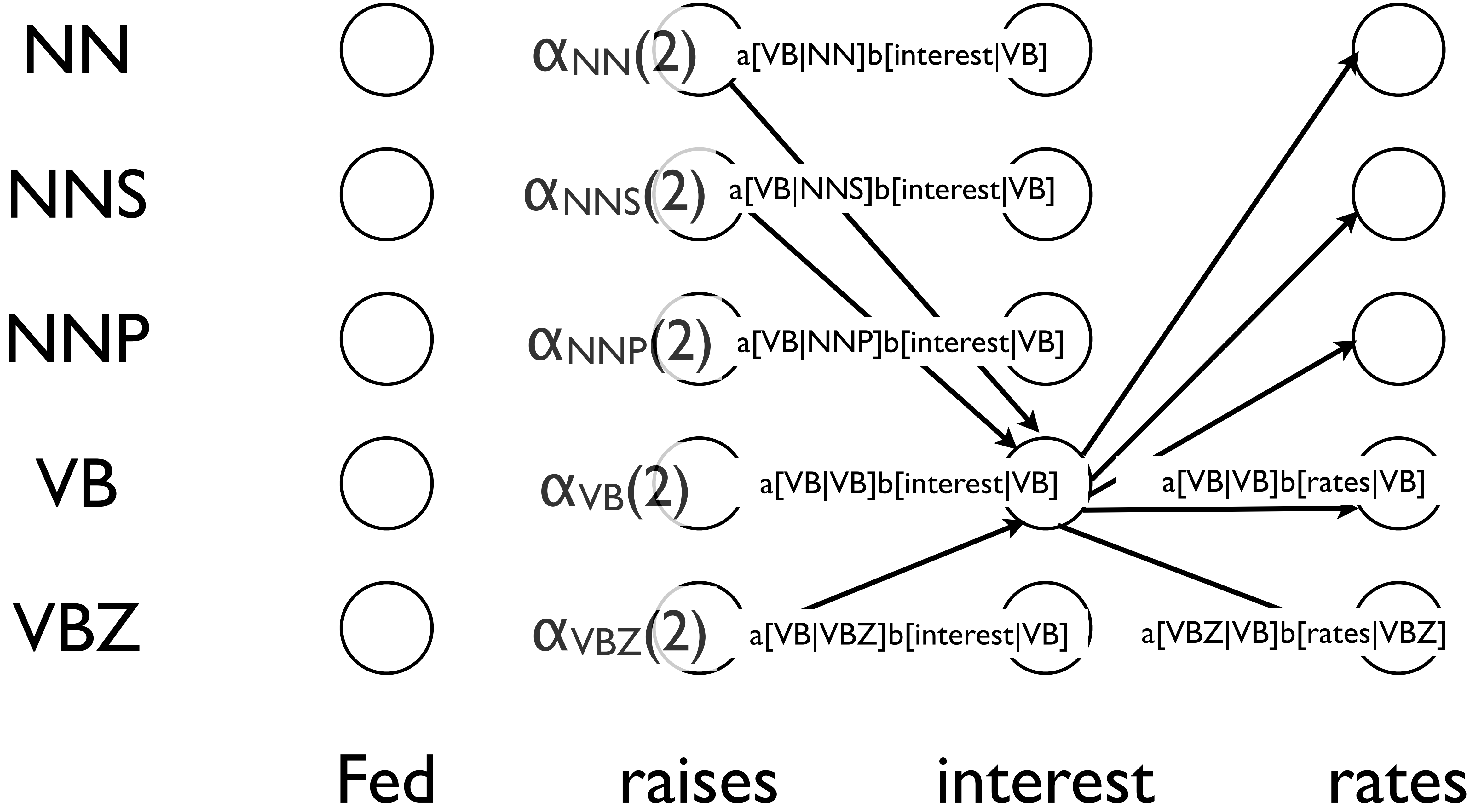
Forward-Backward Algorithm



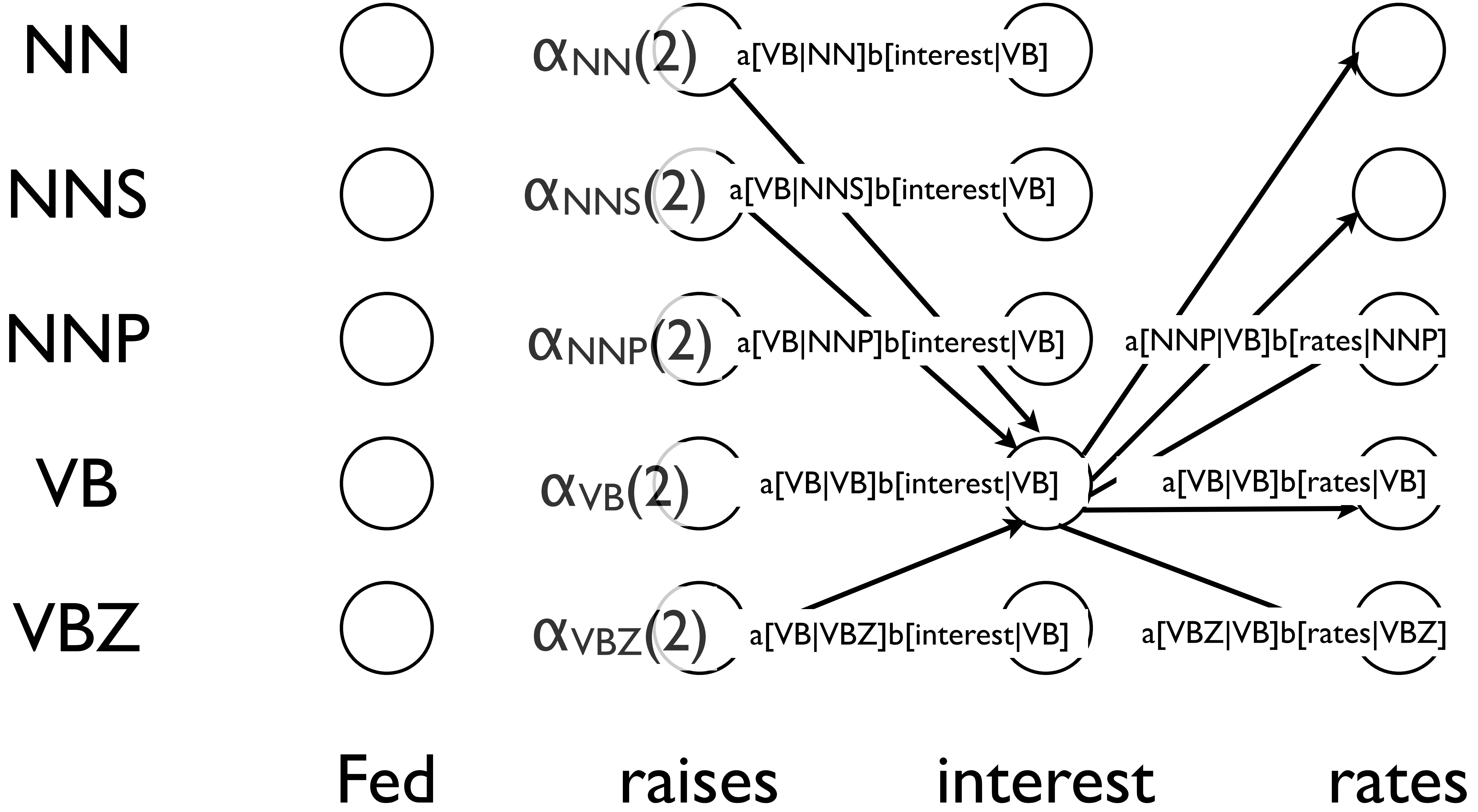
Forward-Backward Algorithm



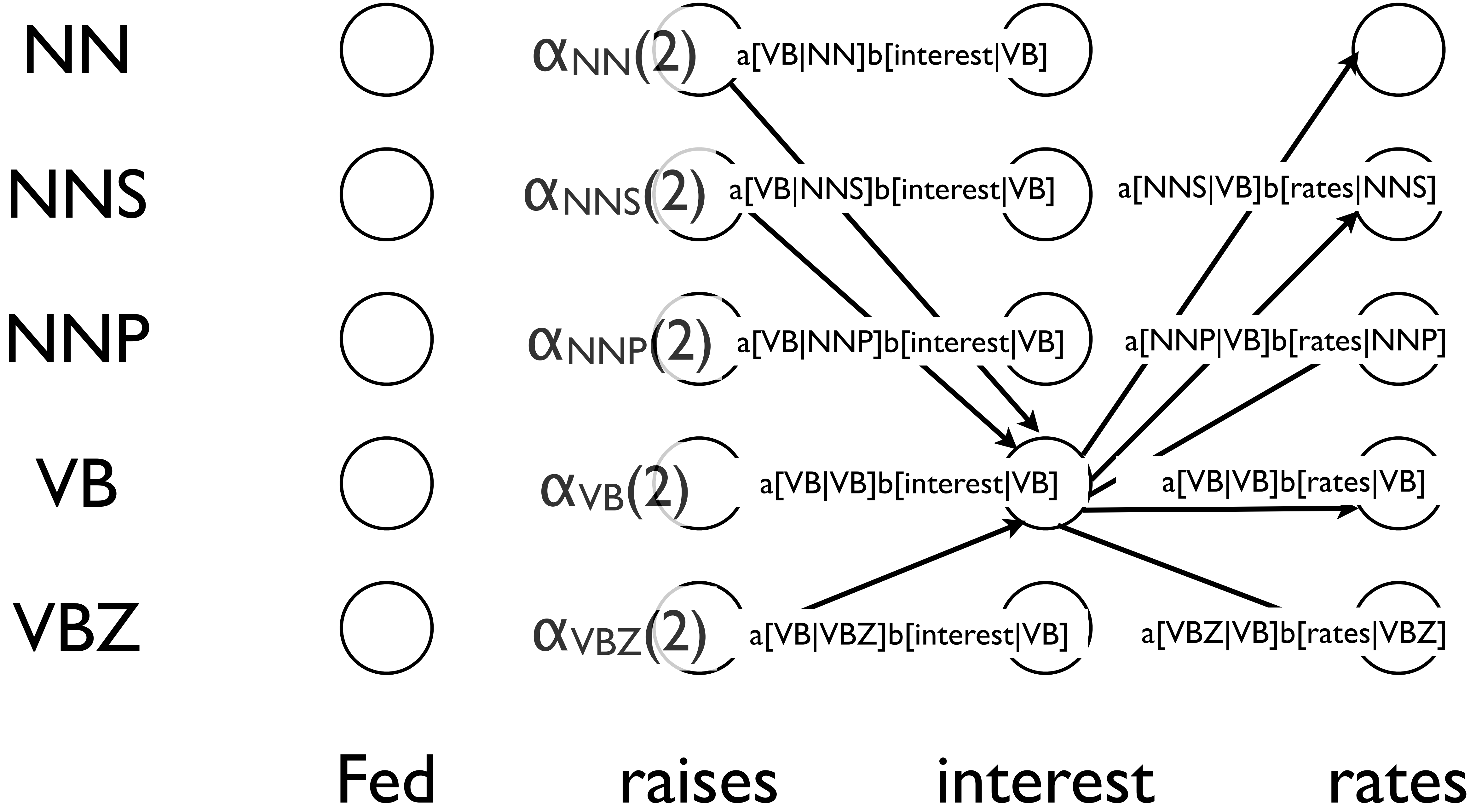
Forward-Backward Algorithm



Forward-Backward Algorithm



Forward-Backward Algorithm



NN

NNS

NNP

VB

VBZ

$\alpha_{NN}(2)$ a[VB|NN]b[interest|VB]

$\alpha_{NNS}(2)$ a[VB|NNS]b[interest|VB] a[NNS|VB]b[rates|NNS]

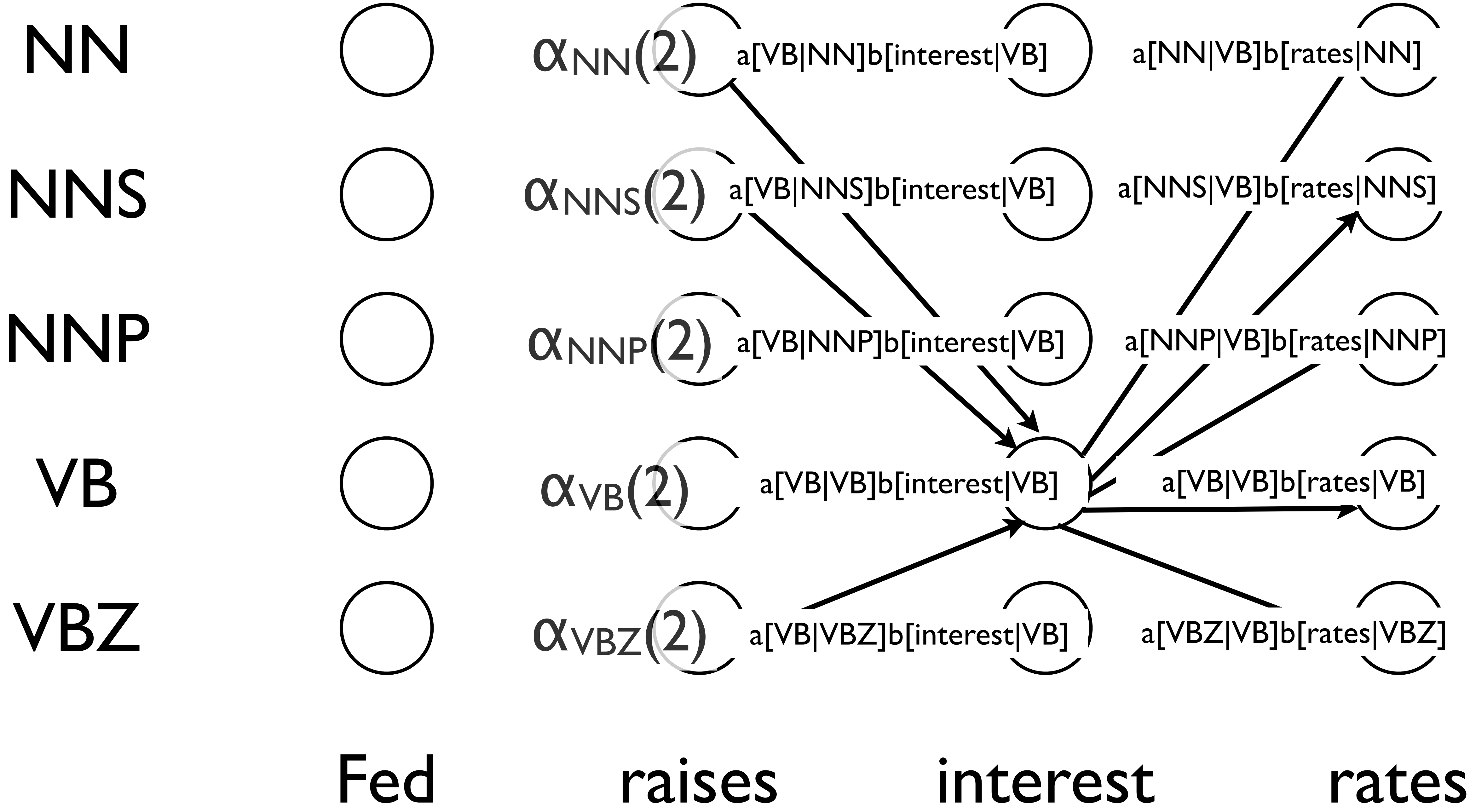
$\alpha_{NNP}(2)$ a[VB|NNP]b[interest|VB] a[NNP|VB]b[rates|NNP]

$\alpha_{VB}(2)$ a[VB|VB]b[interest|VB] a[VB|VB]b[rates|VB]

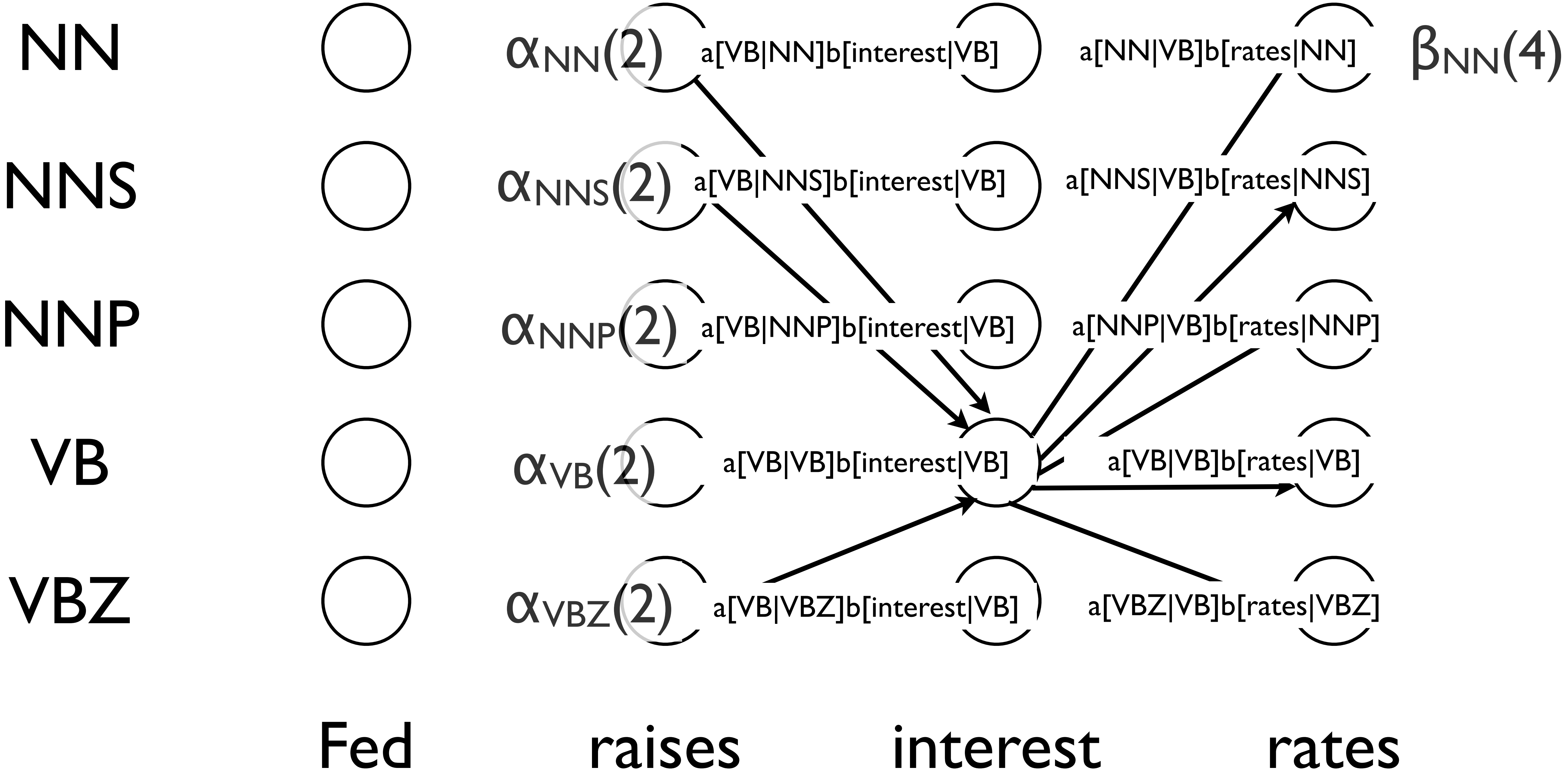
$\alpha_{VBZ}(2)$ a[VB|VBZ]b[interest|VB] a[VBZ|VB]b[rates|VBZ]

Fed raises interest rates

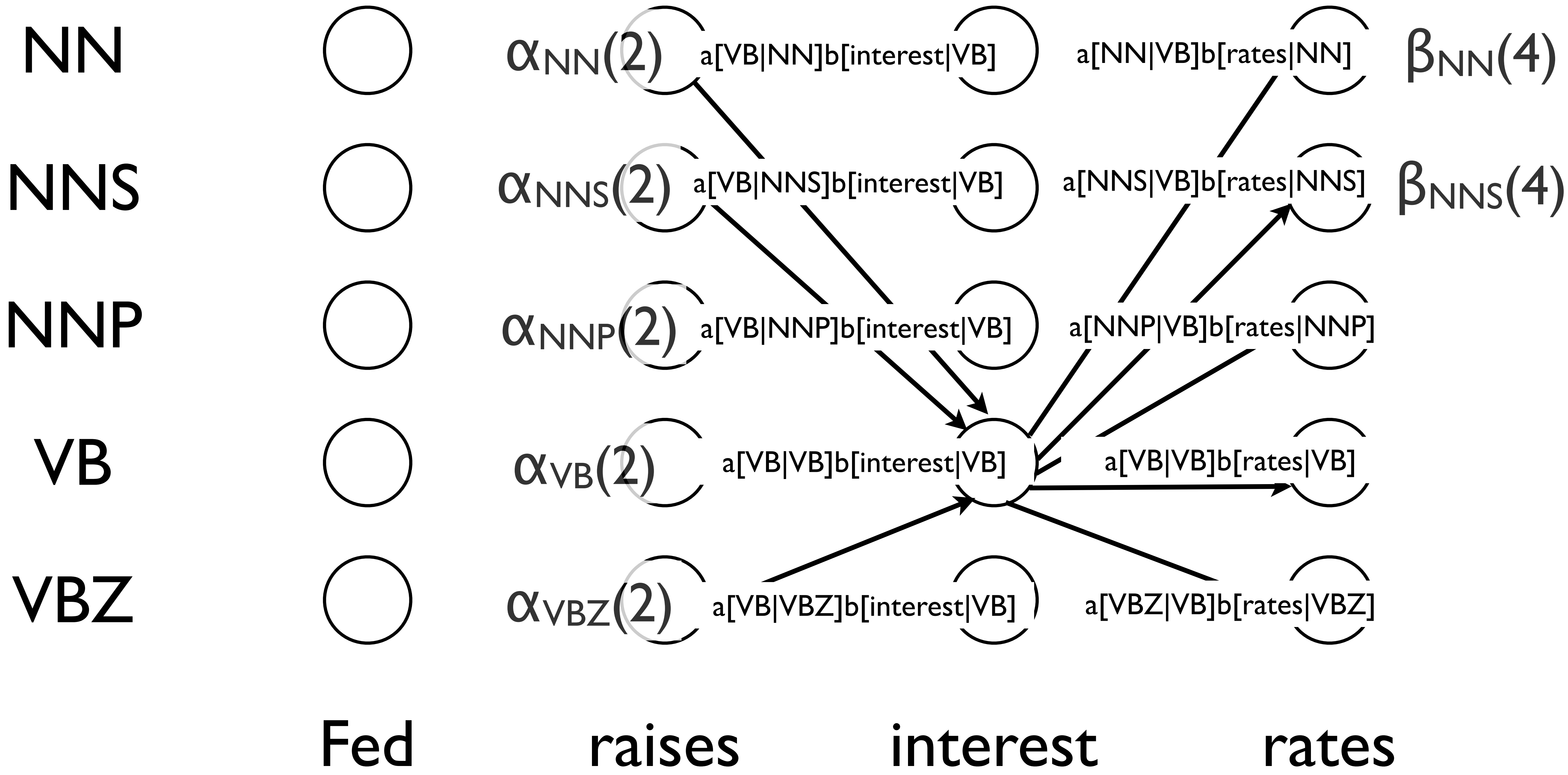
Forward-Backward Algorithm



Forward-Backward Algorithm



Forward-Backward Algorithm



NN $\alpha_{NN}(2)$ a[VB|NN]b[interest|VB] a[NN|VB]b[rates|NN] $\beta_{NN}(4)$

NNS $\alpha_{NNS}(2)$ a[VB|NNS]b[interest|VB] a[NNS|VB]b[rates|NNS] $\beta_{NNS}(4)$

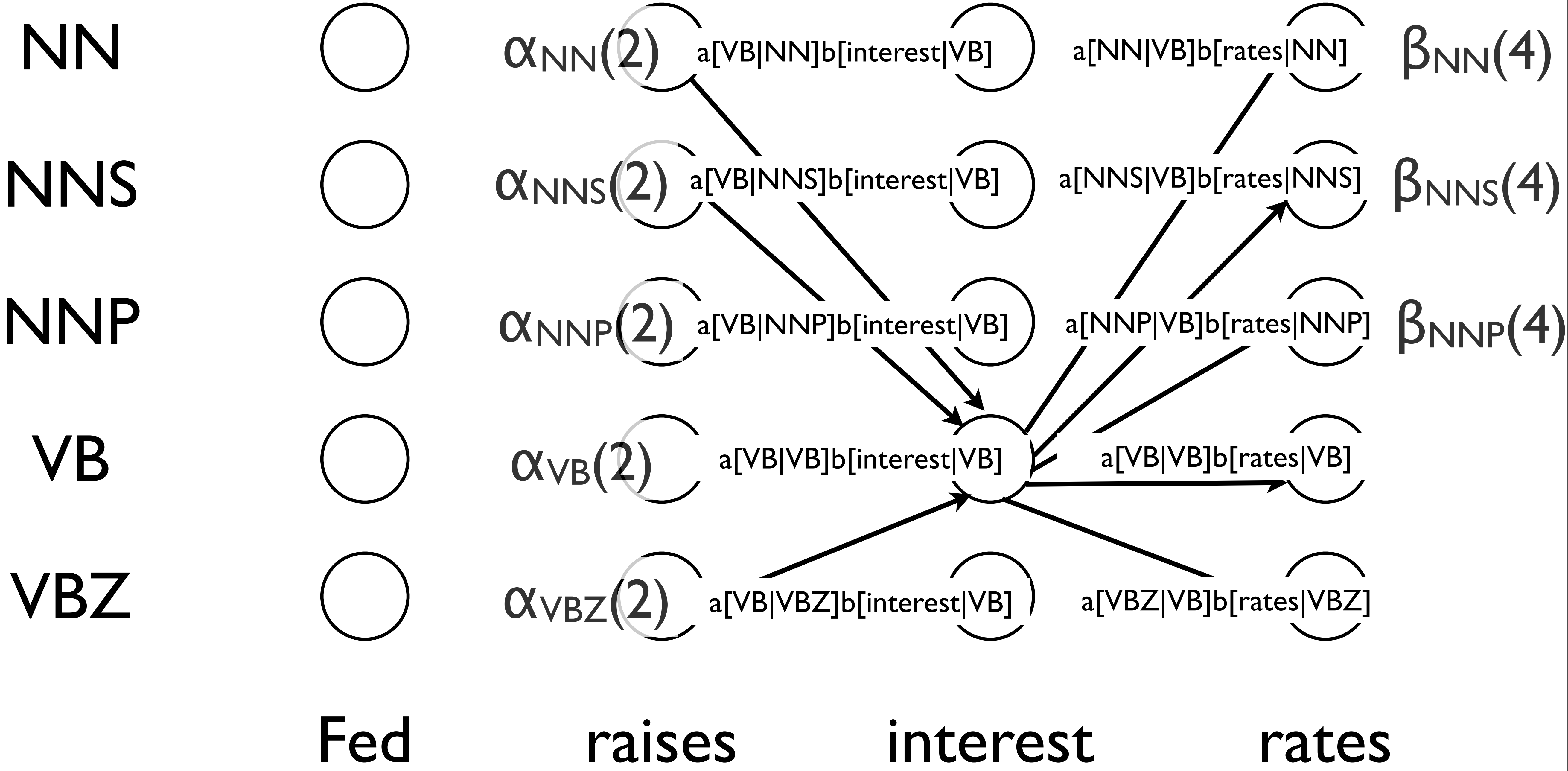
NNP $\alpha_{NNP}(2)$ a[VB|NNP]b[interest|VB] a[NNP|VB]b[rates|NNP]

VB $\alpha_{VB}(2)$ a[VB|VB]b[interest|VB] a[VB|VB]b[rates|VB]

VBZ $\alpha_{VBZ}(2)$ a[VB|VBZ]b[interest|VB] a[VBZ|VB]b[rates|VBZ]

Fed raises interest rates

Forward-Backward Algorithm



NN

NNS

NNP

VB

VBZ

$\alpha_{NN}(2)$ a[VB|NN]b[interest|VB] a[NN|VB]b[rates|NN] $\beta_{NN}(4)$

$\alpha_{NNS}(2)$ a[VB|NNS]b[interest|VB] a[NNS|VB]b[rates|NNS] $\beta_{NNS}(4)$

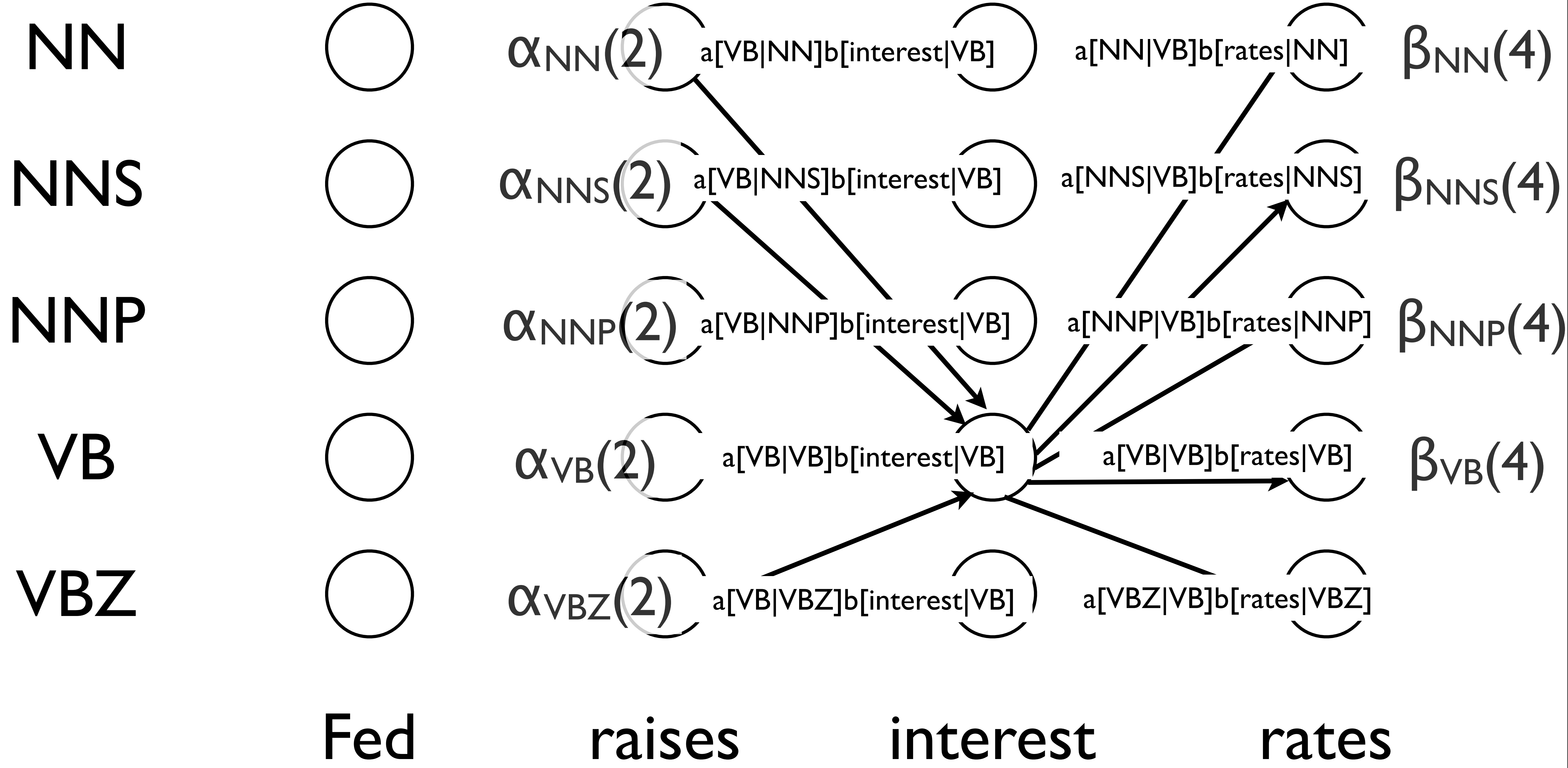
$\alpha_{NNP}(2)$ a[VB|NNP]b[interest|VB] a[NNP|VB]b[rates|NNP] $\beta_{NNP}(4)$

$\alpha_{VB}(2)$ a[VB|VB]b[interest|VB] a[VB|VB]b[rates|VB]

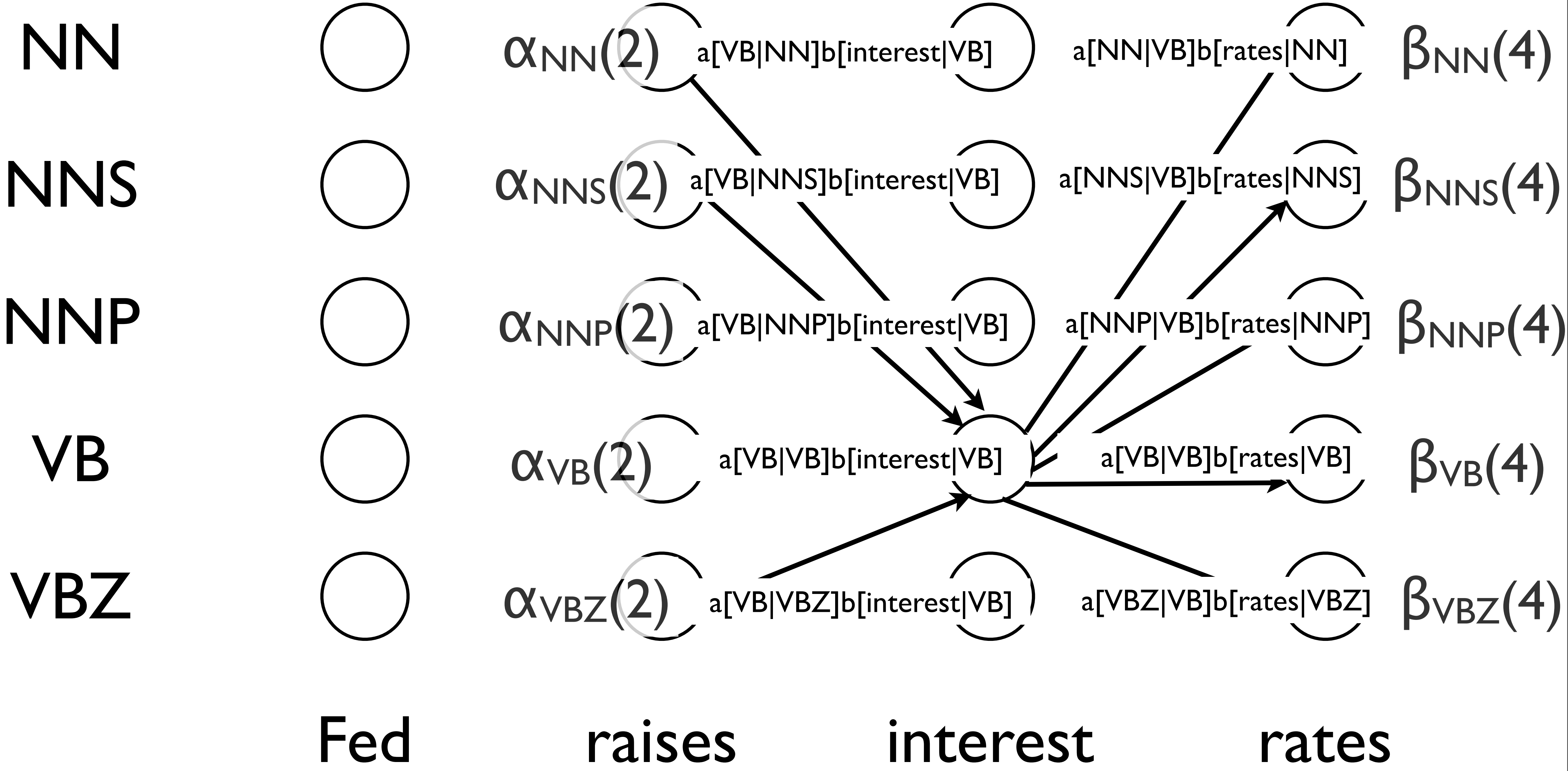
$\alpha_{VBZ}(2)$ a[VB|VBZ]b[interest|VB] a[VBZ|VB]b[rates|VBZ]

Fed raises interest rates

Forward-Backward Algorithm



Forward-Backward Algorithm



NN $\alpha_{NN}(2)$ a[VB|NN]b[interest|VB] a[NN|VB]b[rates|NN] $\beta_{NN}(4)$

NNS $\alpha_{NNS}(2)$ a[VB|NNS]b[interest|VB] a[NNS|VB]b[rates|NNS] $\beta_{NNS}(4)$

NNP $\alpha_{NNP}(2)$ a[VB|NNP]b[interest|VB] a[NNP|VB]b[rates|NNP] $\beta_{NNP}(4)$

VB $\alpha_{VB}(2)$ a[VB|VB]b[interest|VB] a[VB|VB]b[rates|VB] $\beta_{VB}(4)$

VBZ $\alpha_{VBZ}(2)$ a[VB|VBZ]b[interest|VB] a[VBZ|VB]b[rates|VBZ] $\beta_{VBZ}(4)$

Fed raises interest rates

Forward-Backward Algorithm

$$P(o_1 \cdots o_{t-1}, x_t = j \mid \mu) = \alpha_j(t)$$

$$P(o_t \cdots o_T \mid x_t = j, \mu) = \beta_j(t)$$

$$P(o_1 \cdots o_T, x_t = j \mid \mu) = \alpha_j(t)\beta_j(t)$$

$$P(x_t = j \mid O, \mu) = \frac{P(x_t = j, O \mid \mu)}{P(O \mid \mu)} = \frac{{}_j(t)\quad {}_j(t)}{{}_{\#}(T)}$$

$$\begin{aligned} P(x_t = i, x_{t+1} = j \mid O, \mu) &= \frac{P(x_t = i, x_{t+1} = j, O \mid \mu)}{P(O \mid \mu)} \\ &= \frac{{}_i(t)a[j \mid i]b[o_t \mid j]\quad {}_j(t+1)}{{}_{\#}(T)} \end{aligned}$$

Expectation Maximization (EM)

- Iterative algorithm to maximize likelihood of observed data in the presence of hidden data (e.g., tags)
- Choose an initial model μ
- **Expectation step**: find the expected value of hidden variables given current μ
- **Maximization step**: choose new μ to maximize probability of hidden and observed data
- Guaranteed to increase likelihood
- Not guaranteed to find global maximum

Supervised vs. Unsupervised

Supervised	Unsupervised
Annotated training text	Plain text
Simple count/normalize	EM
Fixed tag set	Set during training
Training reads data once	Training needs multiple passes

Logarithms for Precision

$$P(Y) = p(y_1)p(y_2)\cdots p(y_T)$$

$$\log P(Y) = \log p(y_1) + \log p(y_2) \cdots + \log p(y_T)$$

Increased dynamic range of [0,1] to [-∞,0]

Semirings

	⊕	⊗	0	1
Real	+	x	0	1
Max	max	x	0	1
Log	log+	+	-∞	0
“Tropical”	max	+	-∞	0
Shortest path	min	+	∞	0
Boolean	∨	∧	F	T