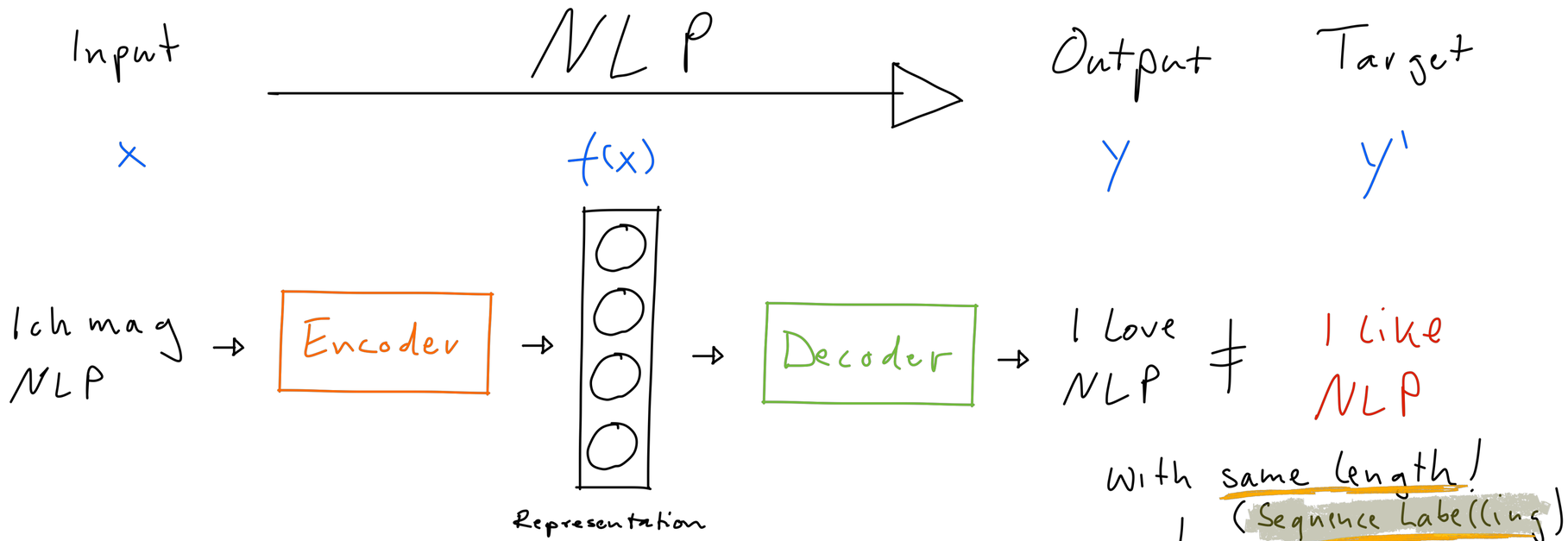
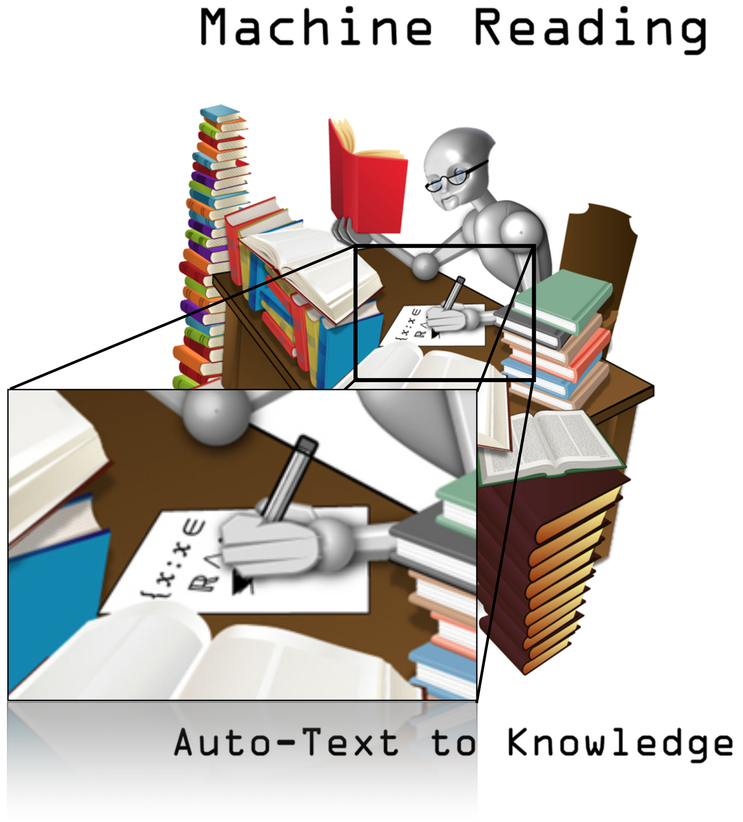
# Sequence Labelling

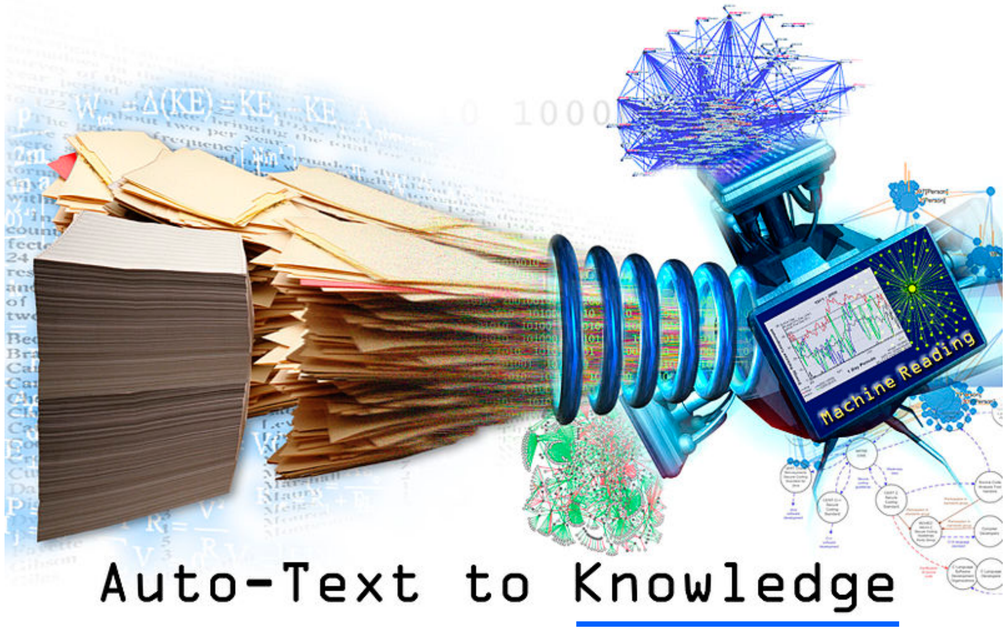
Tim Rocktäschel & **Sebastian Riedel**
COMP0087 Natural Language Processing

# NLP in a Nutshell



# Machine Reading




Huh?

# Machine Reading

andrew mccallum

Web Images Maps Shopping More Search tools

About 4,380,000 results (0.20 seconds)

Cookies help us deliver our services. By using our services, you agree to our use of cookies.

OK Learn more

**Andrew McCallum** Homepage
www.cs.umass.edu/~**mccallum**
Machine learning, text and information retrieval and extraction, reinforcement learning.
Andrew McCallum Publications - Andrew McCallum Bio - People - Teaching

**Andrew McCallum** - London Metropolitan University
www.londonmet.ac.uk/faculties/faculty-of...k.../**andrew-mccallum**/
**Andrew** taught English in London secondary schools for 15 years before coming to London Met in 2008. He is course tutor for the PGCE in Secondary English ...

**Andrew McCallum** - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Andrew_McCallum**
**Andrew McCallum** is a professor and researcher in the computer science department at University of Massachusetts Amherst. His primary specialties are in ...

**Andrew Mccallum** - United Kingdom profiles | LinkedIn
uk.linkedin.com/pub/dir/**Andrew**/**Mccallum**
View the profiles of professionals on LinkedIn named **Andrew Mccallum** located in the United Kingdom. There are 25 professionals named **Andrew Mccallum** in ...

## Andrew McCallum

Software Developer

Andrew McCallum is a professor and researcher in the computer science department at University of Massachusetts Amherst. Wikipedia

**Education:** Dartmouth College, University of Rochester

**Awards:** Best 10-year Paper Award of the ICML

People also search for


Tom M. Mitchell


Lee Giles


David M. Blei


Michael Collins


Robert Schapire

Feedback/More info

# Machine Reading

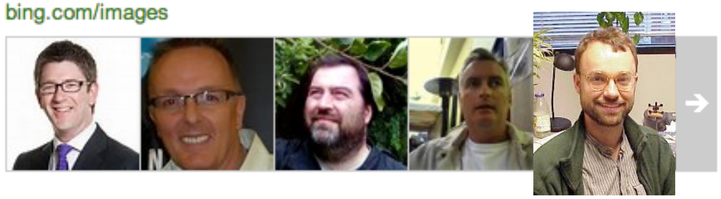
bing Andrew McCallum

382,000 RESULTS Any time

**Andrew McCallum** Homepage - UMass CS | School of Computer …
www.cs.umass.edu/~**mccallum**
Machine learning, text and information retrieval and extraction, reinforcement learning.

**Andrew McCallum** - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/**Andrew_McCallum**
**Andrew McCallum** is a professor and researcher in the computer science department at University of Massachusetts Amherst. His primary specialties are in ...

Images of **Andrew McCallum**
bing.com/images



**Andrew_McCallum** - Twitter
www.twitter.com/**andrew_mccallum**
We would like to show you a description here but the site won't allow us.

**andrew mccallum** profiles | LinkedIn
www.linkedin.com/pub/dir/**andrew/mccallum**
View the profiles of professionals named **andrew mccallum** on LinkedIn. There are 25 professionals named **andrew mccallum**, who use LinkedIn to exchange information ...

## Andrew McCallum

Andrew McCallum is a professor and researcher in the computer science department at University of Massachusetts Amherst. His primary specialties are in machine learning, natural language processing, informa… + en.wikipedia.org

Education: Dartmouth College · University of Rochester

Awards: Best 10-year Paper Award of the ICML

### People also search for

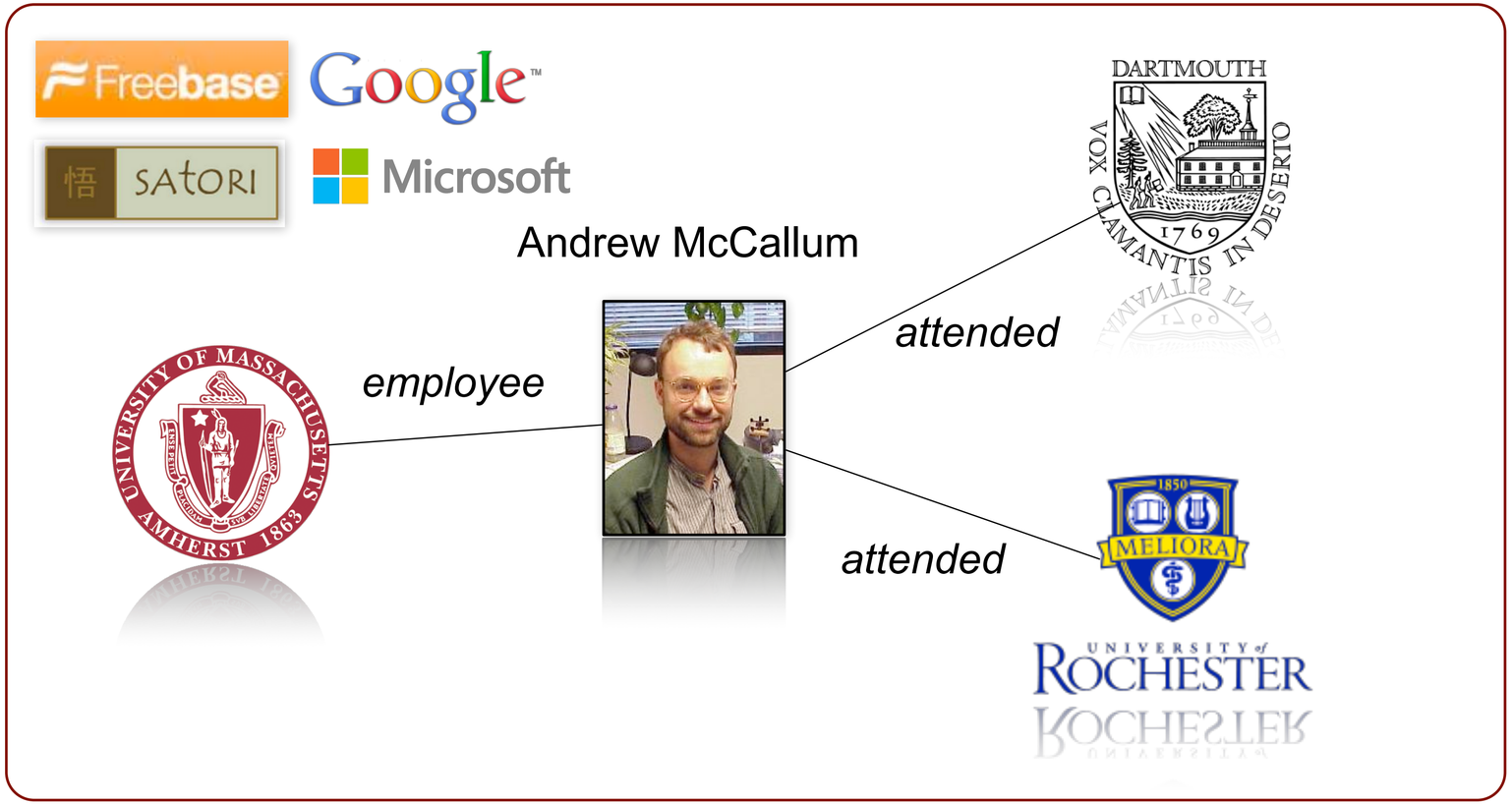

Tom M. Mitchell


Peter Norvig


Scott Fahlman

Report a problem

# Machine Reading



# Machine Reading

Professor

**Andrew McCallum**

Computer Science Department
University of Massachusetts Amherst
mccallum@cs.umass.edu
+1 413 545-1323 (vox)
+1 413 545-1789 (fax)

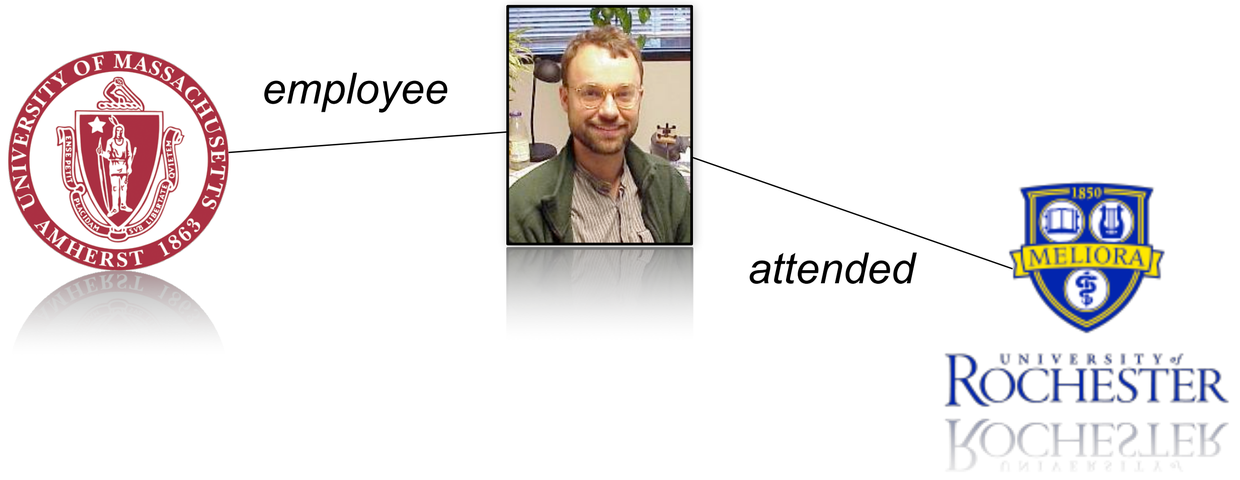
**Contact**
**Bio**
**Vita**

**Publications**
**Talks**
**Projects**

## Current Bio-sketch

Andrew McCallum is a Professor and Director of the Information Extraction and Synthesis Laboratory in the School of Computer Science at University of Massachusetts Amherst. He has published over 250 papers in many areas of AI, including natural language processing, machine learning, data mining and reinforcement learning, and his work has received over 30,000 citations.

He obtained his PhD from University of Rochester in 1995 with Dana Ballard and a postdoctoral fellowship from CMU with Tom Mitchell and Sebastian Thrun. In the early 2000's he was Vice President of Research and Development at at WhizBang Labs, a 170-person start-up company that used machine learning for information extraction from the Web.



# Usages of Structured Knowledge

- Search
- Question Answering
- Data Mining
- Intelligent Agents
- Visualization

# Tasks

- Named Entity Recognition
- Relation Extraction
- Coreference
- Entity Linking
- Semantic Parsing
- Reasoning

# Tasks considered here

- Named Entity Recognition
- Relation Extraction
- Coreference
- Entity Linking
- Semantic Parsing
- Reasoning

# Named Entity Recognition

Identify mentions of entities in text (will become nodes in graph)

[Org] US President [Per] Obama spoke to ...

...journalists in the [Loc] White House

# Named Entity Recognition

Needed in all kinds of domains

[Phos] Phosphorylation of [Prot TRAF2 [Reg] inhibits...

... binding to the [Prot] CD40 domain

# Phrase Labeling

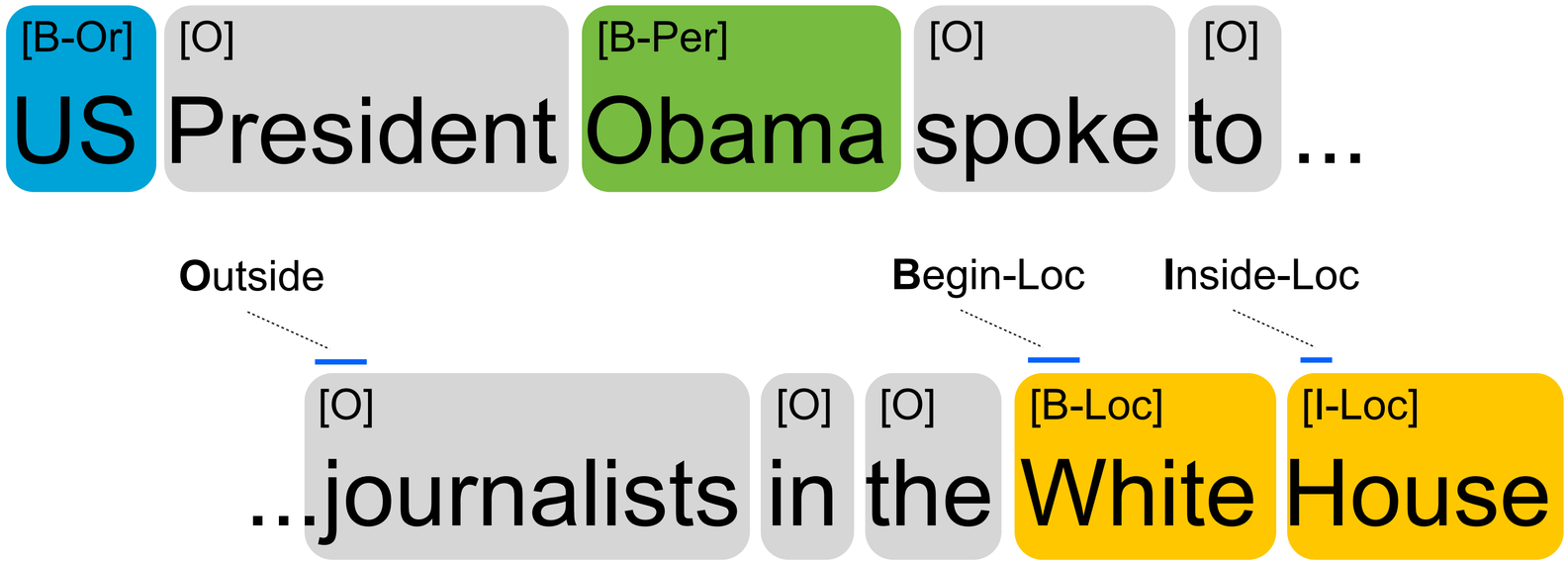
Label whole phrases with entity types (naively requires $O(n^2)$ time)

[Org] US President [Per] Obama spoke to ...

...journalists in the [Loc] White House

# Token Labeling (using BIO scheme)

Adapt label set and label on per-token basis (needs O(n)) time



# Token Labeling

Useful for all kinds of things: Shallow Parsing (Chunking)

[B-NP] US [I-NP] President [I-NP] Obama spoke to ...

...journalists in the White House

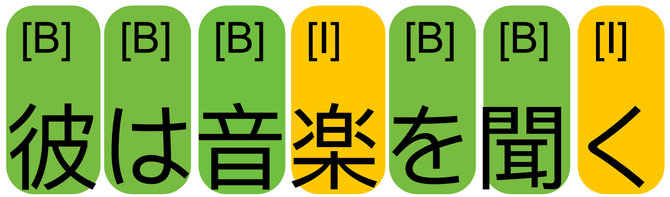
# Token Labeling

Useful for all kinds of things: Part-of-Speech Tagging


...journalists in the White House

# Token Labeling

Useful for all kinds of things: Word Segmentation



# NER Evaluation

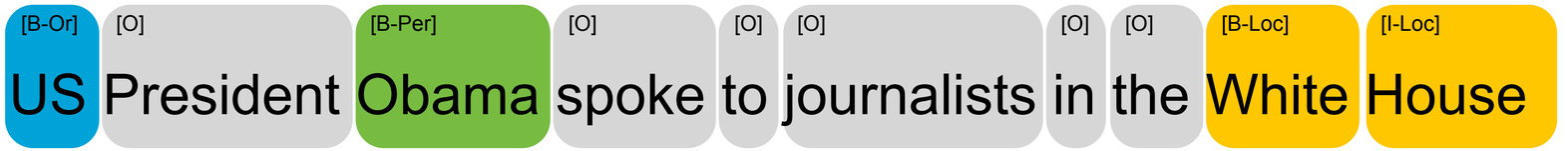
Precision: $$\frac{|\text{True positives (correctly predicted spans)}|}{|\text{All predicted spans}|}$$

Recall: $$\frac{|\text{True positives (correctly predicted spans)}|}{|\text{All true spans}|}$$

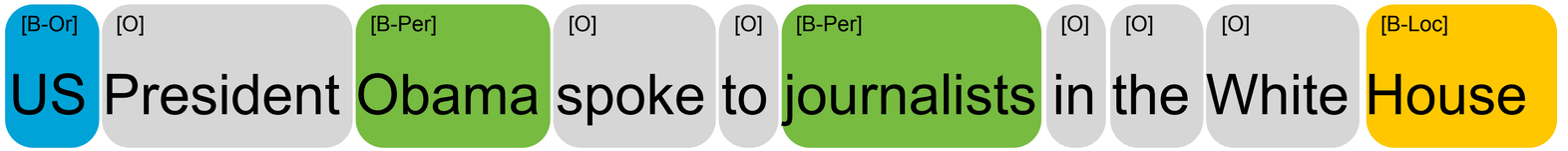
$F_1$: $$\frac{2 \cdot P \cdot R}{P + R}$$ Harmonic Mean

# NER Evaluation Example

TRUTH:

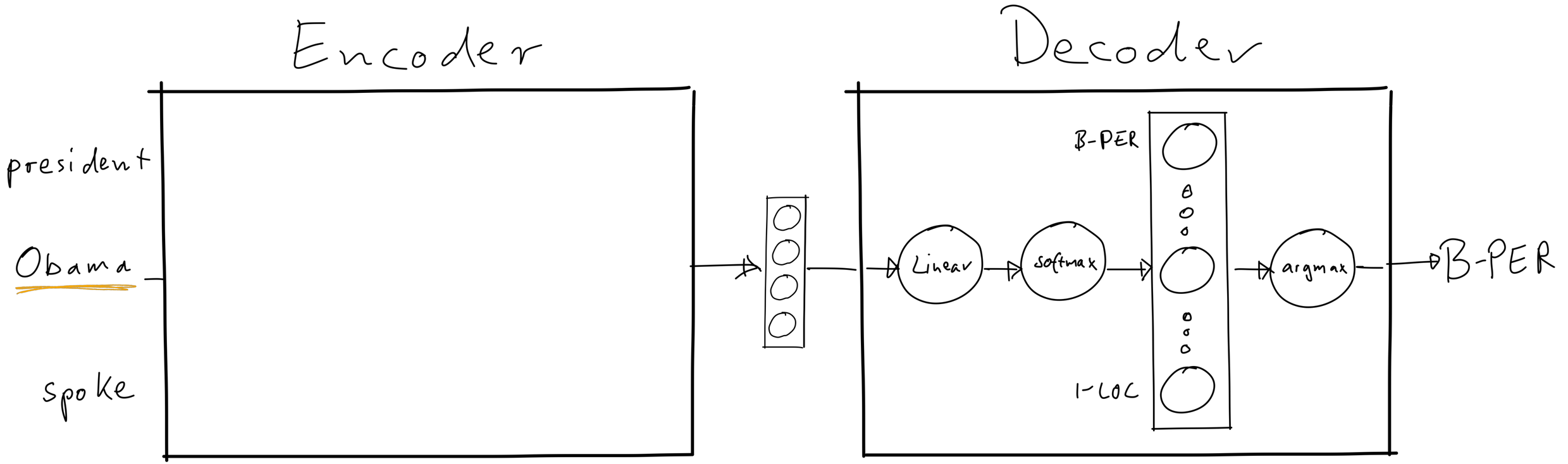



GUESS:



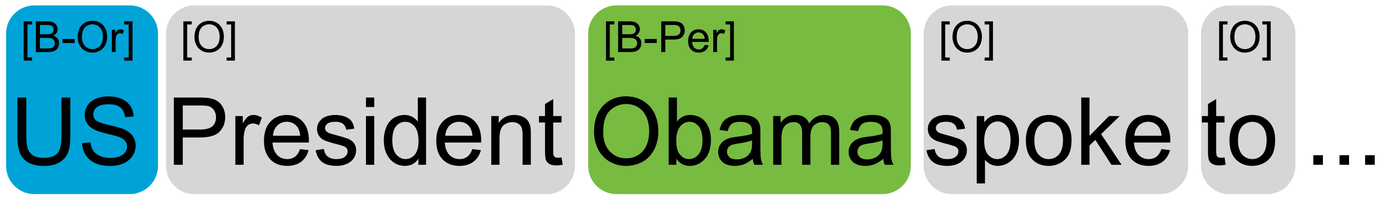

Precision:? Recall:? F1:?

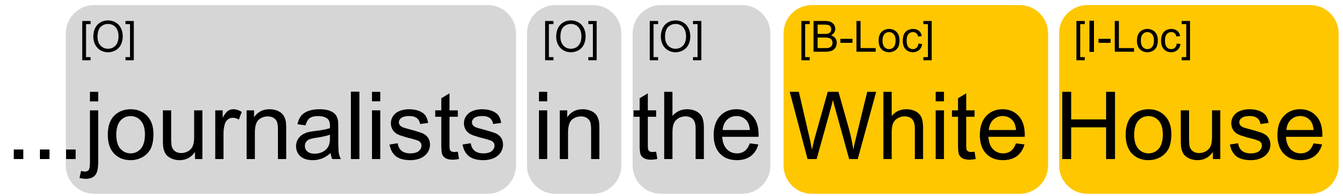
# Local Model


Encoder
Decoder
president
Obama
spoke
B-PER
I-LOC
Linear
softmax
argmax
B-PER

# Feature Engineering

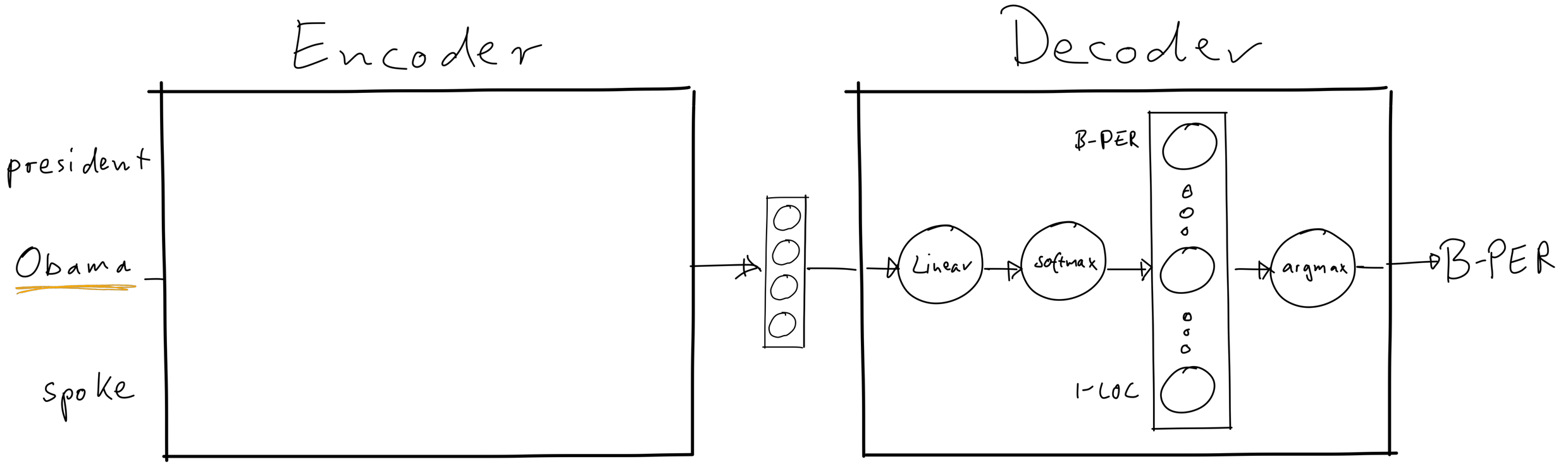
What could be good features (templates) for Named Entity Recognition?




Bias Feature?

# Model 1


Encoder
Decoder
president
Obama
spoke
B-PER
I-LOC
Linear
softmax
argmax
B-PER

# Model 1

Only a *bias feature*

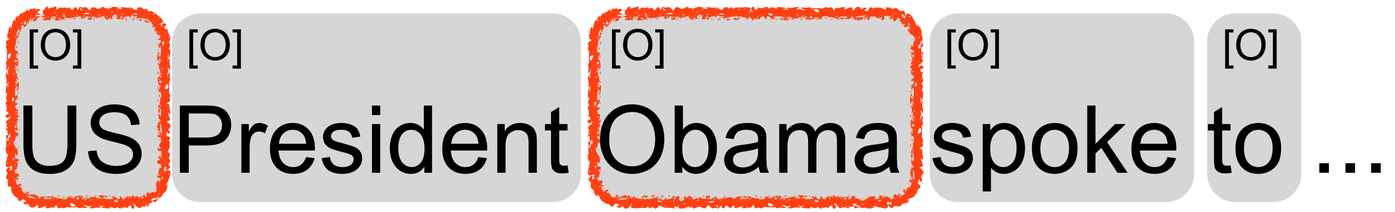
| [O] | [O] | [O] | [O] | [O] | |
|---|---|---|---|---|---|
| US | President | Obama | spoke | to | ... |

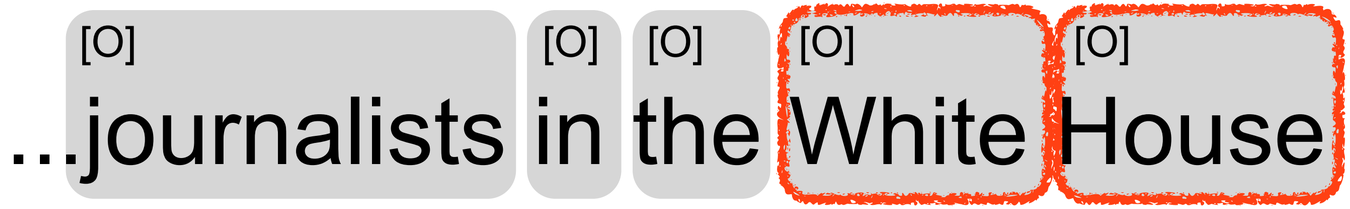
| | [O] | [O] | [O] | [O] | [O] |
|---|---|---|---|---|---|
| ... | journalists | in | the | White | House |

(I am showing predictions on a development set after adding feature)

# Model 1

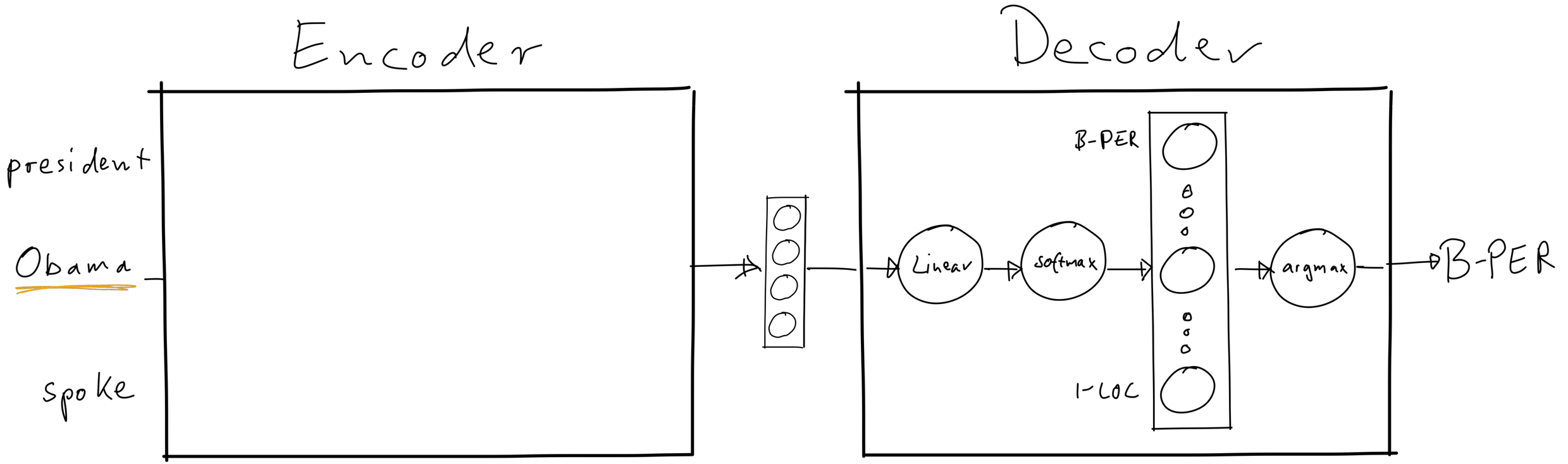
Only a bias



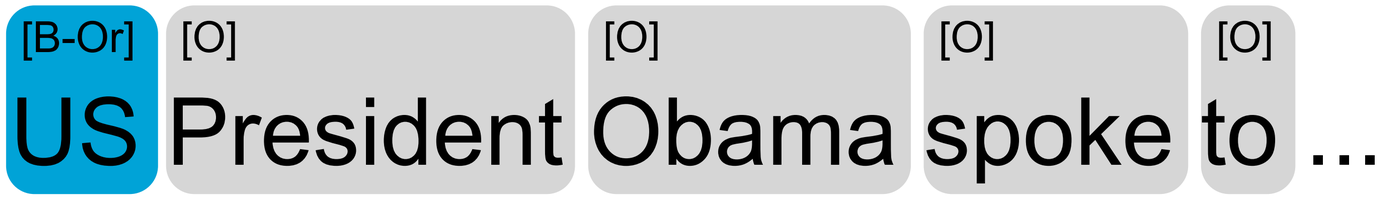

Take into account words?

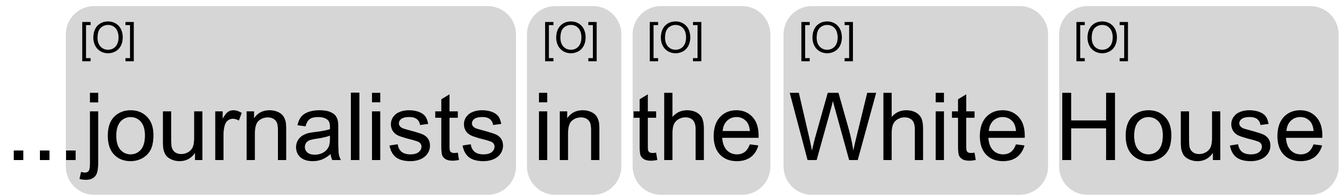
# Model 2


Encoder
Decoder
president
Obama
spoke
B-PER
I-LOC
Linear
softmax
argmax
B-PER

# Model 2

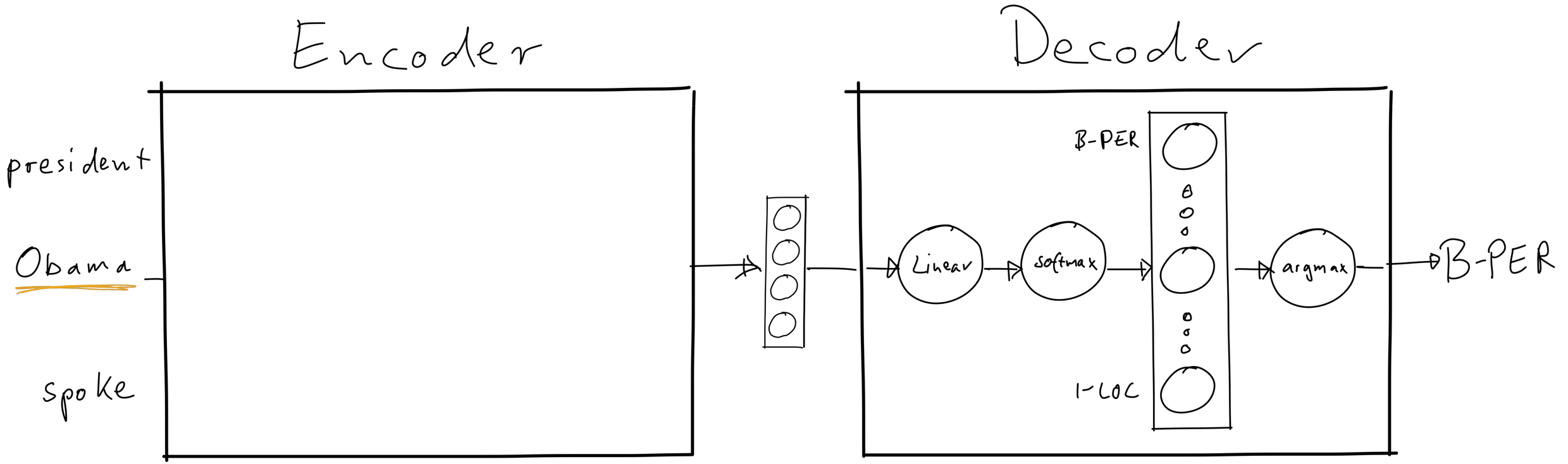
Add word identity



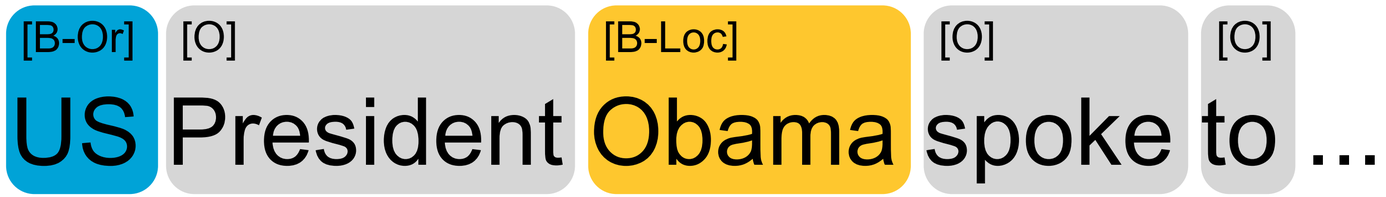

Take into account Capitalization?

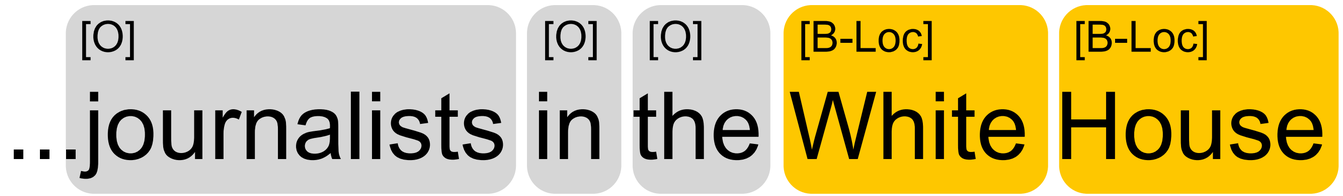
# Model 3


Encoder
Decoder
president
Obama
spoke
B-PER
I-LOC
Linear
softmax
argmax
B-PER

# Model 3

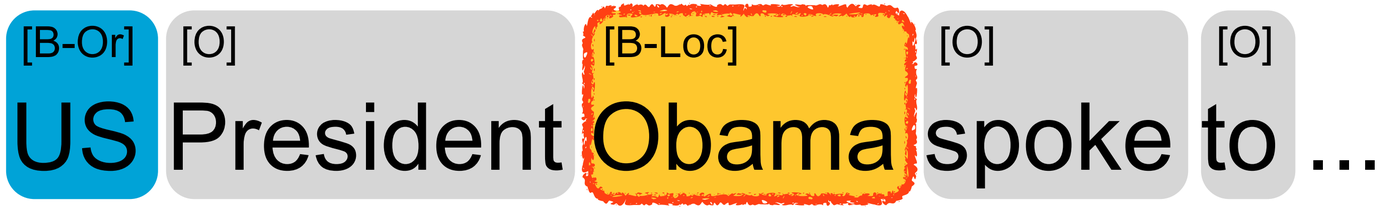
With firstCapital

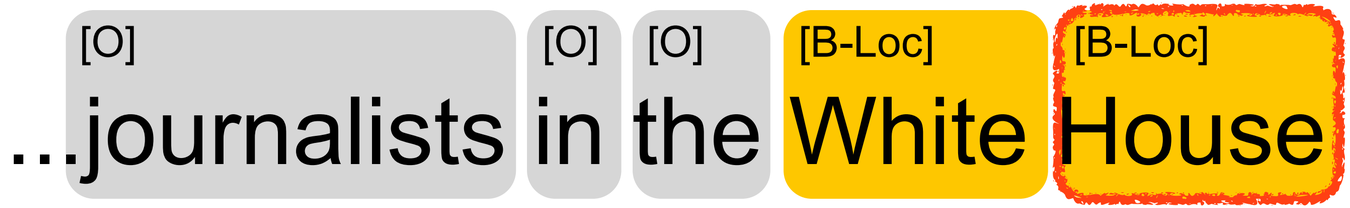


# Model 3

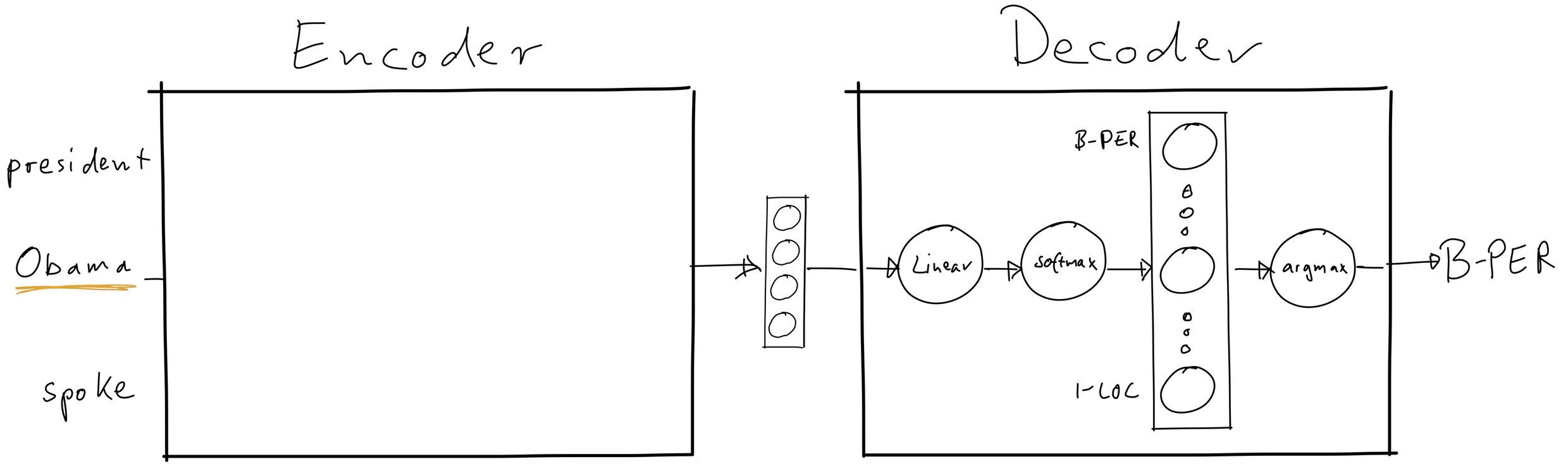
With firstCapital



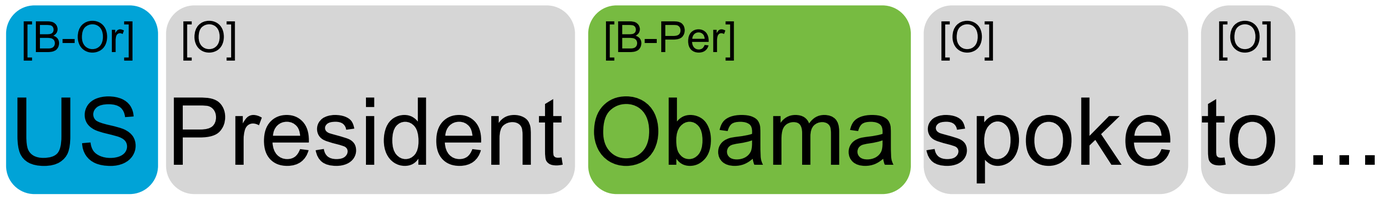

Use a person last name dictionary feature

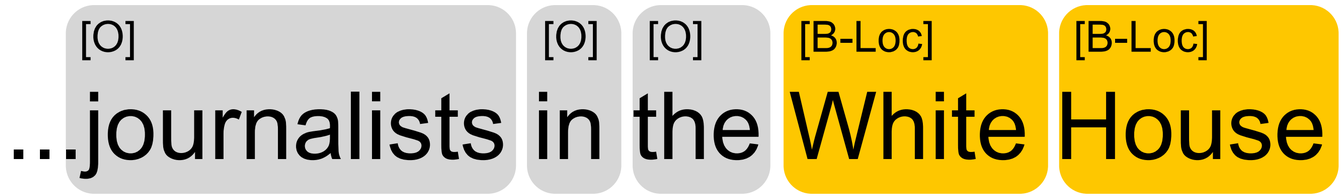
# Model 4


Encoder
Decoder
president
Obama
spoke
B-PER
I-LOC
Linear
softmax
argmax
B-PER

# Model 4

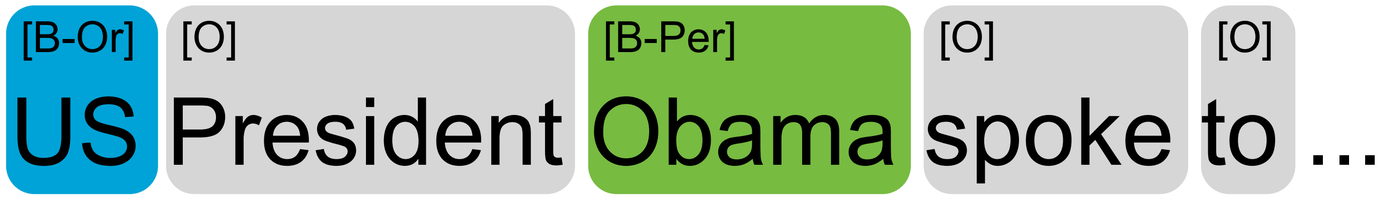
Use a person last name dictionary feature

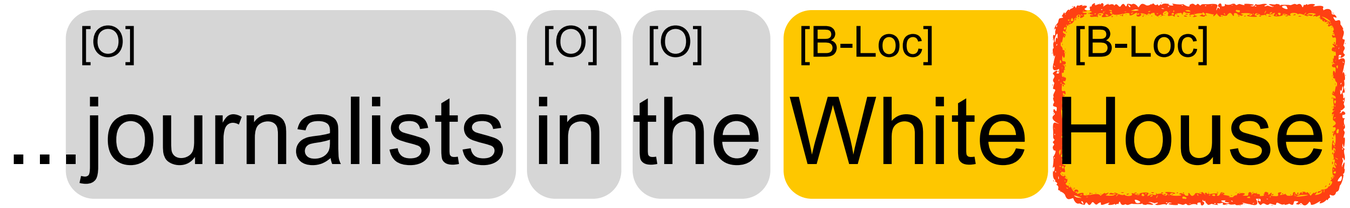


# Model 4

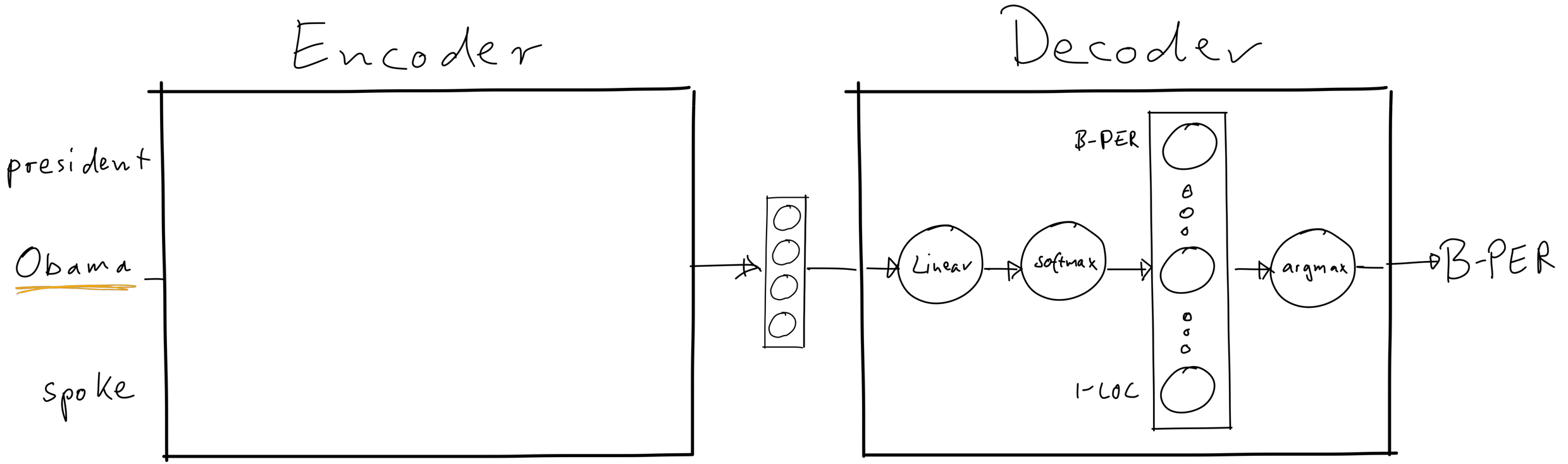
Use a person last name dictionary feature



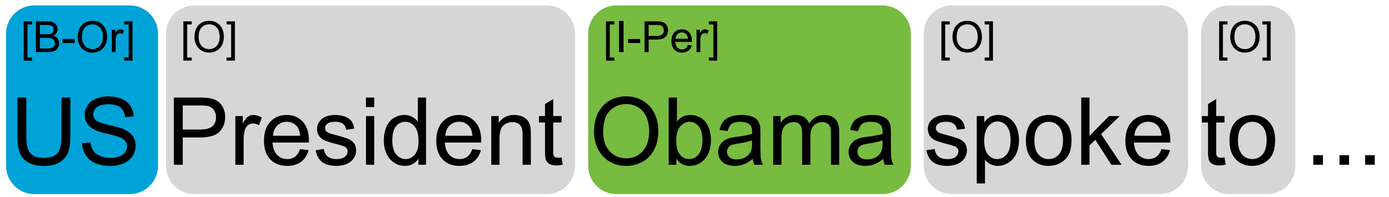

After capitalized words “I-” is more likely than “B-”

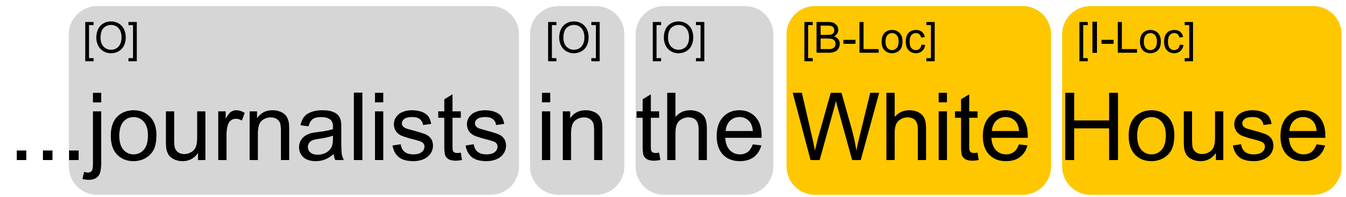
# Model 5


Encoder
Decoder
president
Obama
spoke
B-PER
I-LOC
Linear
softmax
argmax
B-PER

# Model 5

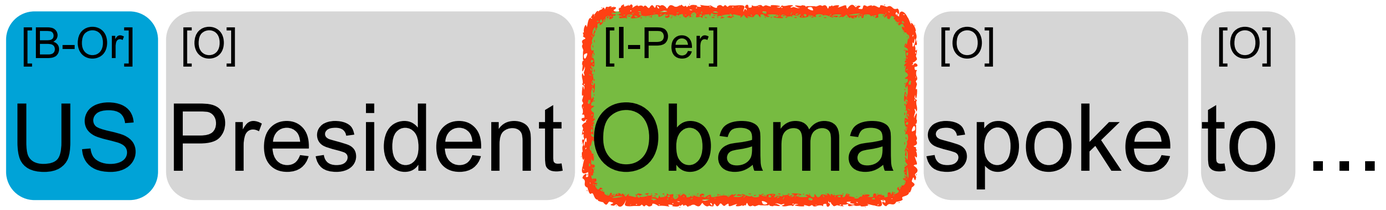
Previous word capitalized

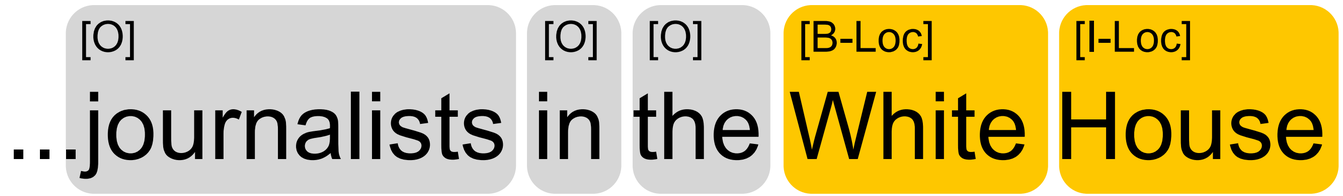


# Model 5

Previous word capitalized



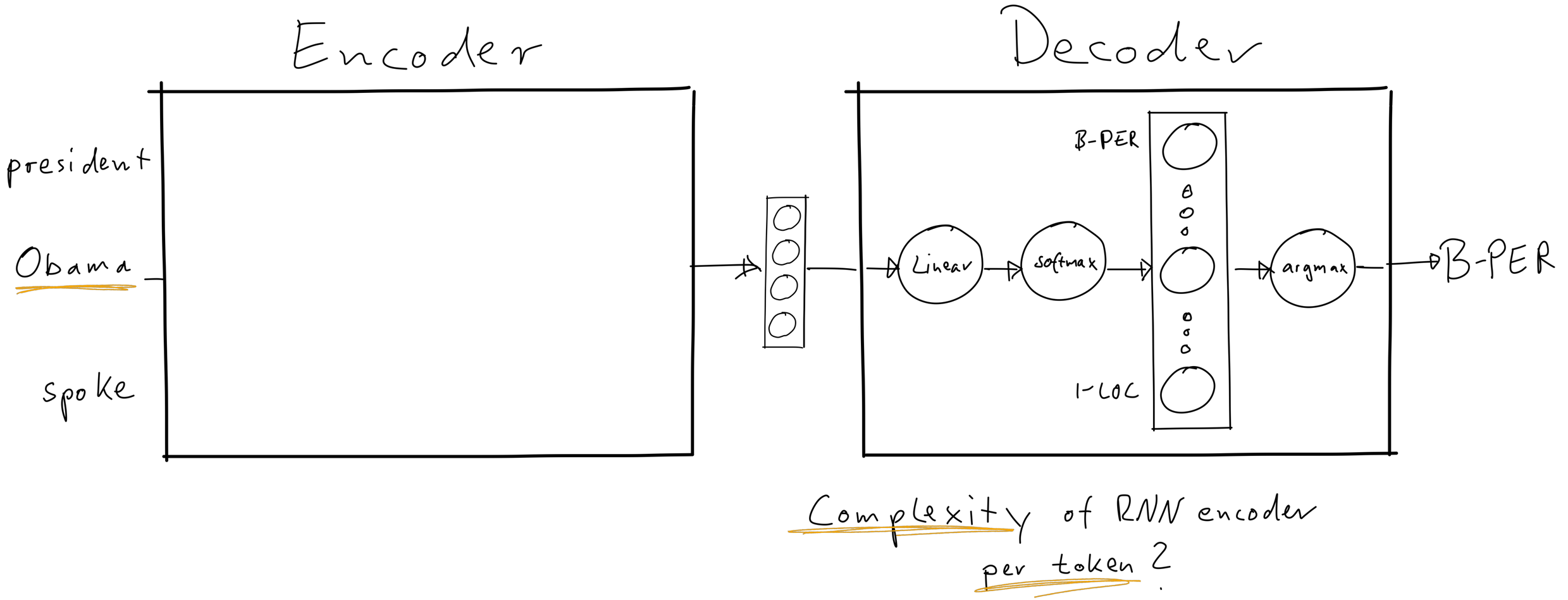
# Context Dependencies

Previous word affects the current label

# Context Dependencies

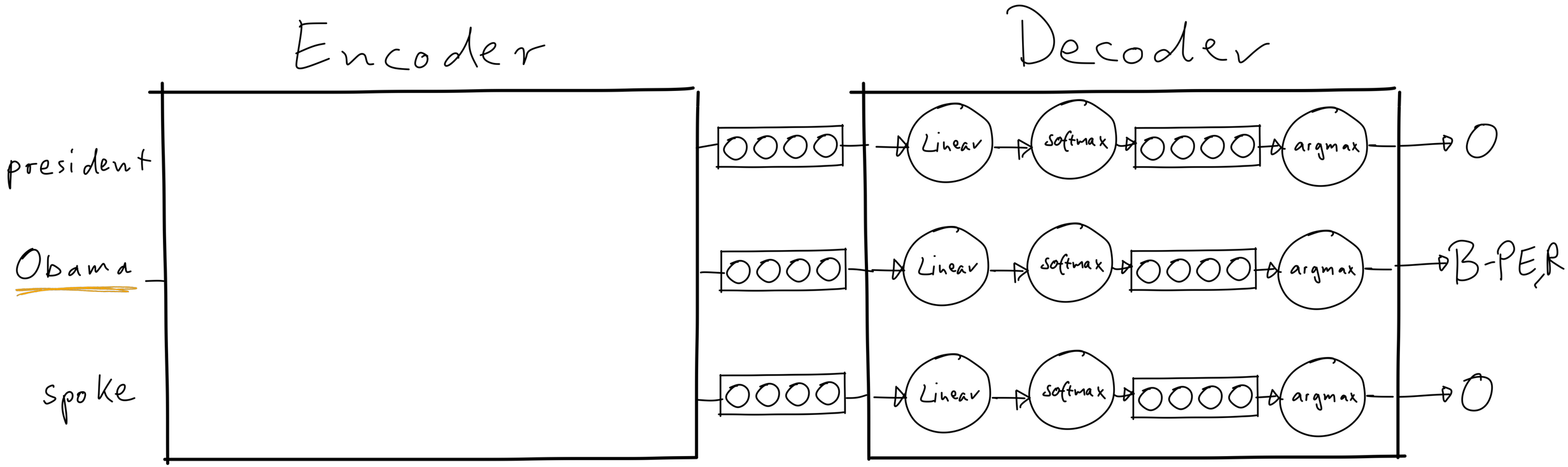
Previous word affects the current label

# Model 6



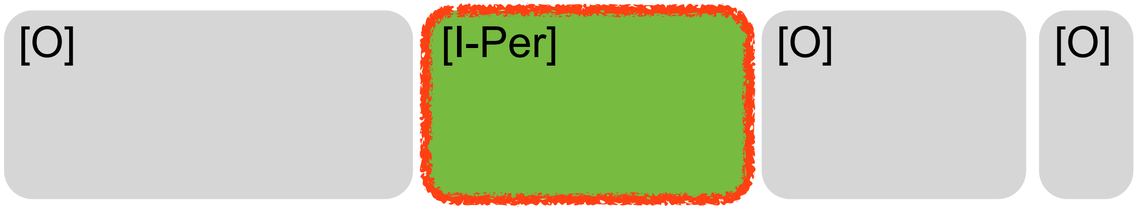

Complexity of RNN encoder per token ?

# Reusing Computation



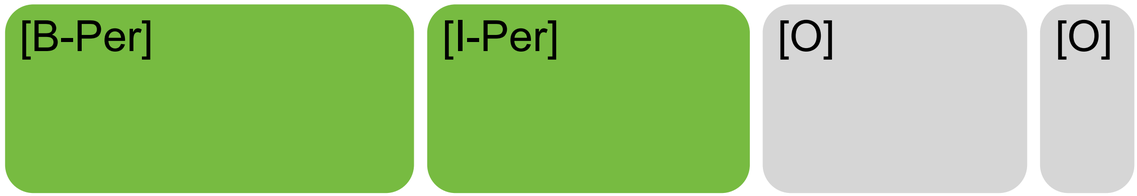
# Label Dependencies

Previous **label** affects the current label



# Label Dependencies

Previous **label** affects the current label

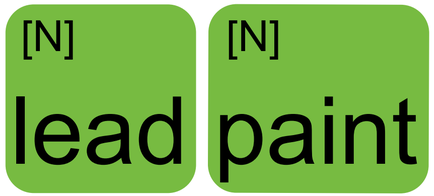


# Label Dependencies: Part-of-Speech

Previous label affects the current label

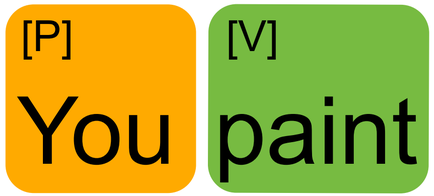
# Label Dependencies: Part-of-Speech

Previous label affects the current label

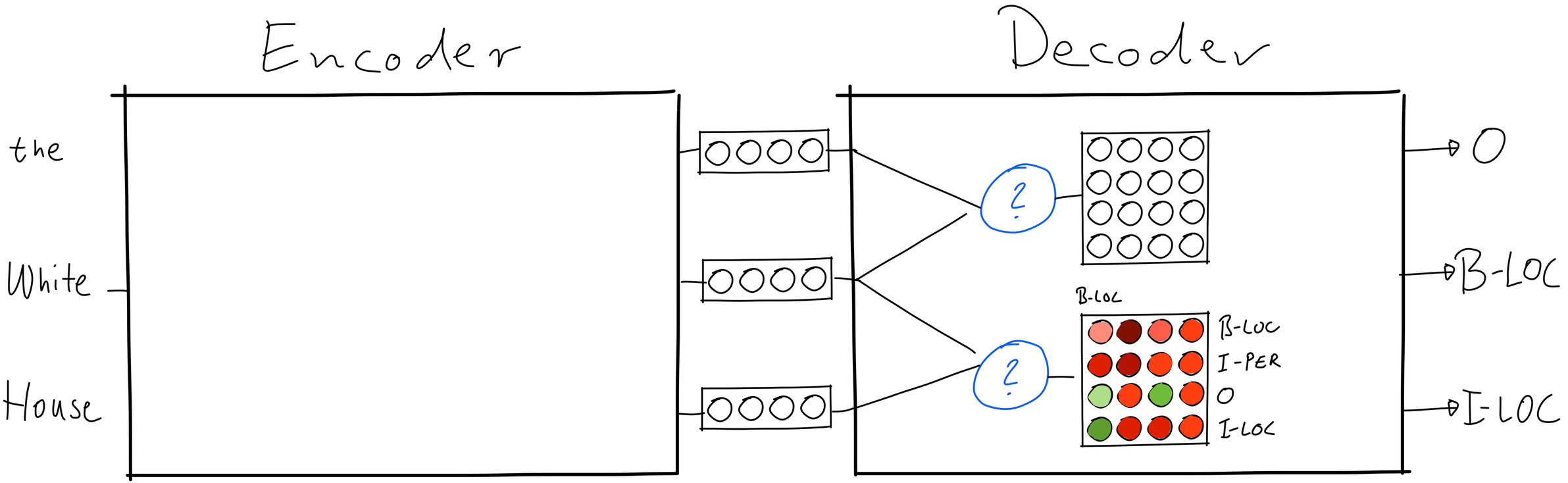


# Label Dependencies: Part-of-Speech

Previous label affects the current label



# Modelling Label Dependencies

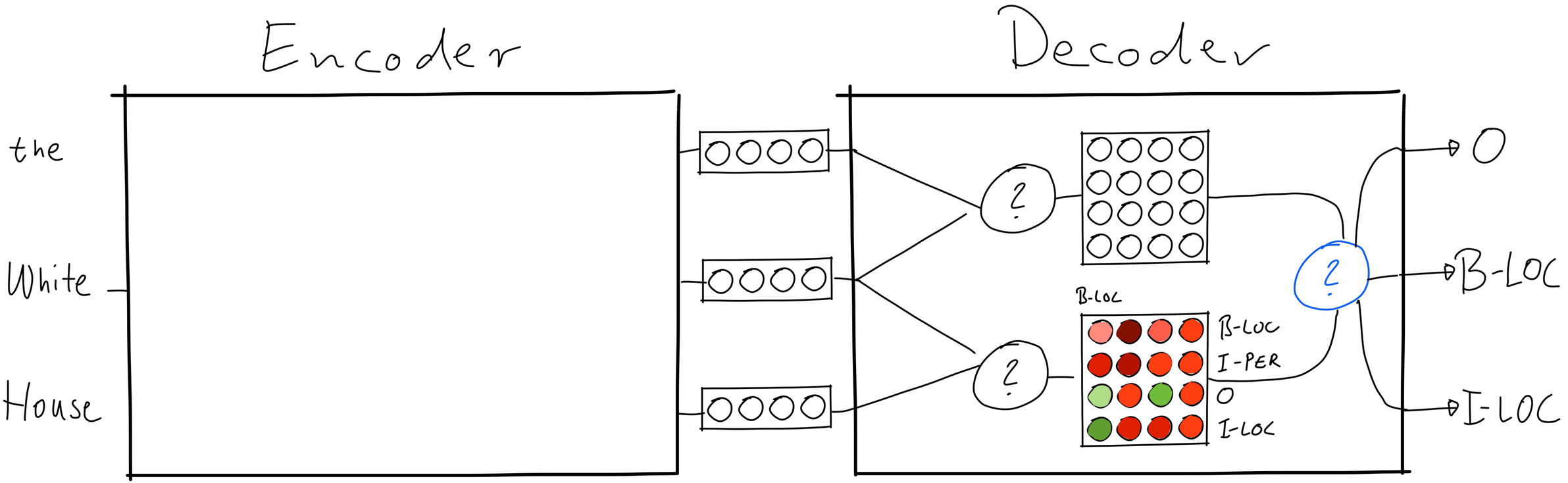



$$s_i(y_i, y_{i+1}) = s_i(y_i) + s_{i+1}(y_{i+1}) + \theta_{y_i, y_{i+1}}$$

Local scorer

Transition Matrix

# Modelling Label Dependencies



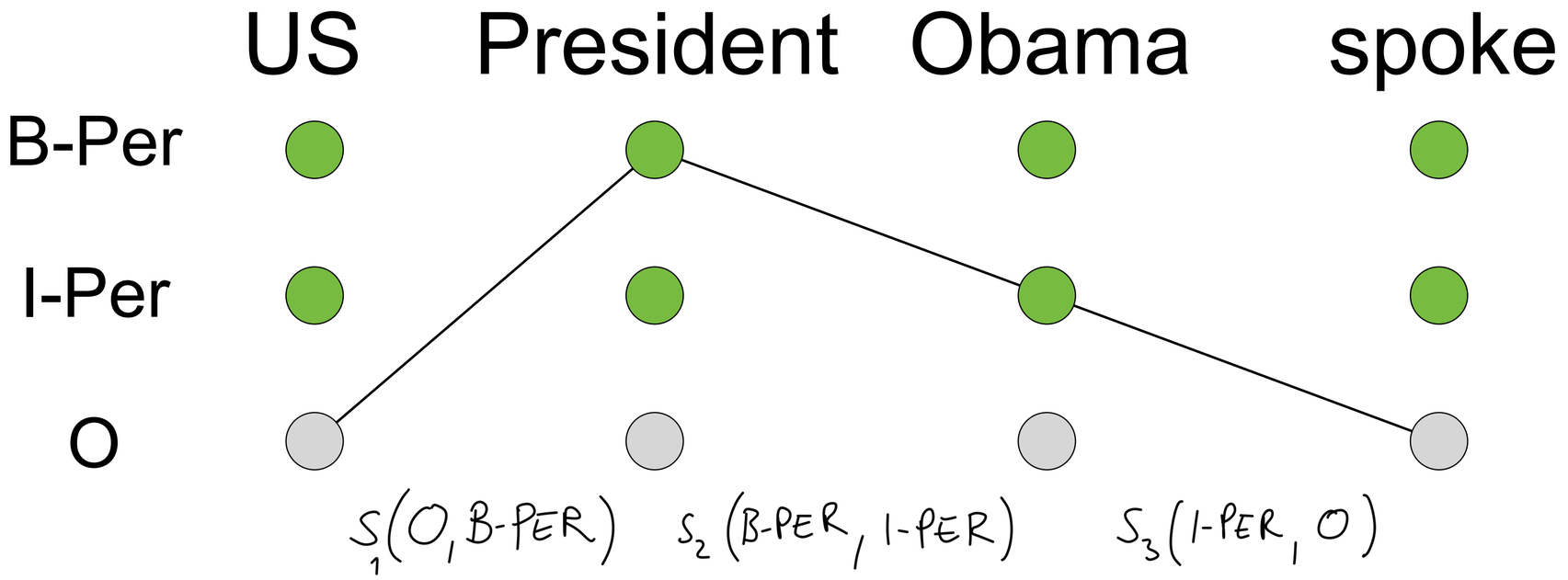

global score: $s(y_1, \dots y_n) = \sum_{i=1}^{n-1} s_i(y_i, y_{i+1})$

Exhaustive Search?

best label sequence: $\tilde{\mathbf{y}} = \underset{y_1, \dots y_n}{\operatorname{argmax}}\, s(y_1, \dots y_n)$

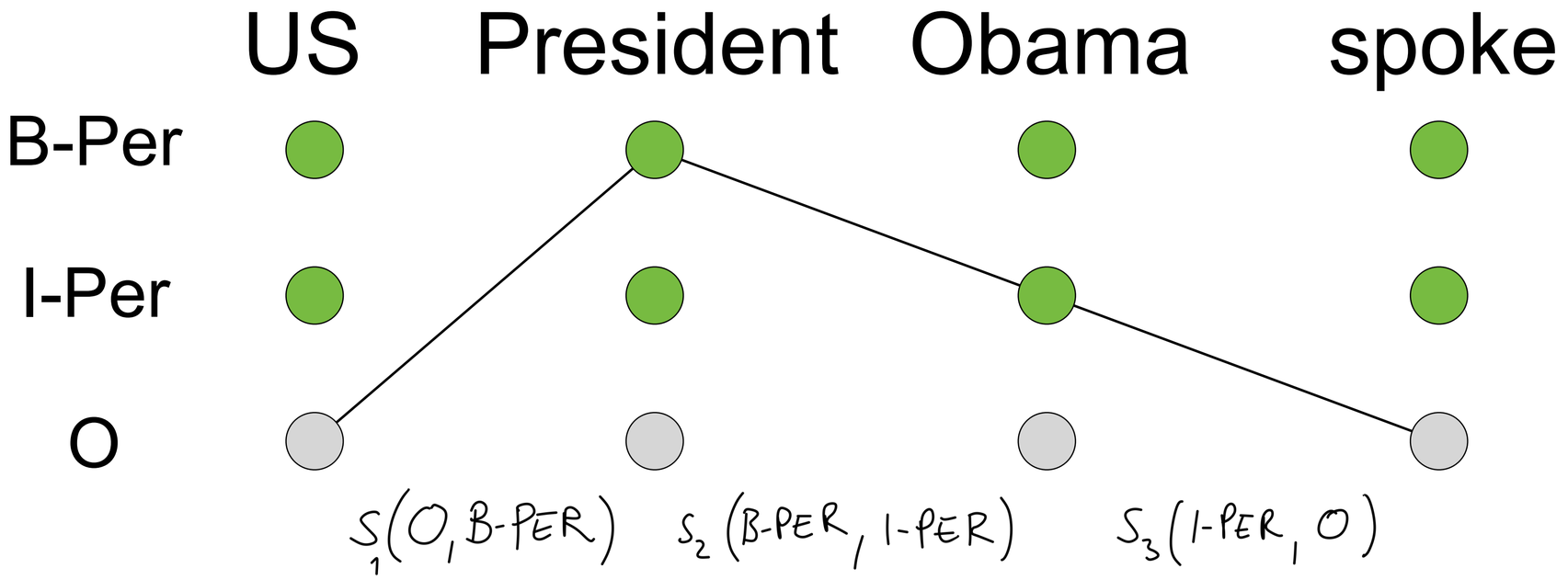
# Viterbi Decoding: Lattice

Representing a label sequence as a path, find highest scoring path



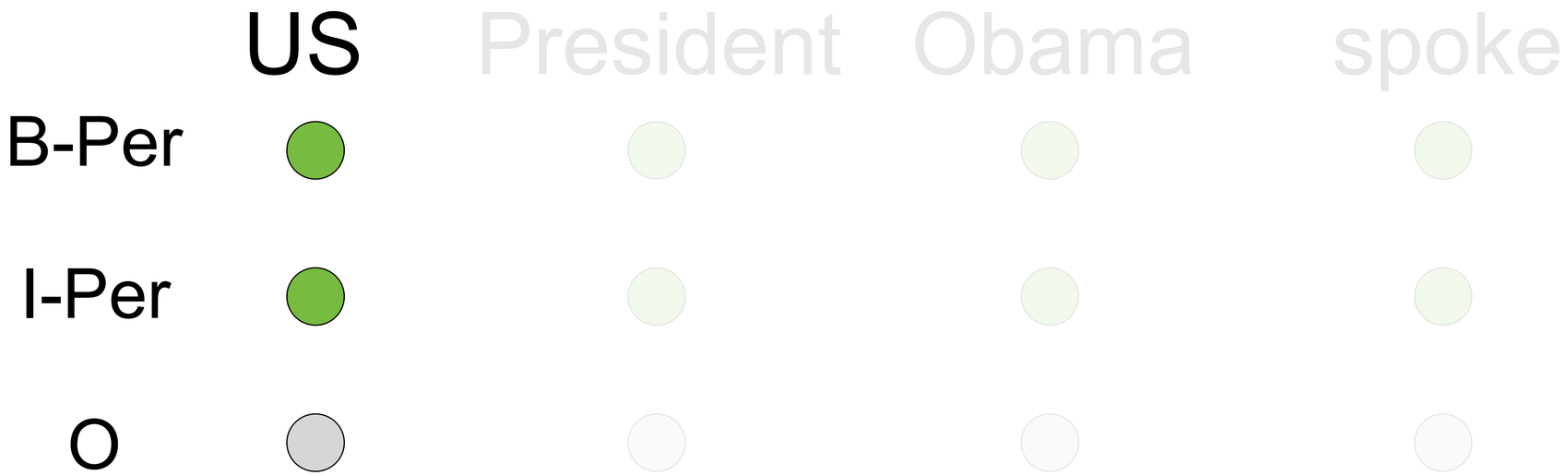
# Viterbi Decoding: Lattice

Representing a label sequence as a path, find highest scoring **path of length 4**



# Viterbi Decoding

For each label, find best path of length 1 ending at this label, and its score

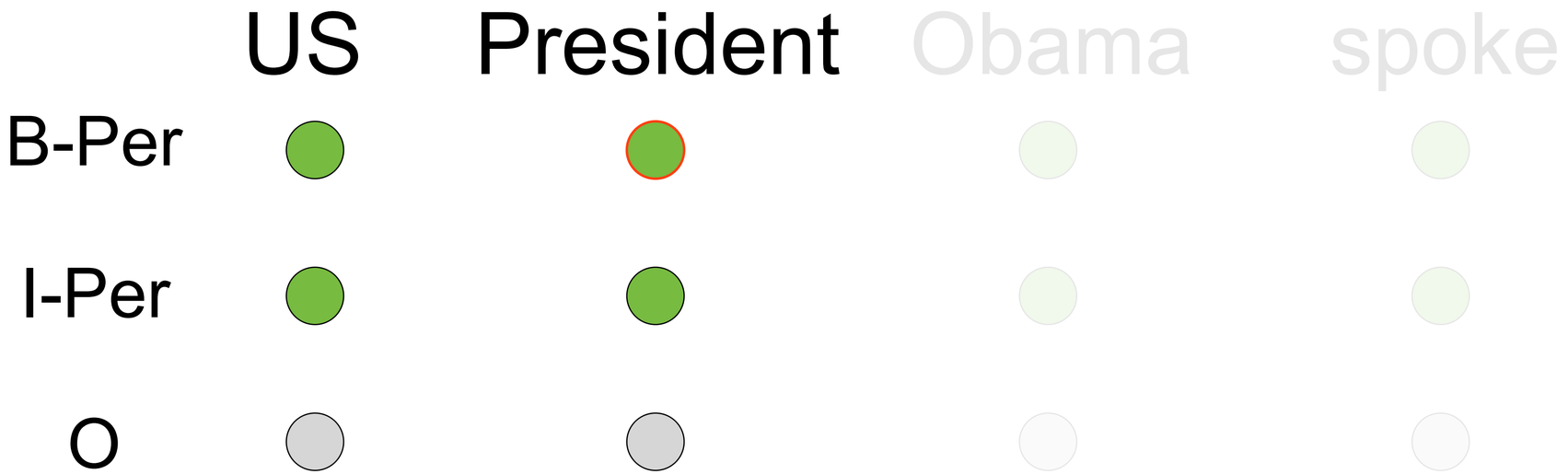



"Message": score of best sequence of length 1 ending in $y_1$

$$\alpha_1(y_1) = 0$$

# Viterbi Decoding

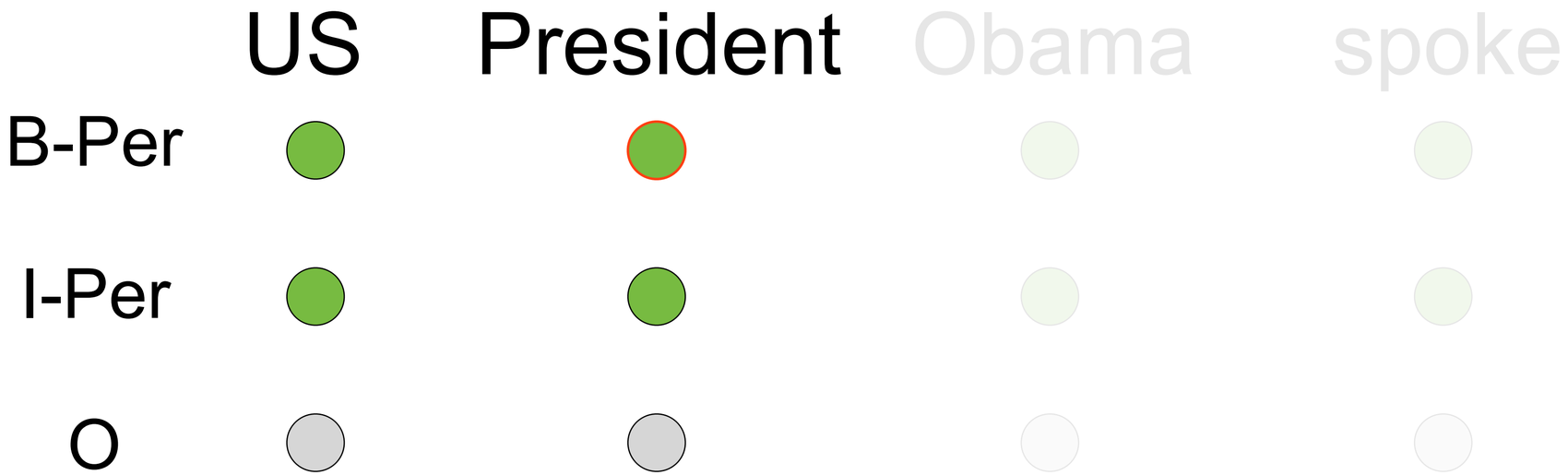
For each label, find best path of length 2 ending at this label, and its score




$$\alpha_2(y_2) = \max_{y_1} S_1(y_1, y_2) + \alpha_1(y_1)$$

# Viterbi Decoding

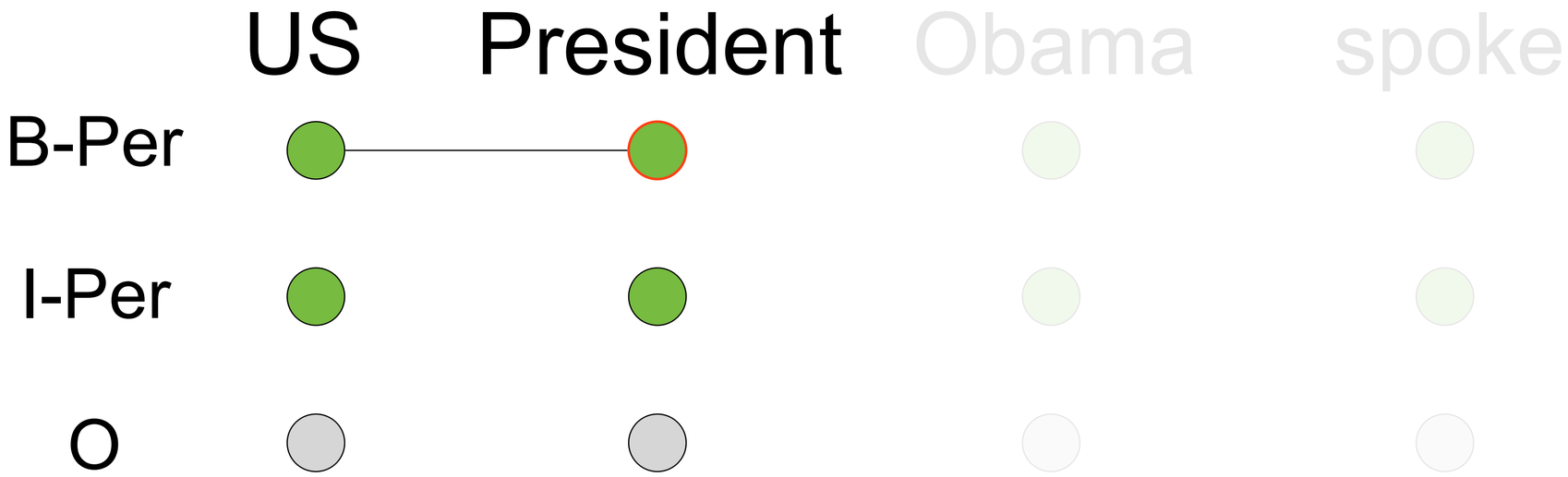
For each label, find best path of length 2 ending at this label, and its score




$$\alpha_2(y_2) = \max_{y_1} s_1(y_1, y_2) + \alpha_1(y_1)$$

# Viterbi Decoding

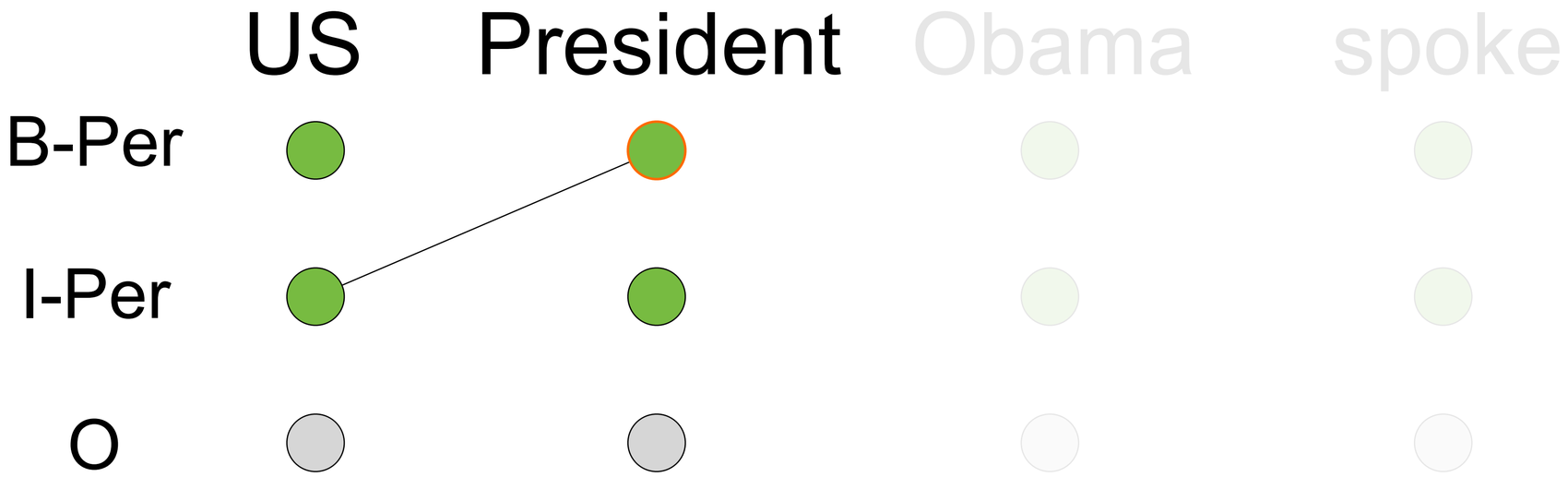
For each label, find best path of length 3 ending at this label, and its score




$$\alpha_2(y_2) = \max_{y_1} S_1(y_1, y_2) + \alpha_1(y_1)$$

# Viterbi Decoding

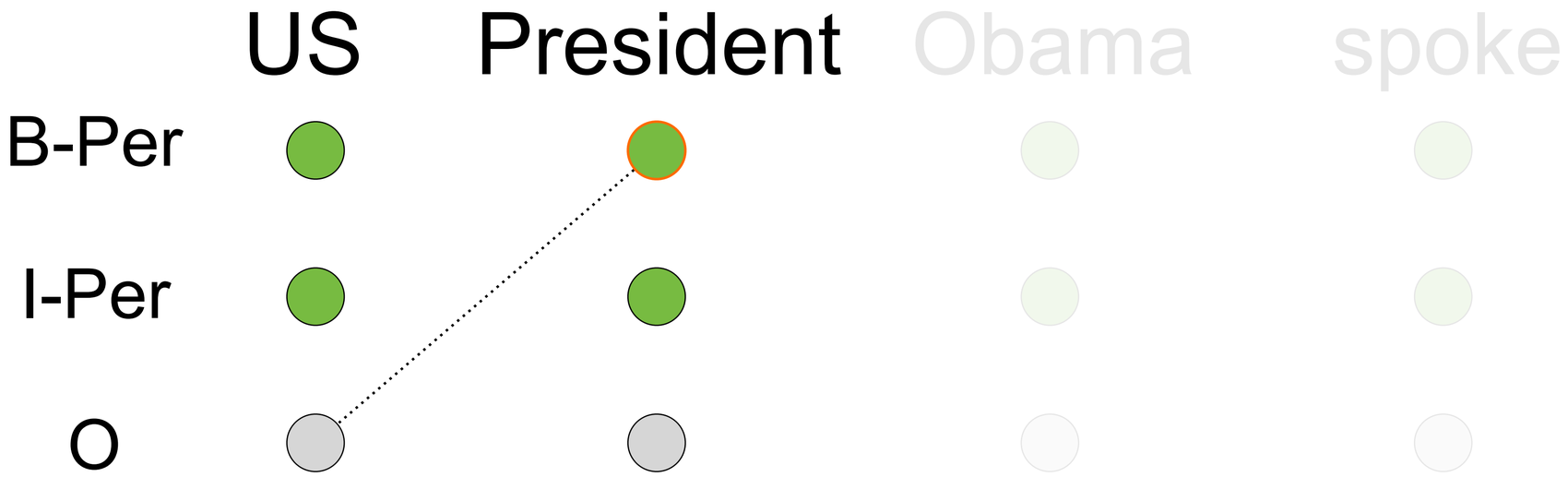
For each label, find best path of length 2 ending at this label, and its score




$$\alpha_2(y_2) = \max_{y_1} S_1(y_1, y_2) + \alpha_1(y_1)$$

# Viterbi Decoding

For each label, find best path of length 2 ending at this label, and its score

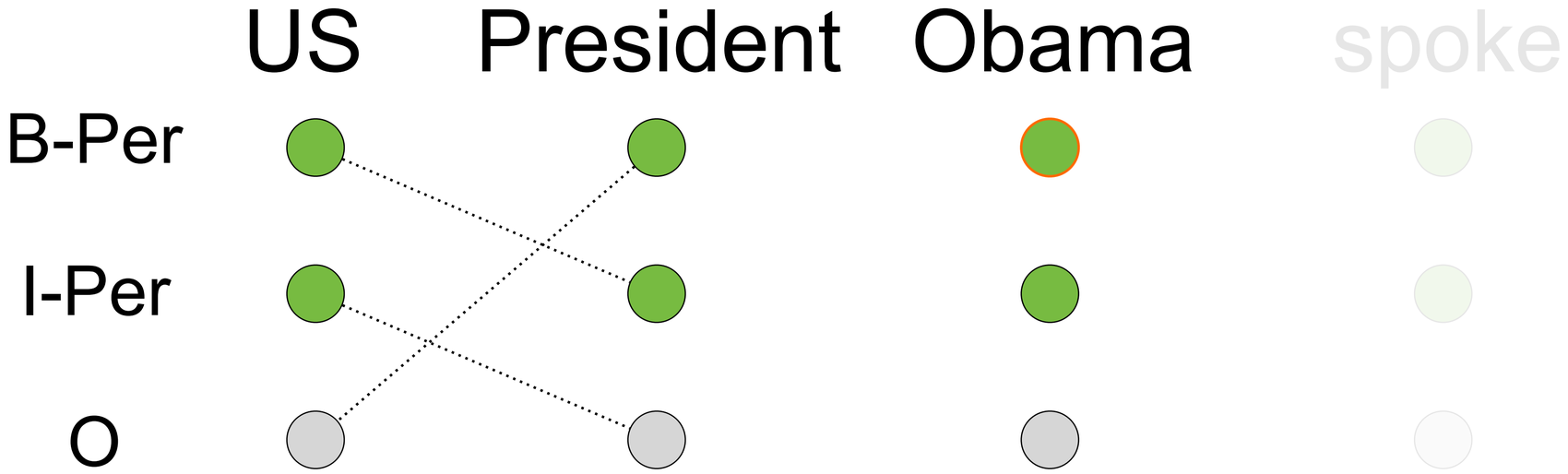



Remember winning $y_1$

$$\alpha_2(y_2) = \max_{y_1} S_1(y_1, y_2) + \alpha_1(y_1)$$

# Viterbi Decoding

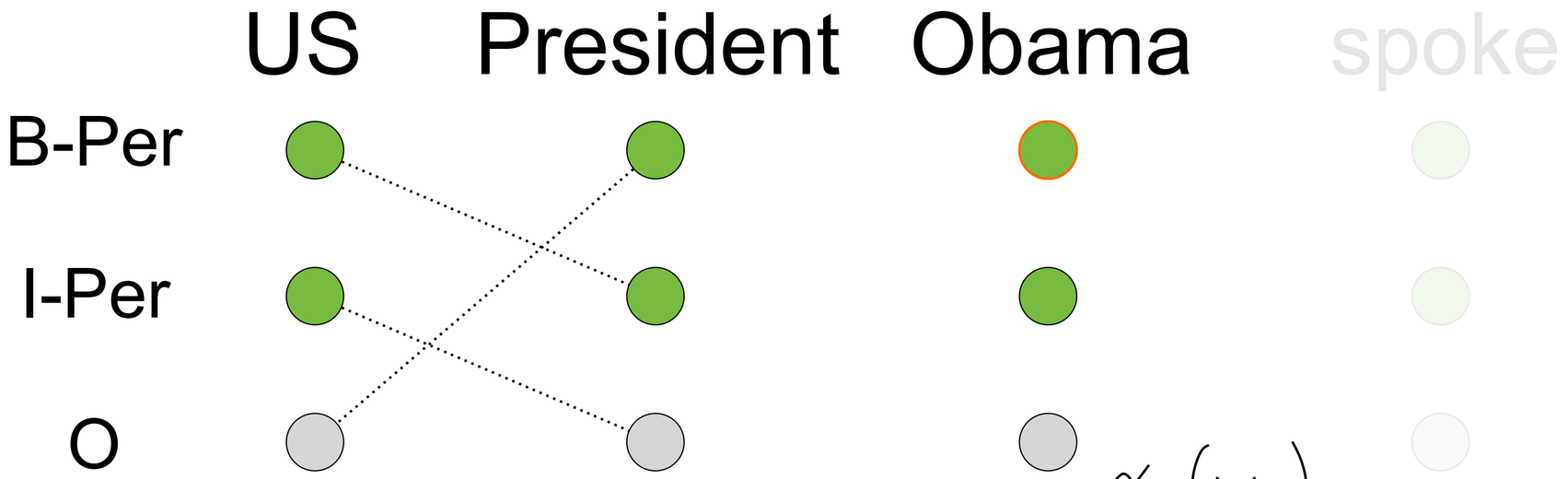
For each label, find best path of length 3 ending at this label, and its score




$$\alpha_3(y_3) = \max_{y_1, y_2} S_1(y_1, y_2) + S_2(y_2, y_3)$$

# Viterbi Decoding

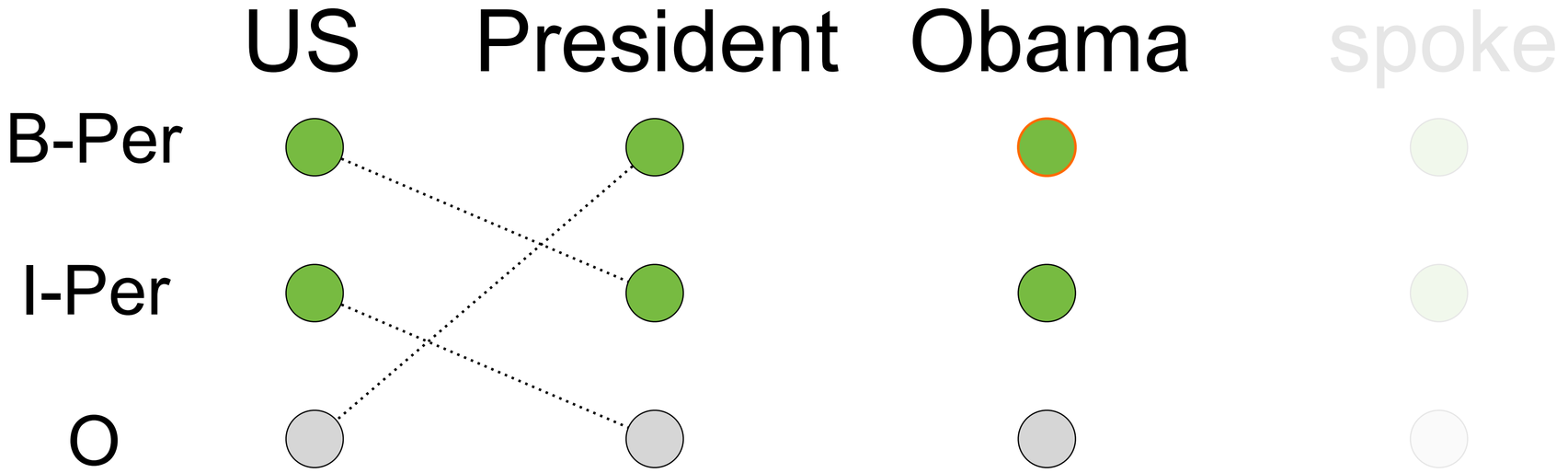
For each label, find best path of length 3 ending at this label, and its score




$$\alpha_3(y_3) = \max_{y_2} \overbrace{\max_{y_1} s_1(y_1, y_2) + s_2(y_2, y_3)}^{\alpha_2(y_2)}$$

# Viterbi Decoding

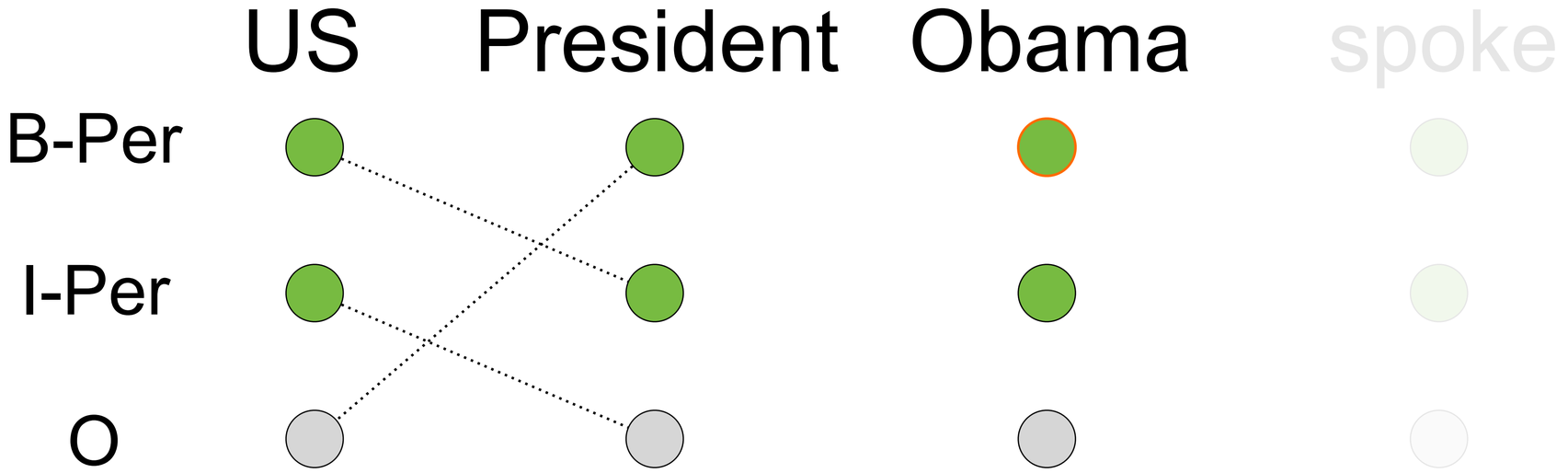
For each label, find best path of length 3 ending at this label, and its score




$$\alpha_3(y_3) = \max_{y_2} \alpha_2(y_2) + s_2(y_2, y_3)$$

# Viterbi Decoding

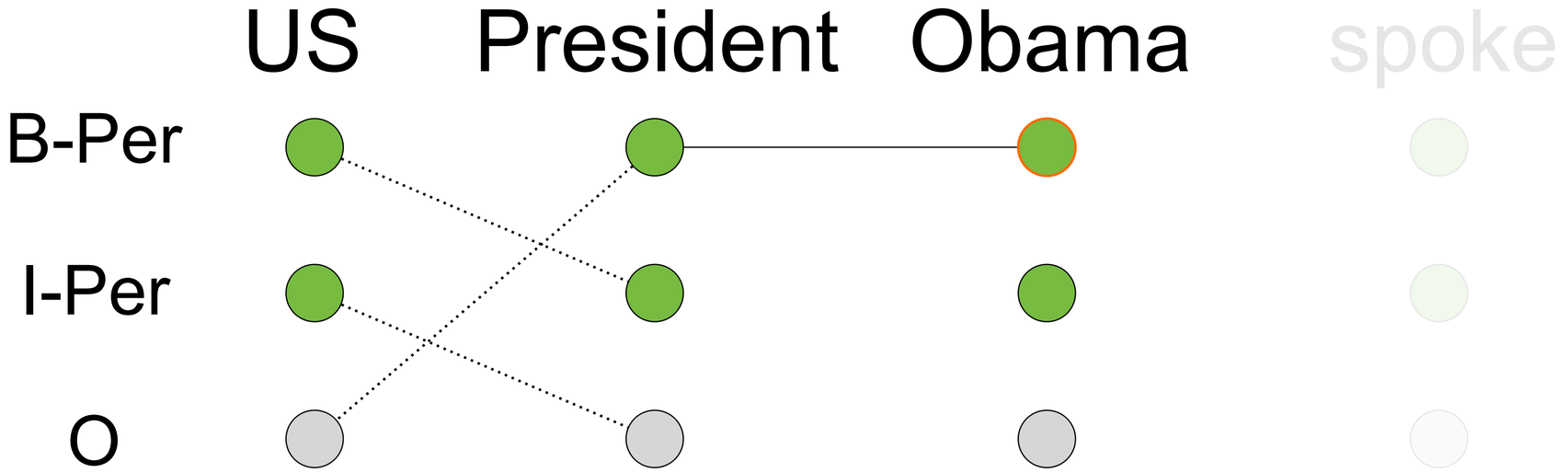
For each label, find best path of length 3 ending at this label, and its score




$$\alpha_3(y_3) = \max_{y_2} \alpha_2(y_2) + s_2(y_2, y_3)$$

# Viterbi Decoding

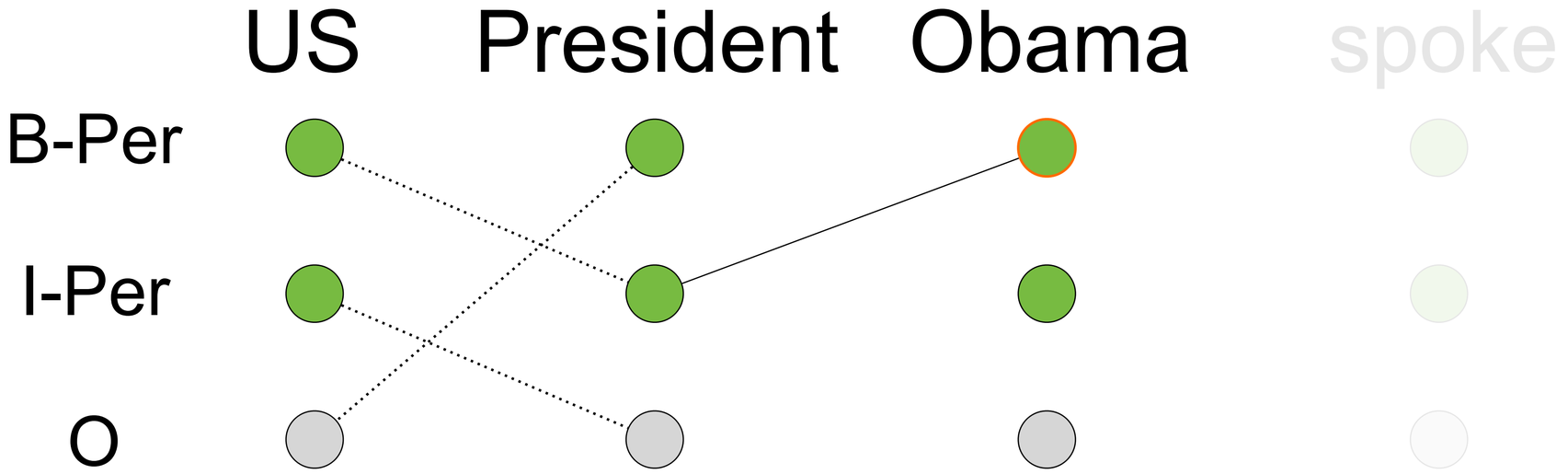
For each label, find best path of length 3 ending at this label, and its score




$$\alpha_3(y_3) = \max_{y_2} \alpha_2(y_2) + s_2(y_2, y_3)$$

# Viterbi Decoding

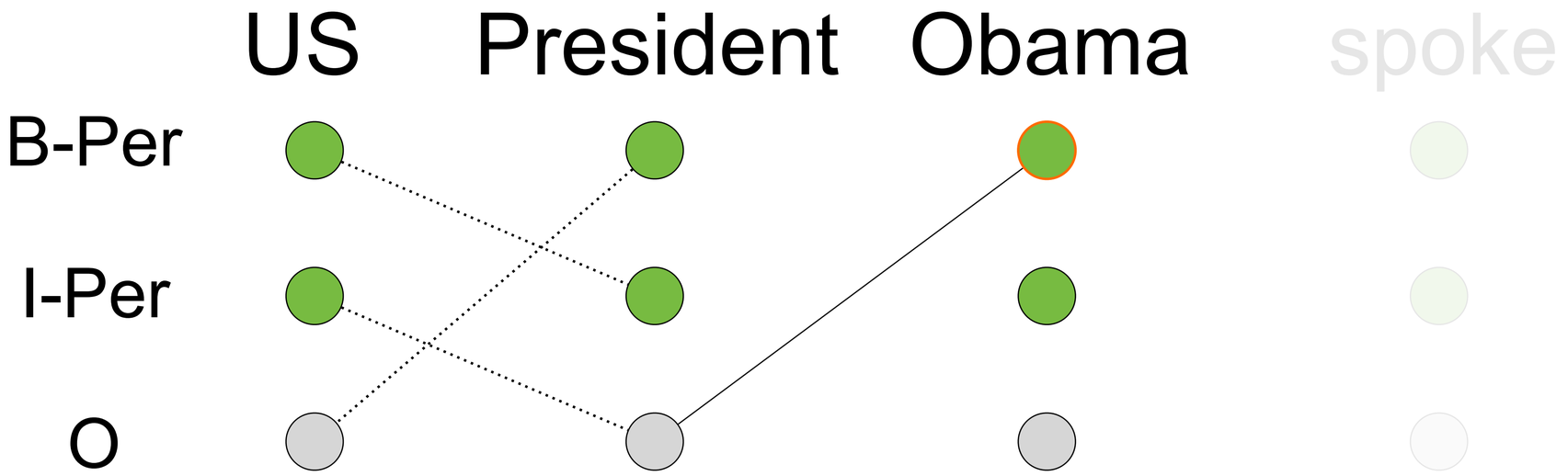
For each label, find best path of length 3 ending at this label, and its score




$$\alpha_3(y_3) = \max_{y_2} \alpha_2(y_2) + s_2(y_2, y_3)$$

# Viterbi Decoding

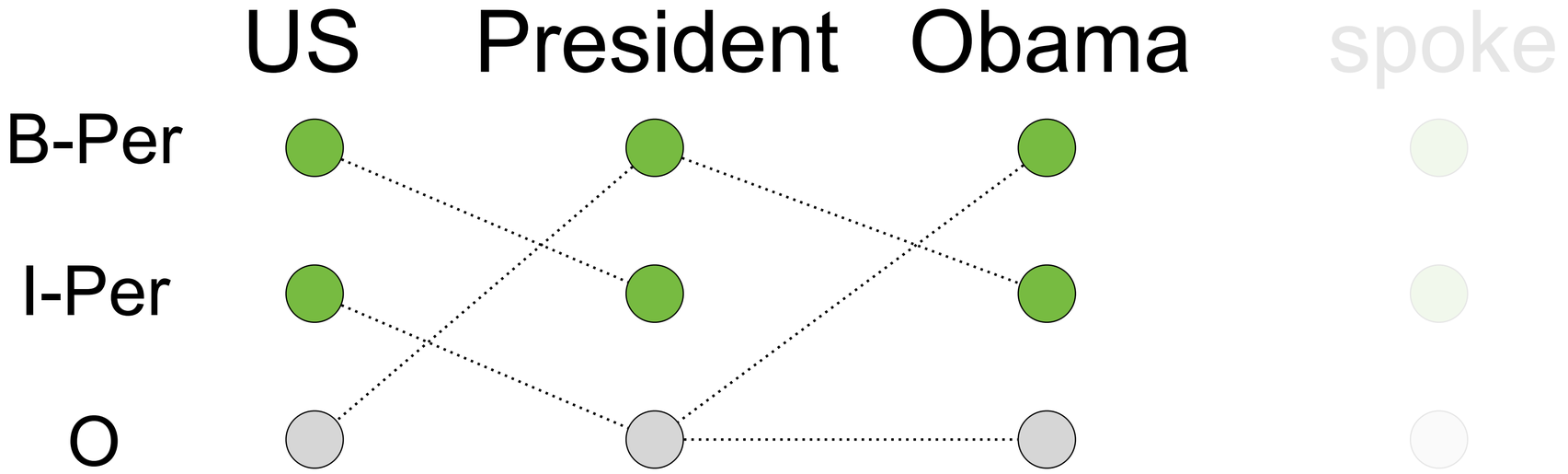
For each label, find best path of length 3 ending at this label, and its score




$$\alpha_3(y_3) = \max_{y_2} \alpha_2(y_2) + s_2(y_2, y_3)$$

# Viterbi Decoding

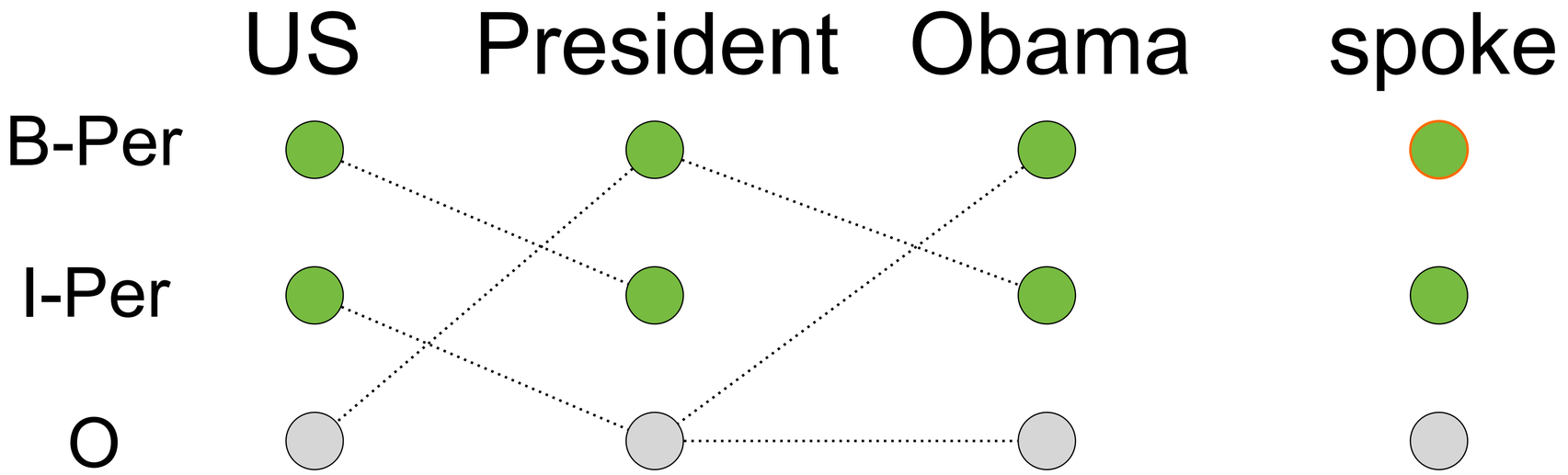
For each label, find best path of length 3 ending at this label, and its score




$$\alpha_3(y_3) = \max_{y_2} \alpha_2(y_2) + s_2(y_2, y_3)$$

# Viterbi Decoding

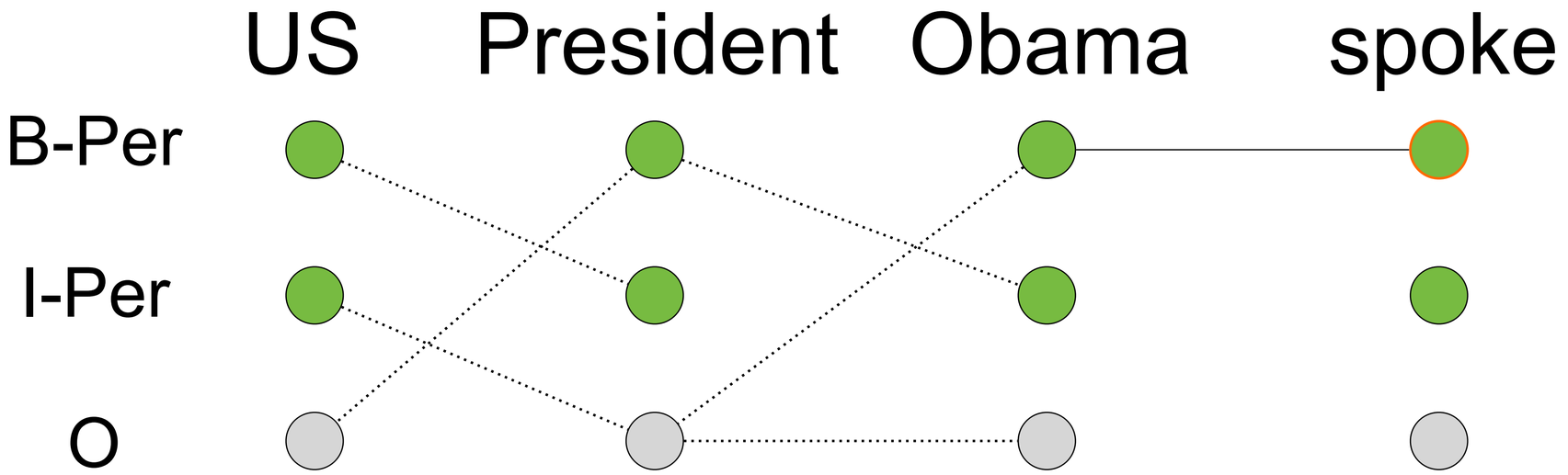
For each label, find best path of length 4 ending at this label, and its score




$$\alpha_4(y_4) = \max_{y_3} \alpha_3(y_3) + s_3(y_3, y_4)$$

# Viterbi Decoding

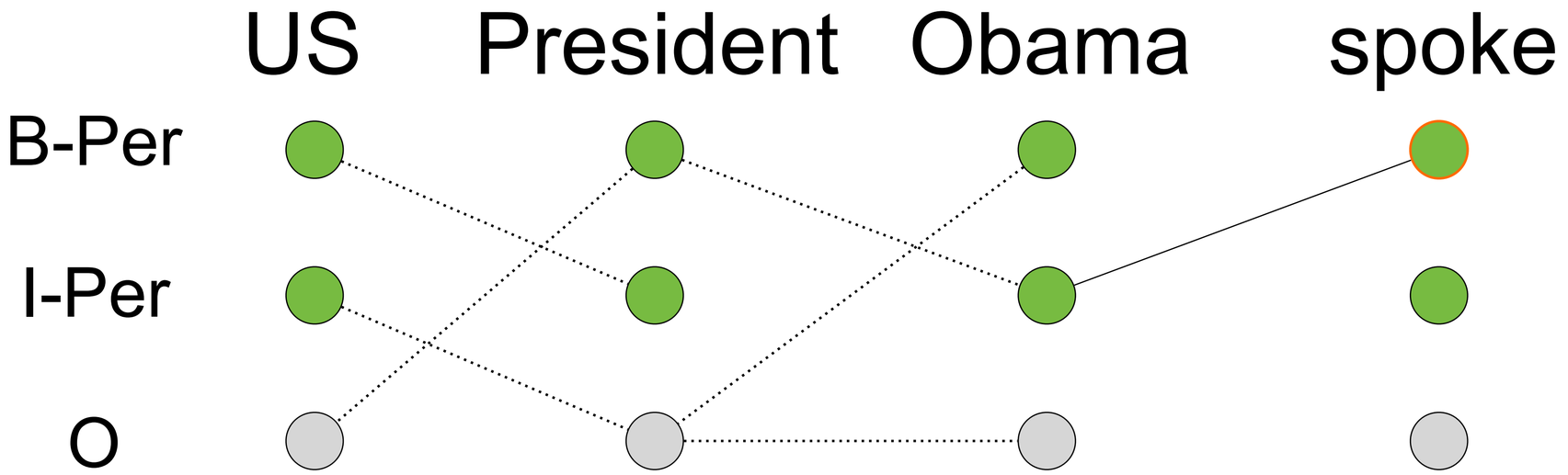
For each label, find best path of length 4 ending at this label, and its score




$$\alpha_4(y_4) = \max_{y_3} \alpha_3(y_3) + s_3(y_3, y_4)$$

# Viterbi Decoding

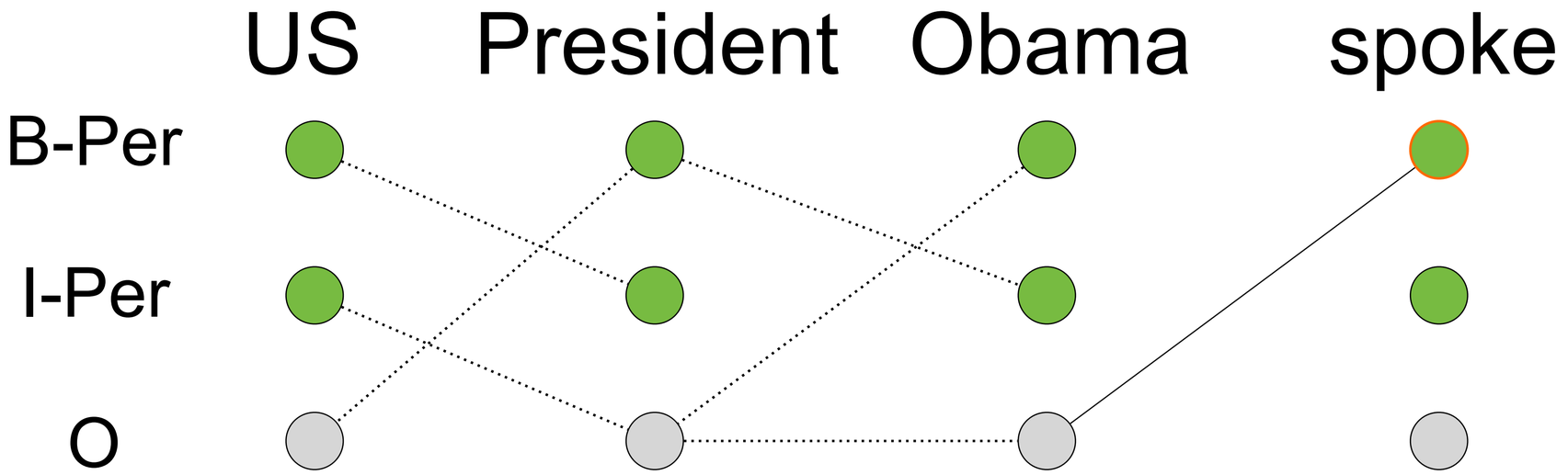
For each label, find best path of length 4 ending at this label, and its score




$$\alpha_4(y_4) = \max_{y_3} \alpha_3(y_3) + s_3(y_3, y_4)$$

# Viterbi Decoding

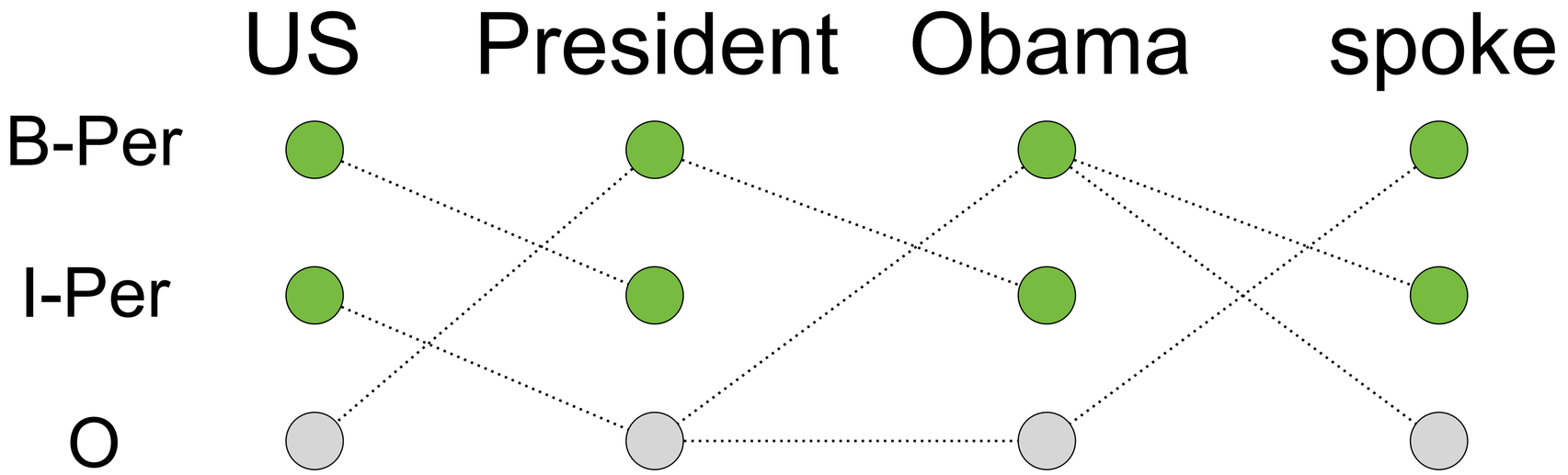
For each label, find best path of length 4 ending at this label, and its score




$$\alpha_4(y_4) = \max_{y_3} \alpha_3(y_3) + s_3(y_3, y_4)$$

# Viterbi Decoding

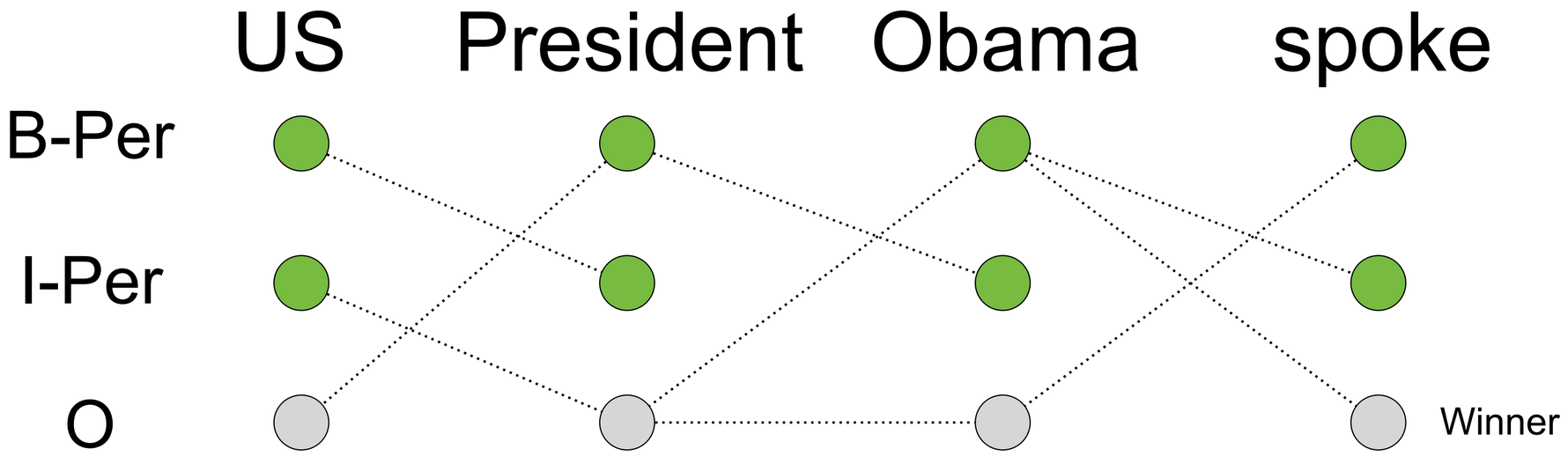
For each label, find best path of length 4 ending at this label, and its score




$$\alpha_4(y_4) = \max_{y_3} \alpha_3(y_3) + s_3(y_3, y_4)$$

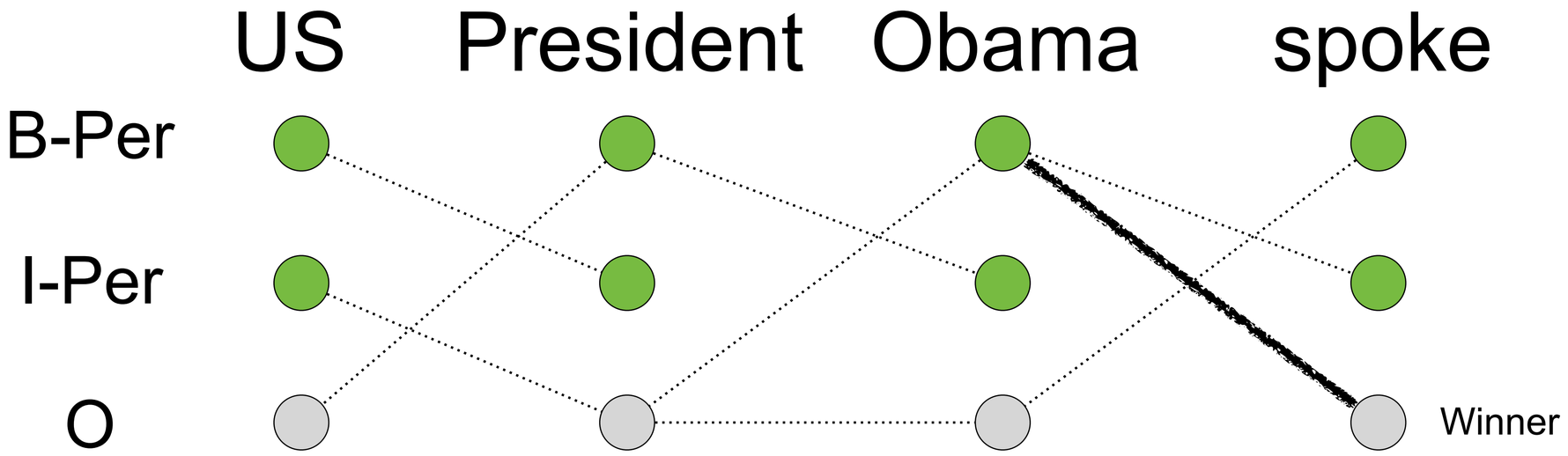
# Viterbi Decoding: Backtracking

You got a winner! Find the path that lead to it...



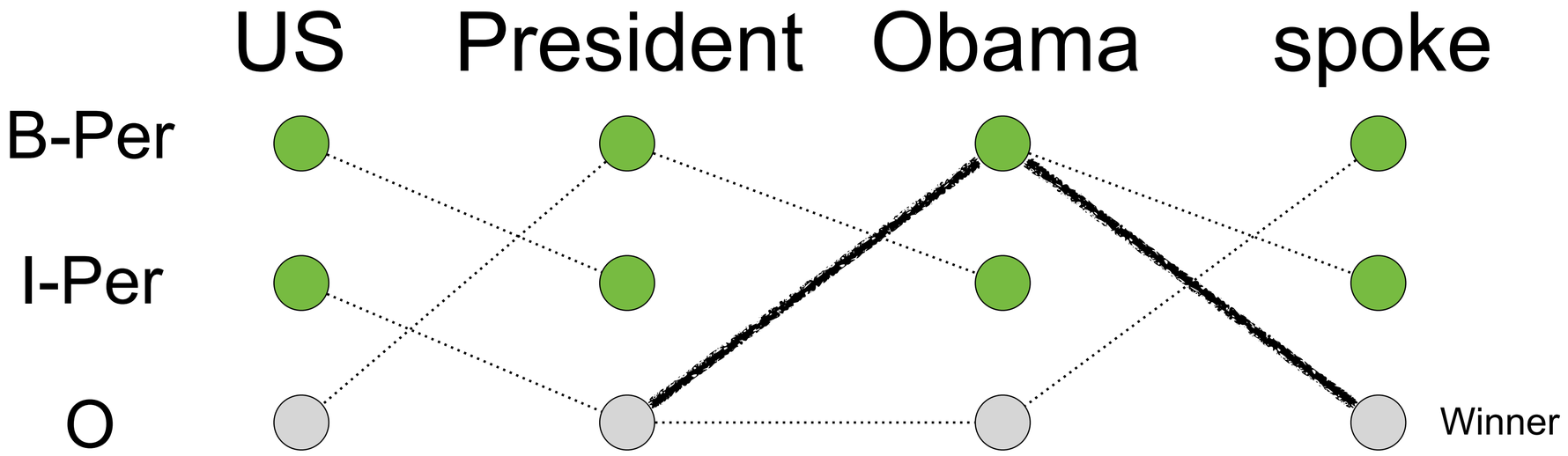
# Viterbi Decoding: Backtracking

Which was the maximizing node coming into label 4?



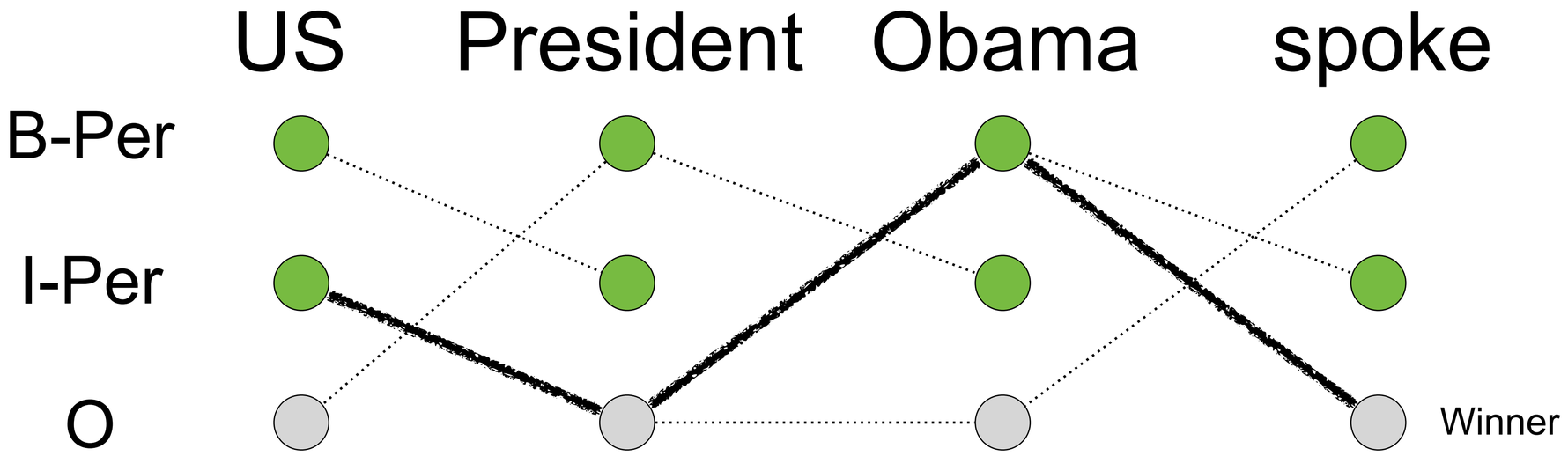
# Viterbi Decoding: Backtracking

Which was the maximizing node coming into label 3?



# Viterbi Decoding: Backtracking

Which was the maximizing node coming into label 2?




Runtime?

# Viterbi Summary

① Calculate scores for each token, left-to-right

- $\alpha_i(y_i) = \max_{y_{i-1}} \alpha(y_{i-1}) + (y_{i-1}, y_i)$
- Remember winning $y_{i-1}$ for each $i$

② Pick $y_i$ with highest $\alpha_i(y_i)$, right-to-left

# Training

We like our LogLikelihood loss!

$$\text{Loss}: \; -\sum_{(\mathbf{x},\mathbf{y})\in D} \log p(\mathbf{y}\mid\mathbf{x})$$

we only have ↓

$$p(y_1,\dots y_n \mid \mathbf{x}) = \frac{e^{s(y_1,\dots y_n\mid \mathbf{x})}}{\sum_{y'_1,\dots y'_n} e^{s(y'_1,\dots y'_n\mid \mathbf{x})}} = Z(\mathbf{x})$$

"Softmax"?

Complexity?

Partition Function

Conditional Random Field

# Marginal Probabilities

Calculating Partition Function $\approx$ Calculating Marg. Probabilities

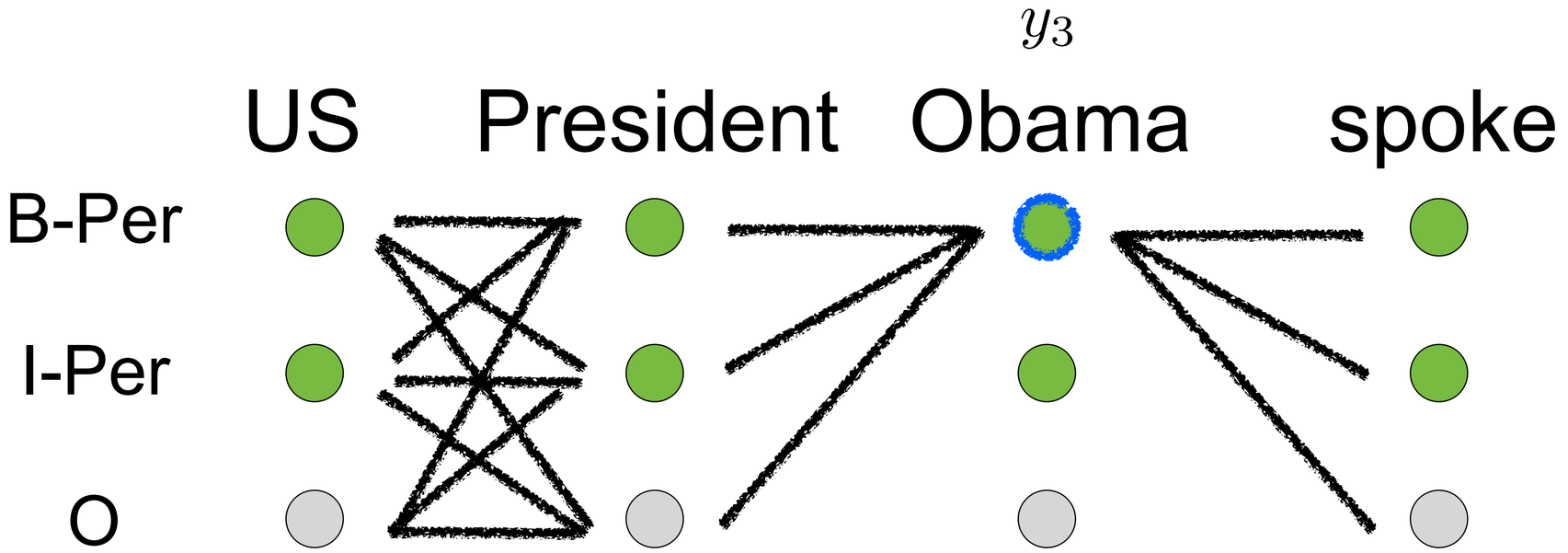

spoke at the White House

How likely is label $y$ at position $j$

$$p_j(y \mid \mathbf{x}) = \sum_{\substack{\mathbf{y} \\ y_j = y}} p(\mathbf{y} \mid \mathbf{x}) \propto \frac{1}{Z(\mathbf{x})} \sum_{\substack{\mathbf{y} \\ y_j = y}} e^{s(\mathbf{y} \mid \mathbf{x})}$$

# Marginal Probabilities

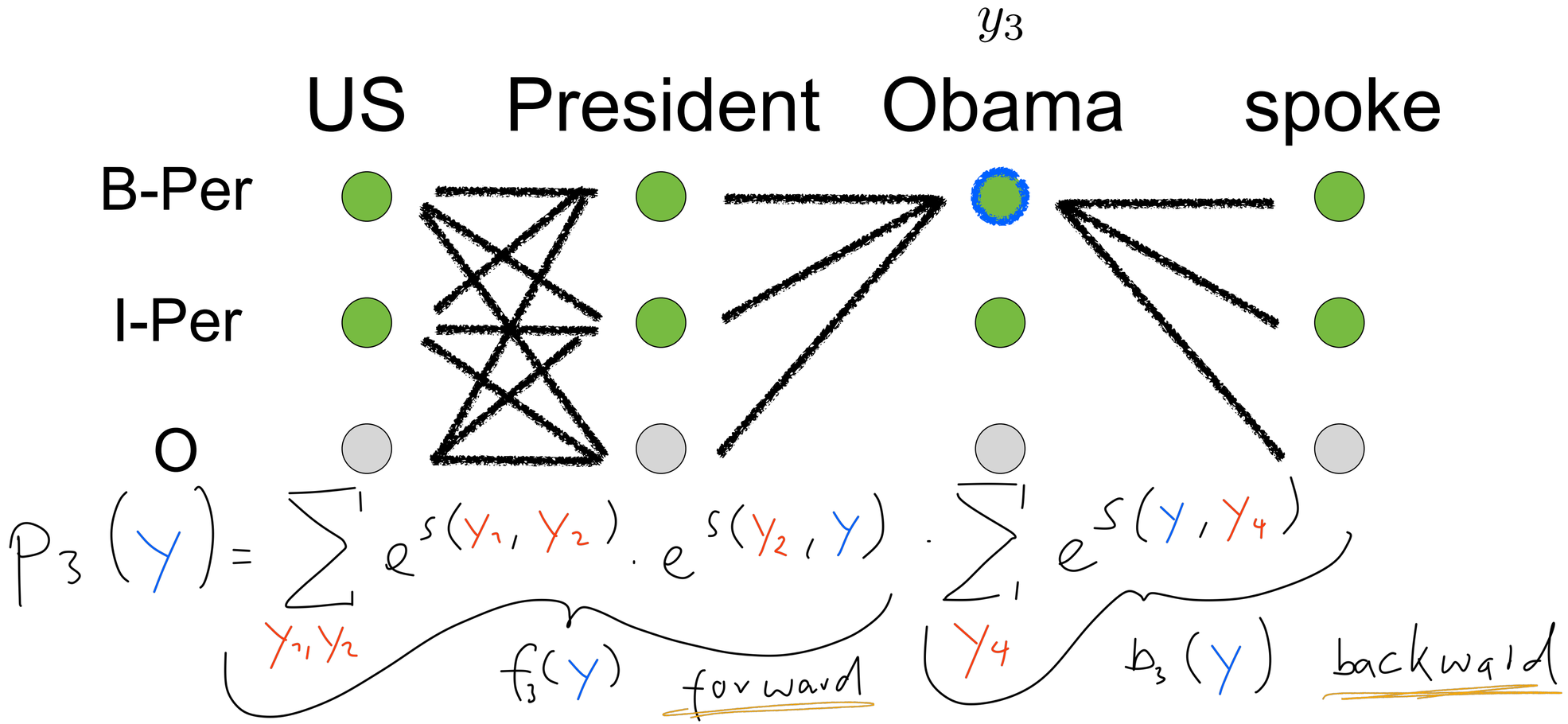
Find the marginal probability of label at token 3 to be $y_3$




$$p_3(y) \propto \sum_{y_1, y_2, y_3 = y_1, y_4} e^{s(\mathbf{y})} = \sum_{y_1, y_2, y_3 = y_1, y_4} \prod_{i}^{n-1} e^{s_i(y_i, y_{i+1})}$$

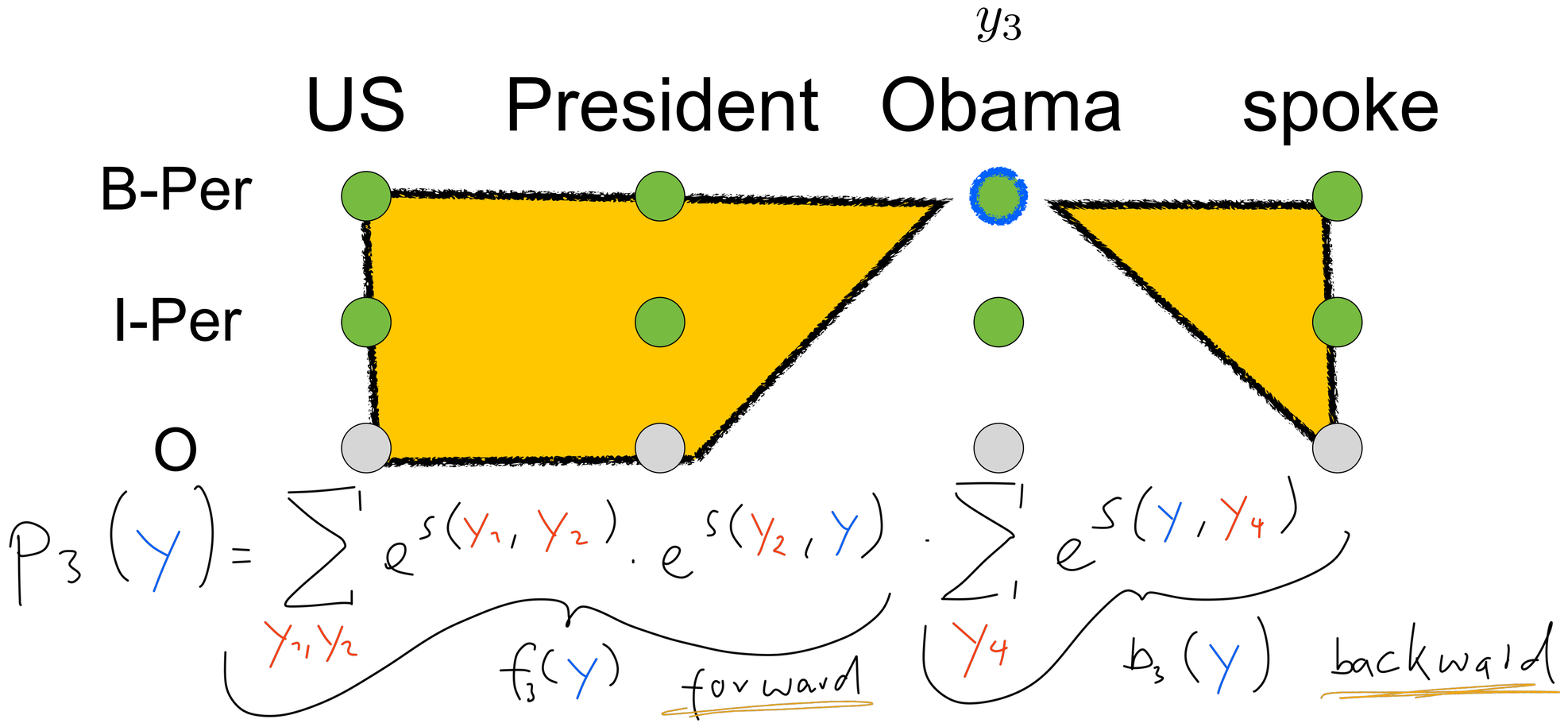
# Forward-Backward

Sum over all paths that end in $y_3$ and start in $y_3$



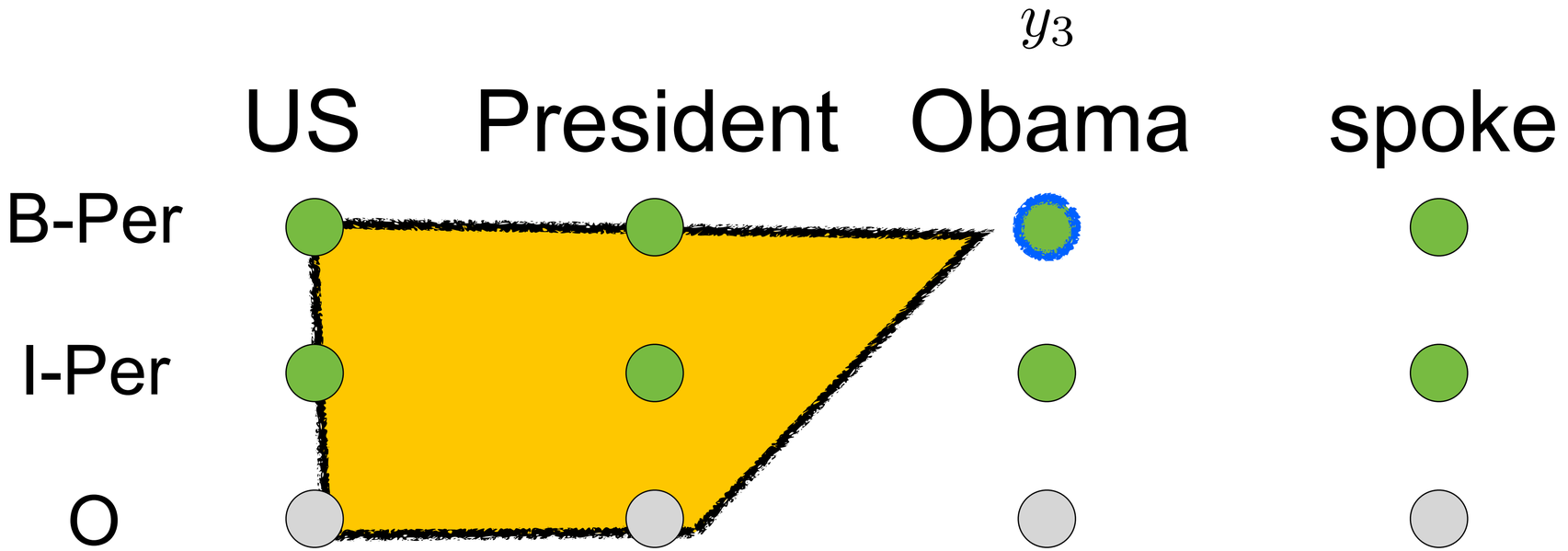
# Forward-Backward

Marginal probability of variable is product of **forward** and **backward** message



# Forward-Backward

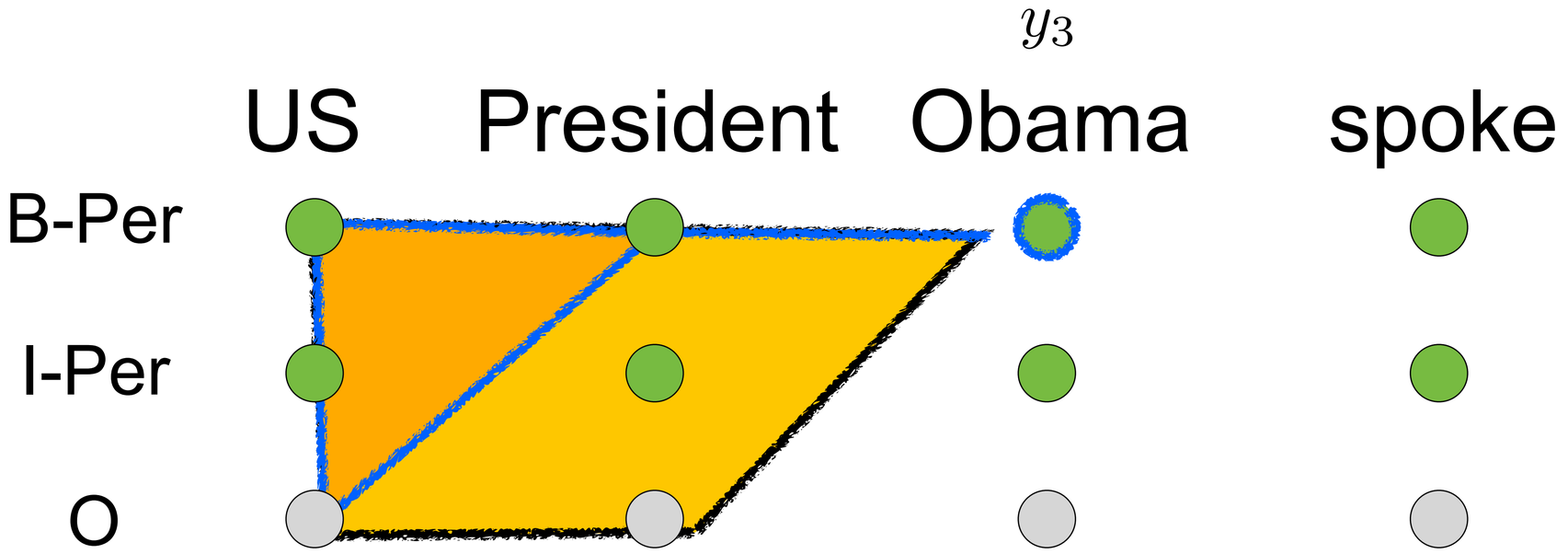
Let’s focus on the forward message (backward equivalent)




$$f_3(y) = \sum_{y_1, y_2} e^{s(y_2, y)} \cdot e^{s(y_1, y_2)}$$

# Forward-Backward

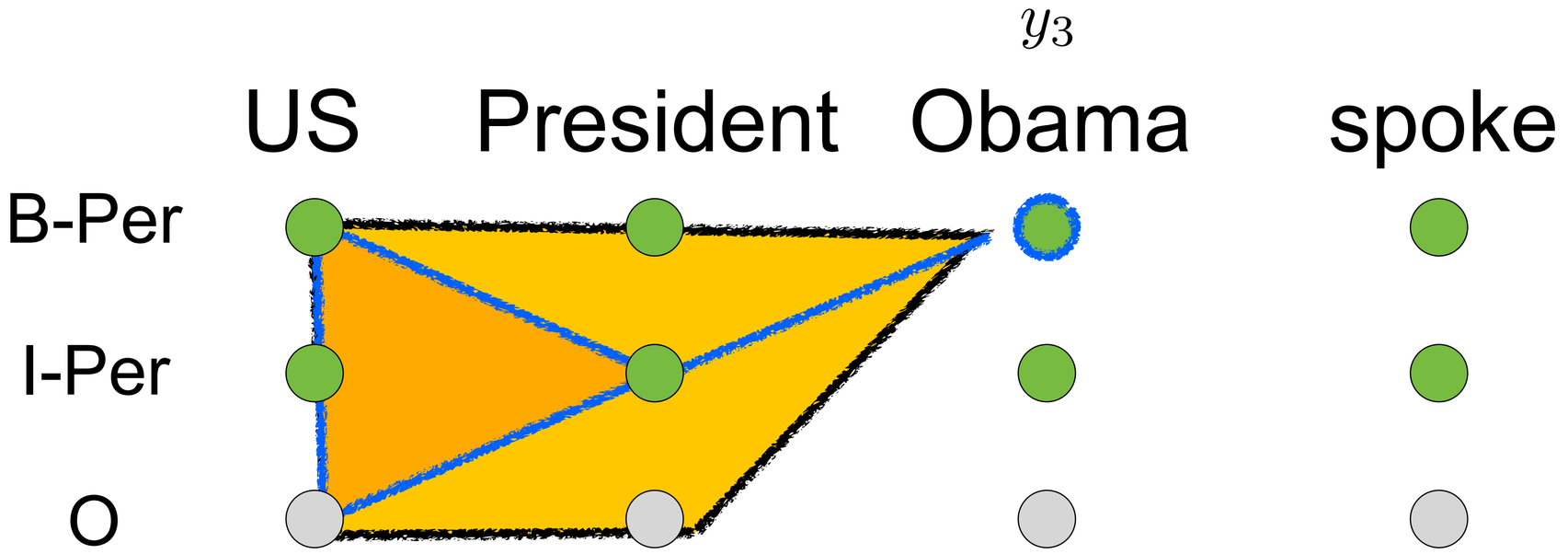
For B-PER multiply transition potential with total mass of paths going into B-PER




$$f_3(y) = \sum_{y_2} e^{s(y_2, y)} \cdot \sum_{y_1} e^{s(y_1, y_2)}$$

# Forward-Backward

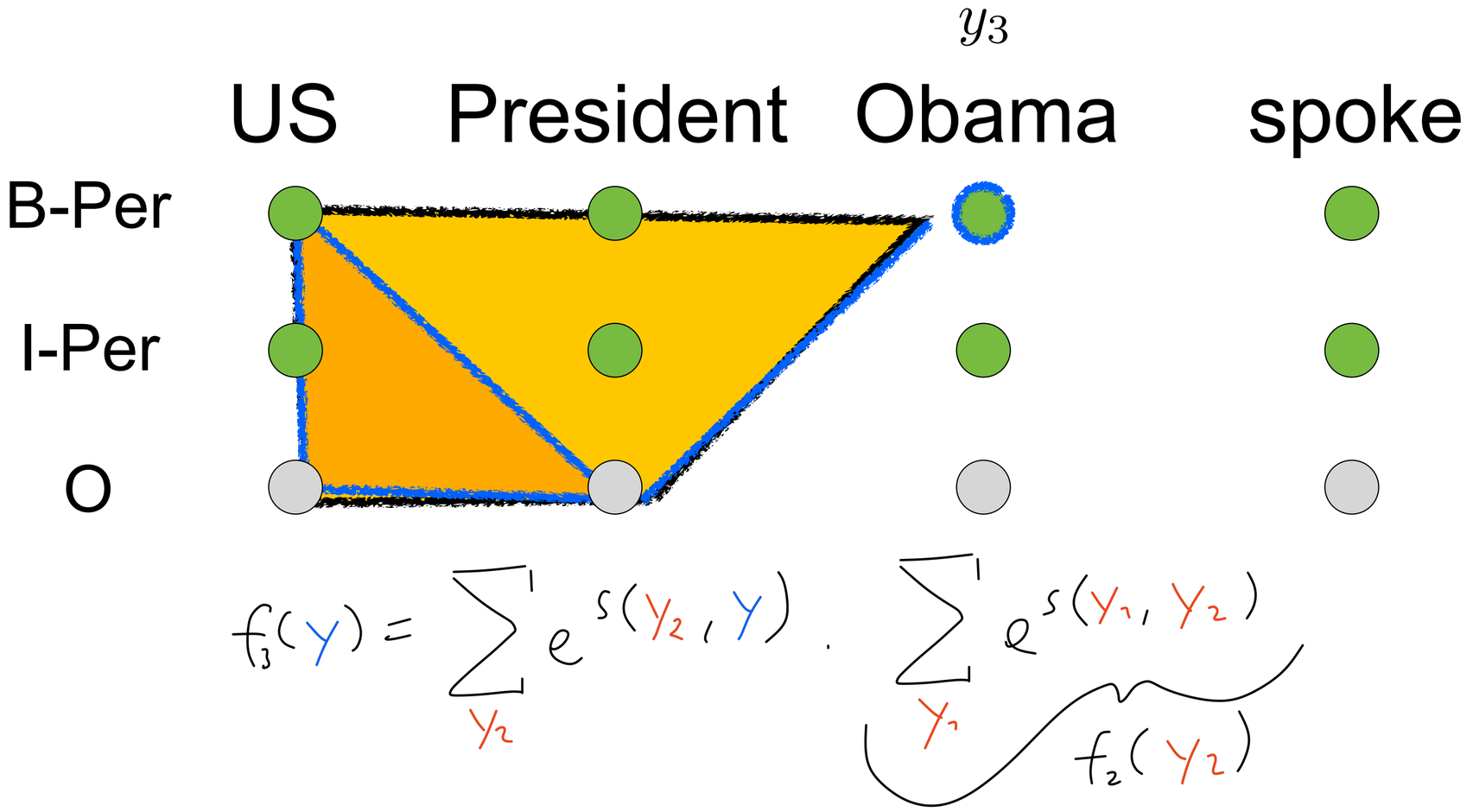
For I-PER multiply transition potential with total mass of paths going into I-PER



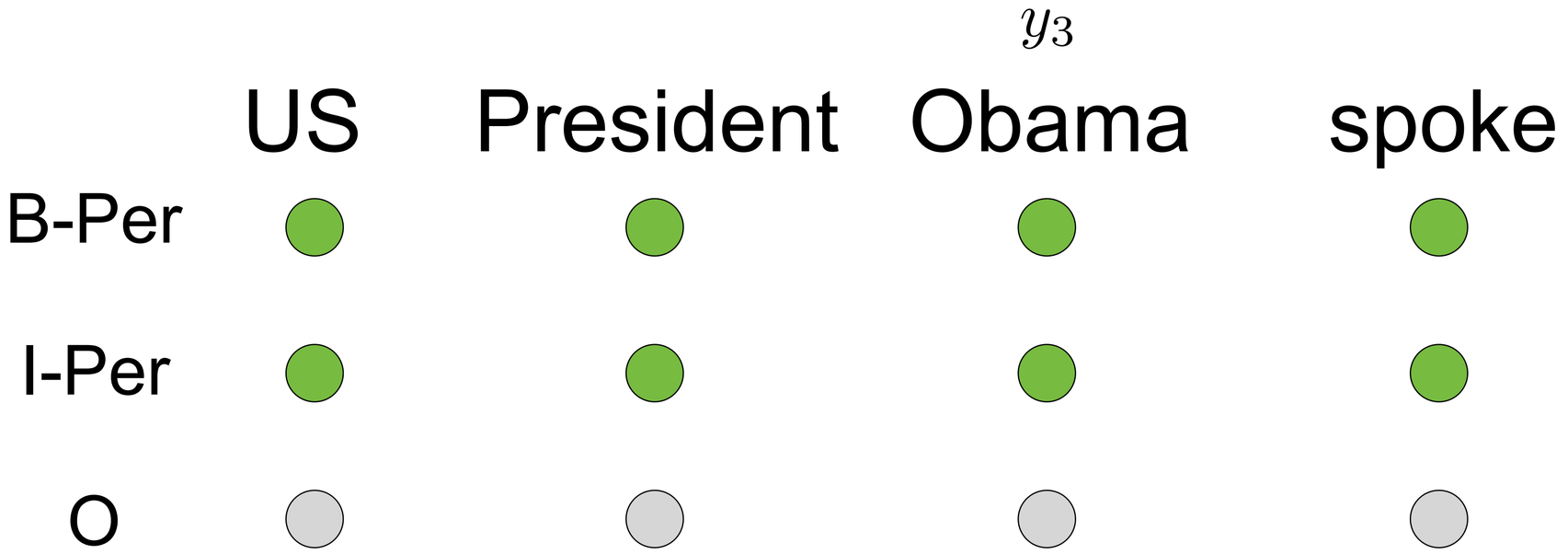

$$f_3(y) = \sum_{y_2} e^{s(y_2, y)} \cdot \sum_{y_1} e^{s(y_1, y_2)}$$

# Forward-Backward

For O multiply transition potential with total mass of paths going into O

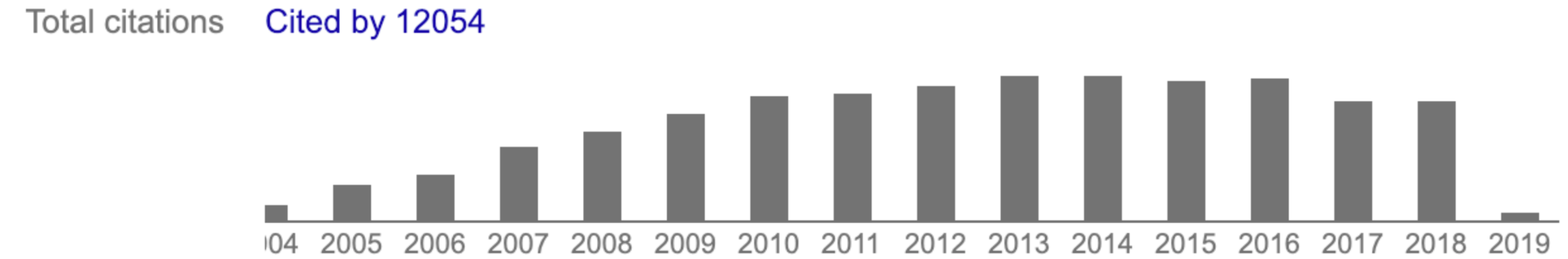


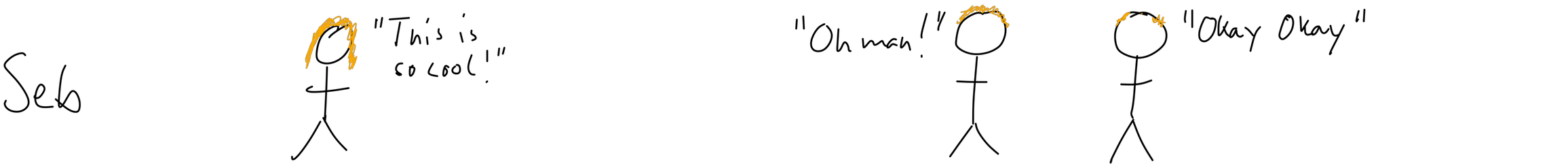
# Forward-Backward

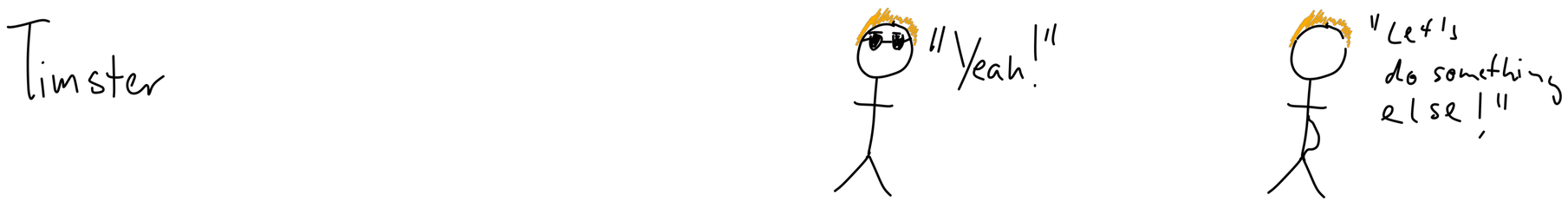



$$f_i(y) = \sum_{y_{i-1}} e^{s(y_{i-1}, y)} \cdot f_{i-1}(y_{i-1})$$

# Some History: CRF



# References

- Neural Sequence Labelling
  - J&M Chapter 9
  - Goldberg Chapter 8
  - Supervised Sequence Labelling with Recurrent Neural Networks, Alex Graves PhD Thesis
  - Design Challenges and Misconceptions in Neural Sequence Labeling, Jie Yang, Shuailong Liang, Yue Zhang, COLING 2018
- Viterbi, Forward-Backward
  - J&M Chapter 8, Appendix 3
  - Goldberg Chapter 8
  - Lafferty, J., McCallum, A., Pereira, F. (2001). "Conditional random fields: Probabilistic models for segmenting and labeling sequence data". ICML 2001