Data Sparsity in Natural Language Processing (NLP)

Lev Ratinov
UIUC
with: Dan Roth, Joseph Turian, Yoshua Bengio

Sample NLP problem

- “Peter Young scores the opener.”
- “Young American soldiers return home.”

Sample NLP problem

- f(Young|“Peter Young scores the opener.”) → PERSON
- f(Young|“Young American soldiers return home.”) → O

Sample NLP problem

- f(Young|“Peter Young scores the opener.”) → PERSON

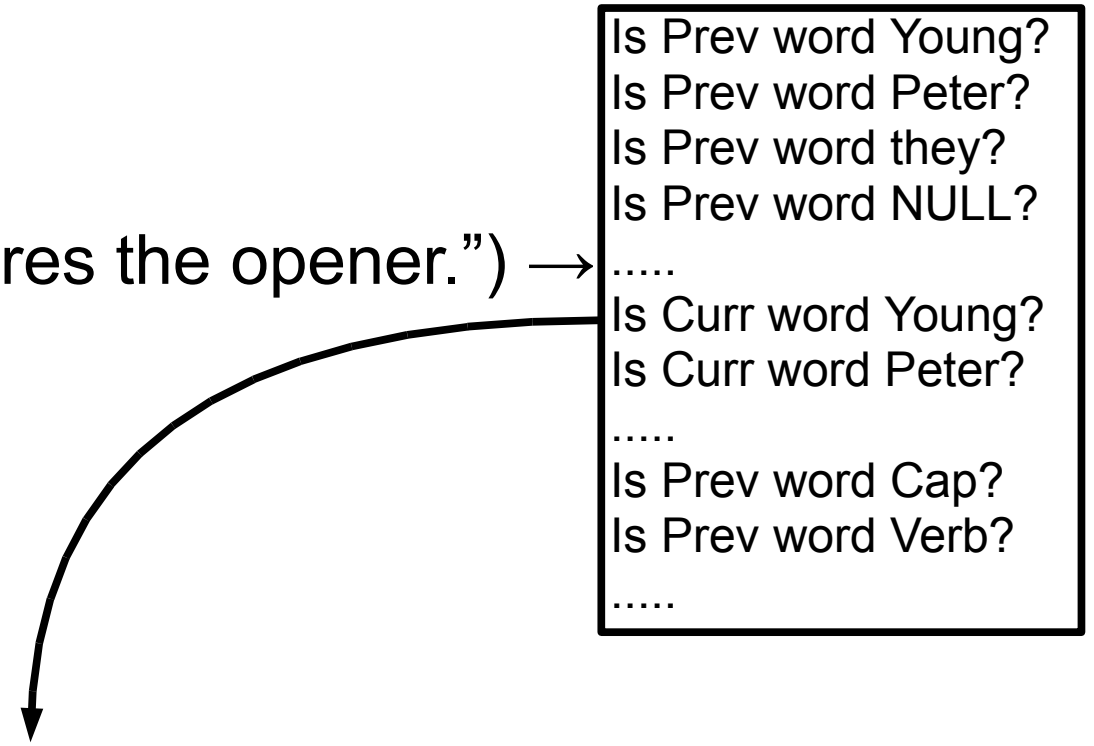
Sample NLP problem

- f(Young|“Peter Young scores the opener.”) → PERSON
- ∅(Young|“Peter Young scores the opener.”) →

Is Prev word Young?
Is Prev word Peter?
Is Prev word they?
Is Prev word NULL?
.....
Is Curr word Young?
Is Curr word Peter?
.....
Is Prev word Cap?
Is Prev word Verb?
.....

Sample NLP problem

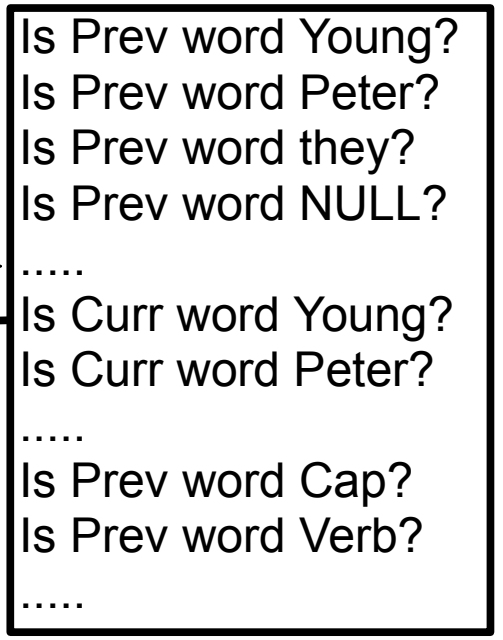
- f(Young|“Peter Young scores the opener.”) → PERSON
- ∅(Young|“Peter Young scores the opener.”) →



f(0010000110000000...000) → PERSON

Sample NLP problem

- f(Young|“Peter Young scores the opener.”) → PERSON
- ∅(Young|“Peter Young scores the opener.”) →



f(00100001100000000...000) → PERSON

SPARSE!!! (but huge models)

NLP

- Words words words ---> Statistics

NLP

- Words words words ---> Statistics
- Words words words ---> Statistics

NLP

- Words words words ---> Statistics
- Words words words ---> Statistics
- Words words words ---> Statistics

NLP

- Words words words ---> Statistics
- Words words words ---> Statistics
- Words words words ---> Statistics
- How do we handle/represent words?

NLP

- Not so well...
- We do well when we see the words we have **already seen** in training examples and have enough statistics about them.
- When we see a word we haven't seen before, we try:
  - Part of speech abstraction
  - Prefixes/suffixes/number/capitalized abstraction.
- We have a lot of text! Can we do better?

Can we do better?

- Yes
- Running example- Named Entity Recognition.
- This work applies to all Machine Learning applications, where
  - There is data sparsity.
  - There is a lot of unlabeled data.

Named Entity Recognition (NER)

SOCCER - [PER BLINKER] BAN LIFTED .
[LOC LONDON] 1996-12-06
[MISC Dutch] forward [PER Reggie Blinker] had his indefinite suspension lifted by [ORG FIFA] on Friday and was set to make his [ORG Sheffield Wednesday] comeback against [ORG Liverpool] on Saturday . [PER Blinker] missed his club 's last two games after [ORG FIFA] slapped a worldwide ban on him for appearing to sign contracts for both [ORG Wednesday] and [ORG Udinese] while he was playing for [ORG Feyenoord] .

- Why is NER Important?
  - Many NLP tasks are modelled similarly.
  - Entities are key in text understanding.

Supervised learning in NLP

- Training examples:
  - "...a damaging row between [LOC Britain] and the [ORG EU] , which ***slapped*** a worldwide ban on [MISC British] beef..."

Supervised learning in NLP

- Training examples:
  - "...a damaging row between [LOC Britain] and the [ORG EU] , which ***slapped*** a worldwide ban on [MISC British] beef..."
  - "In [LOC Detroit], [PER Brad Ausmus] 's three-run homer capped a four-run eighth and ***lifted*** the [ORG Tigers]..."

Supervised learning in NLP

- Training examples:
  - "...a damaging row between [LOC Britain] and the [ORG EU] , which ***slapped*** a worldwide ban on [MISC British] beef..."
  - "In [LOC Detroit], [PER Brad Ausmus] 's three-run homer capped a four-run eighth and ***lifted*** the [ORG Tigers]..."
- What we learn?

Supervised learning in NLP

- Training examples:
  - "...a damaging row between [LOC Britain] and the [ORG EU] , which ***slapped*** a worldwide ban on [MISC British] beef..."
  - "In [LOC Detroit], [PER Brad Ausmus] 's three-run homer capped a four-run eighth and ***lifted*** the [ORG Tigers]..."
- What we learn?
- Inference:
  - ... missed his club's last two games after FIFA slapped a …
  - ... lifted by FIFA on Friday...

Data Sparsity in NLP

- Training examples:
  - "...a damaging row between [LOC Britain] and the [ORG EU] , which ***slapped*** a worldwide ban on [MISC British] beef..."

Data Sparsity in NLP

- Training examples:
  - "...a damaging row between [LOC Britain] and the [ORG EU] , which ***slapped*** a worldwide ban on [MISC British] beef..."
- Data sparsity:
  - Devised
  - Annuled
  - Reimposed
  - Penned
  - ...
  - Issued
  - Authorised
  - Commissioned
  - Drafted
  - ...

Data Sparsity in NLP

- Training examples:
  - "...a damaging row between [LOC Britain] and the [ORG EU] , which ***slapped*** a worldwide ban on [MISC British] beef..."
- Data sparsity:
  - Devised
  - Annuled
  - Reimposed
  - Penned
  - ...
  - Issued
  - Authorised
  - Commissioned
  - Drafted
  - ...
- It is very likely that we will not see all of these words at training time. But we have a lot of unlabeled text- there has to be a way to know they are similar in some sense.

Outline of this talk

- Contributions.
- Induction of word representations from unlabeled text.
  - Preliminaries.
  - HMM-based representations.
  - NN-based representations.
- Using the word representations in NER
- Results & Conclusions.

Contributions

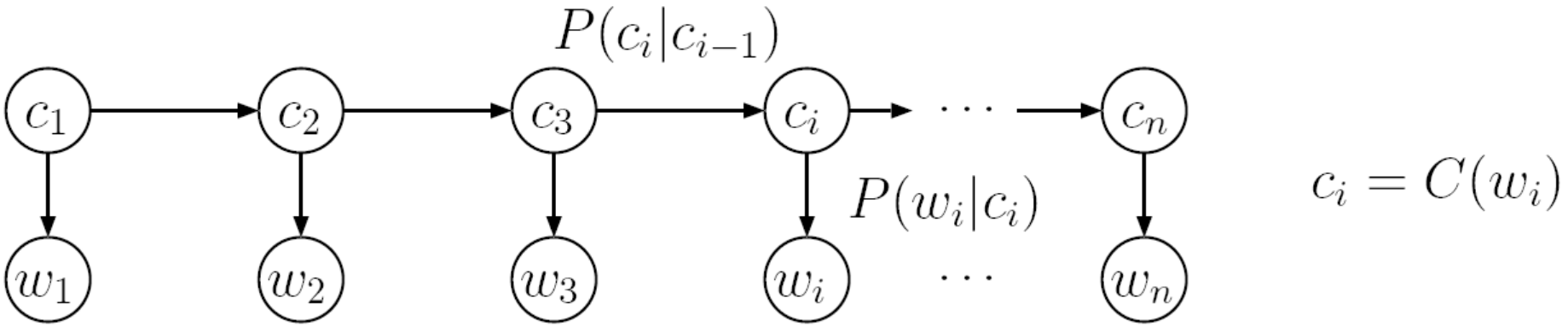
	System	Resources Used	F_1
+	LBJ-NER	Wikipedia, Nonlocal Features, Word-class Model	90.80
-	(Suzuki and Isozaki, 2008)	Semi-supervised on 1G-word unlabeled data	89.92
-	(Ando and Zhang, 2005)	Semi-supervised on 27M-word unlabeled data	89.31
-	(Kazama and Torisawa, 2007a)	Wikipedia	88.02
-	(Krishnan and Manning, 2006)	Non-local Features	87.24
-	(Kazama and Torisawa, 2007b)	Non-local Features	87.17
+	(Finkel et al., 2005)	Non-local Features	86.86

http://l2r.cs.uiuc.edu/~cogcomp/LbjNer.php

Outline of this talk

- Contributions.
- Induction of word representations from unlabeled text.
  - Preliminaries.
  - HMM-based representations.
  - NN-based representations.
- Using the word representations in NER
- Results & Conclusions.

Introduction to HMM

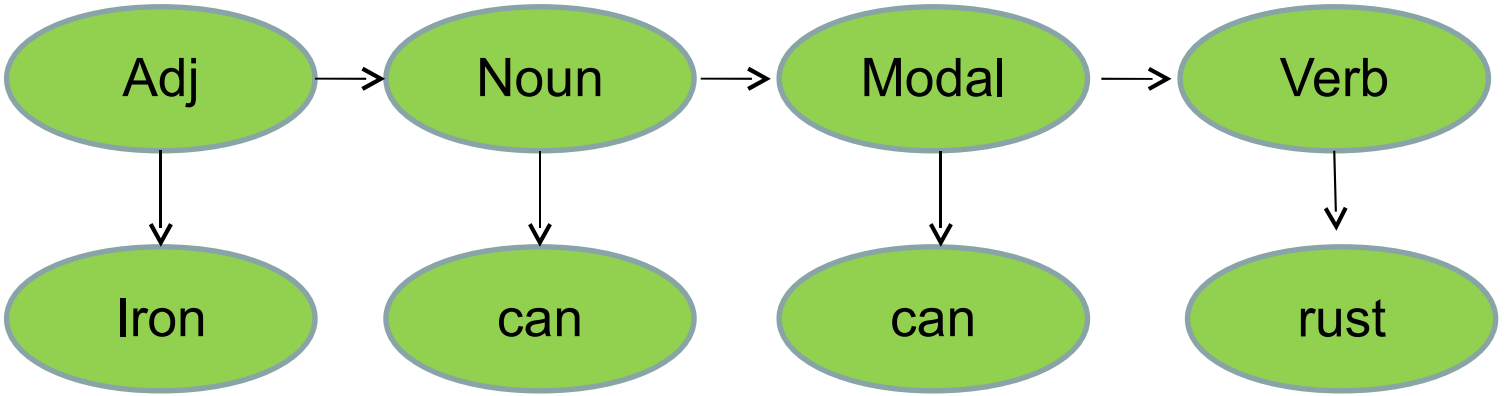


- Type of Bayesian Network.
  - Well researched and understood.
  - Words are generated from hidden states.
  - Parametrized by “emission” and “transition” probabilities.
  - Joint/marginal probabilities calculated efficiently with dynamic programming.

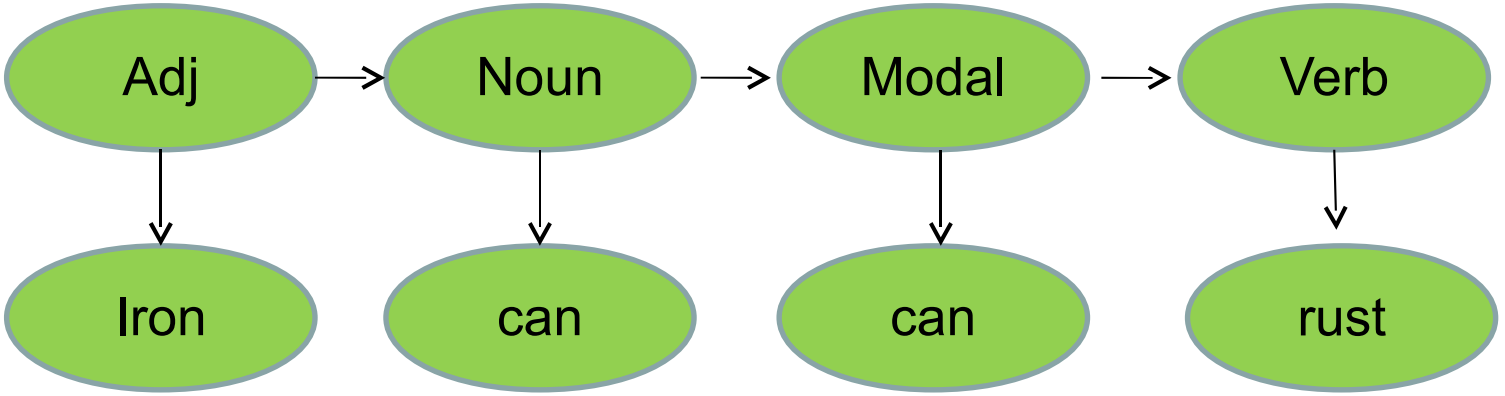
Understanding HMMs

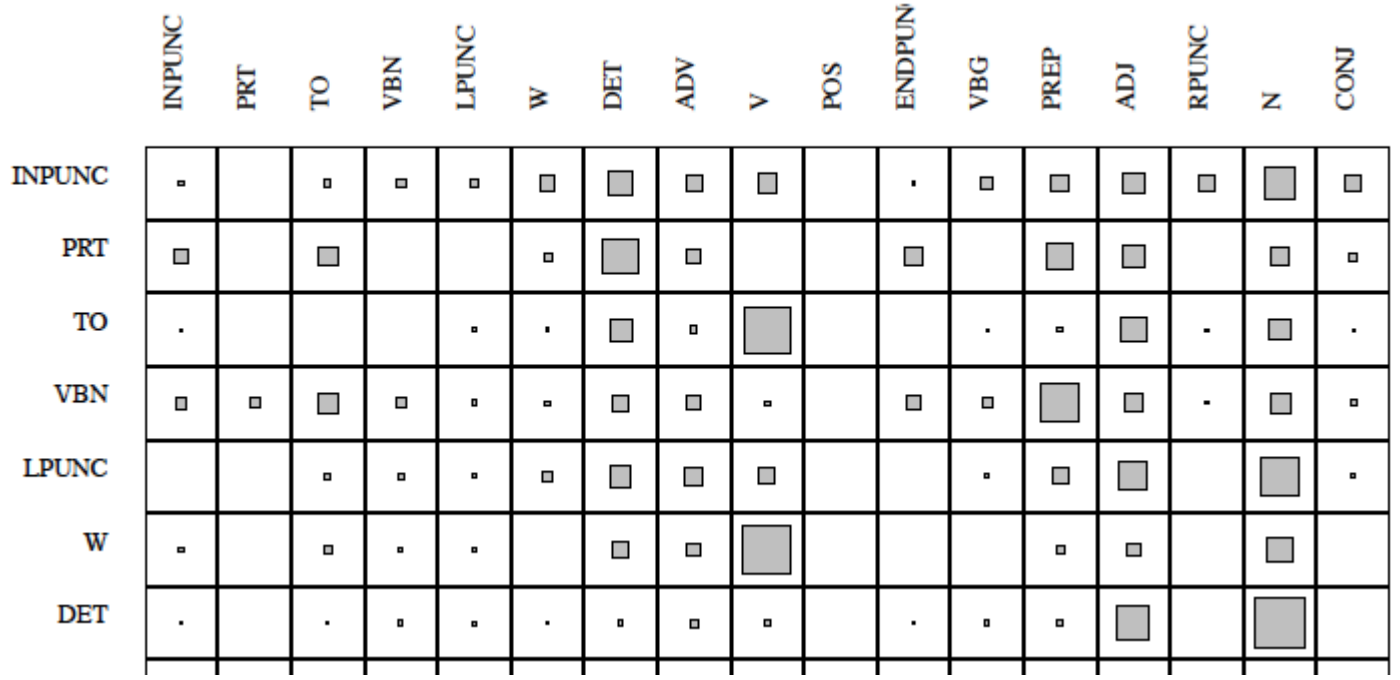
- HMMs are very popular in NLP
- 3 examples of modeling NLP with HMMS
  - POS tagging.
  - Extracting fields from citations.
  - NER.

POS Tagging with HMM.

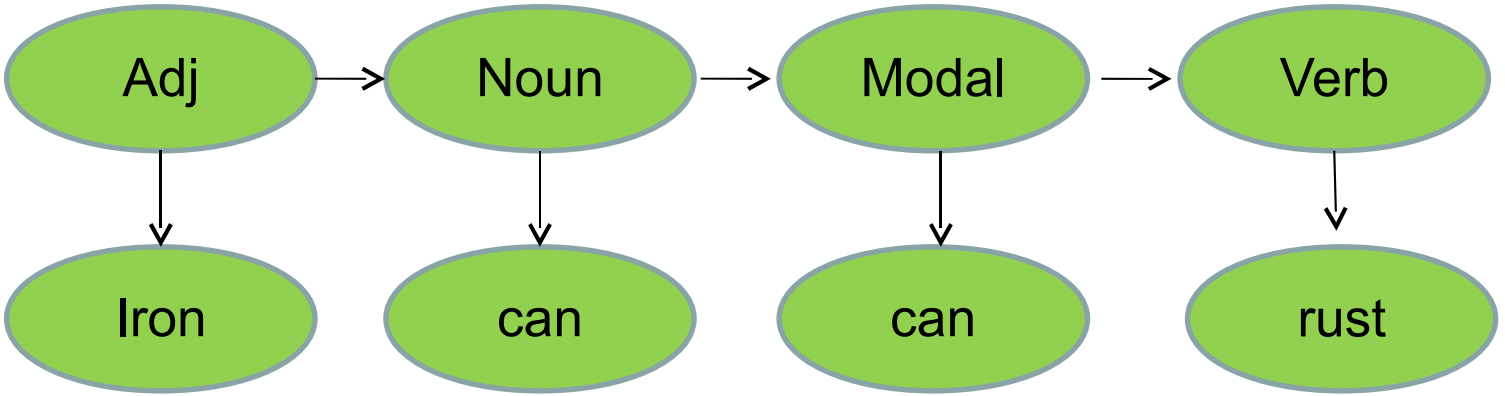


POS Tagging with HMM.





POS Tagging with HMM.



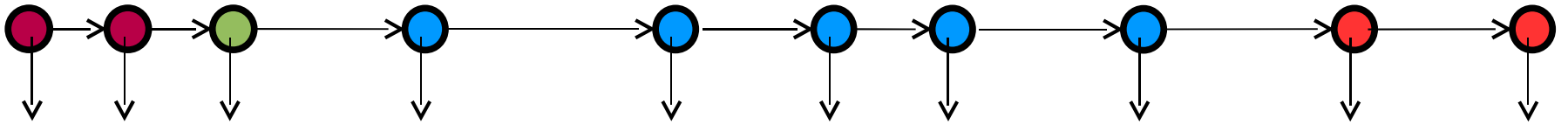
	Adjective	Noun	Modal	Verb
Iron	0.0002	0.095	0	0.0001
can	0	0.005	0.075	0
rust	0	0.004	0	0.006

$$\Sigma = 1$$

Example – Field Extraction

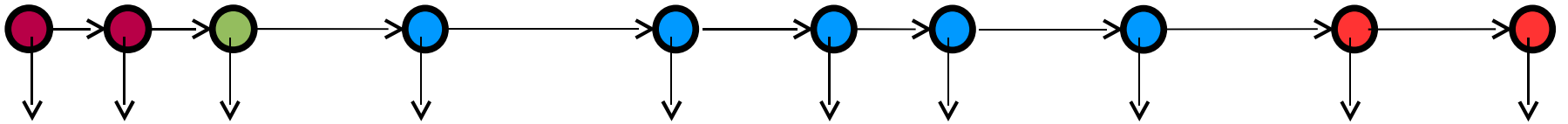
Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. Morgan Kaufman.

Example – Field Extraction

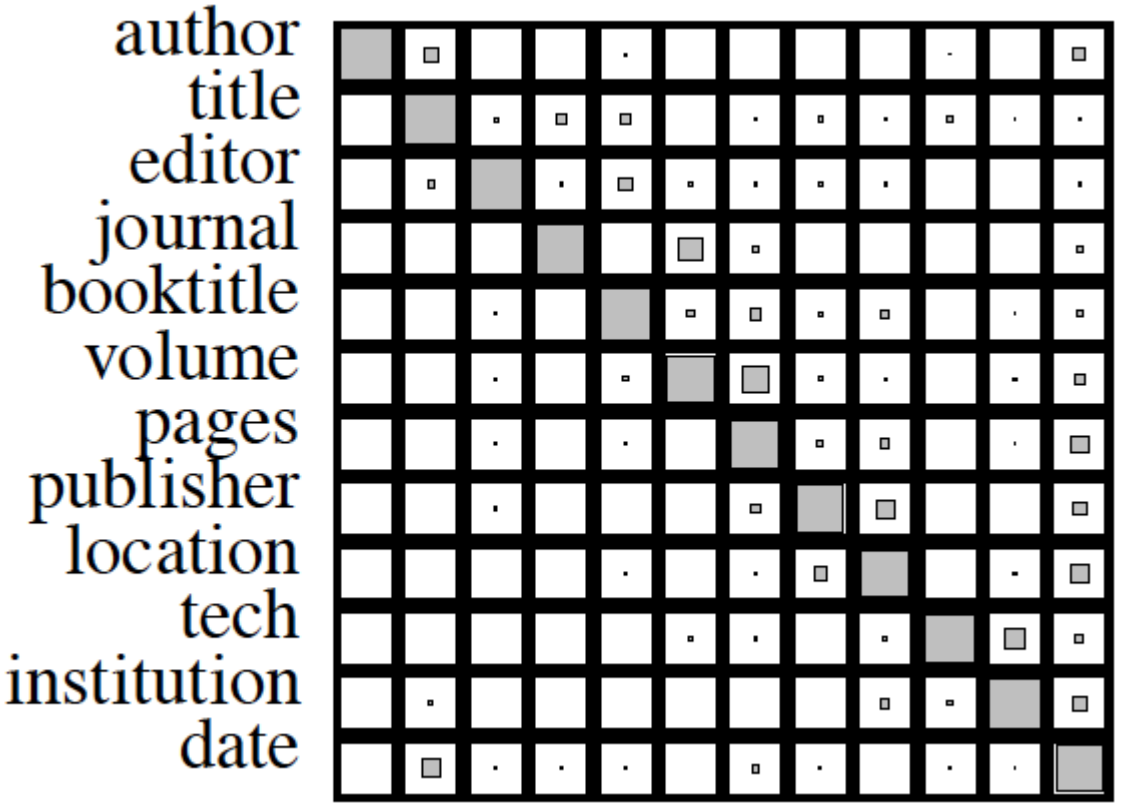


Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. Morgan Kaufman.

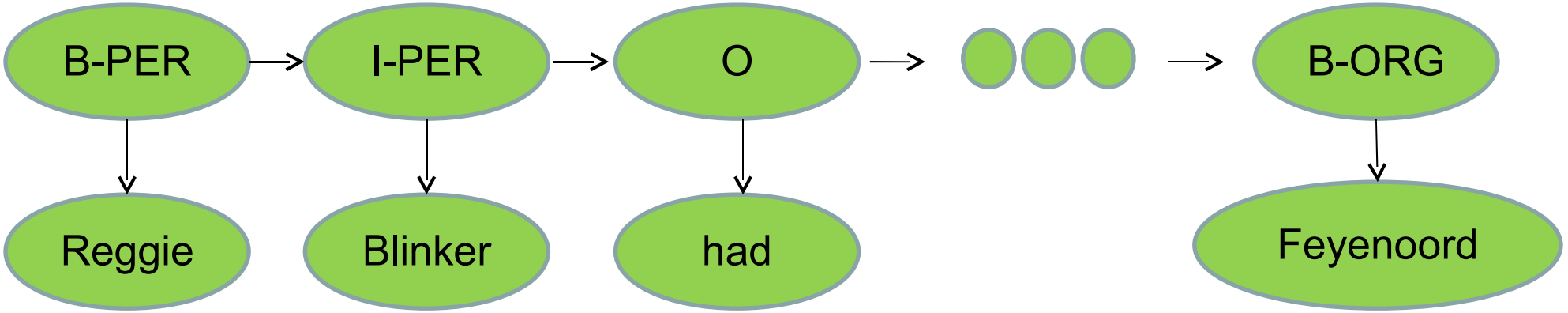
Example – Field Extraction



Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. Morgan Kaufman.



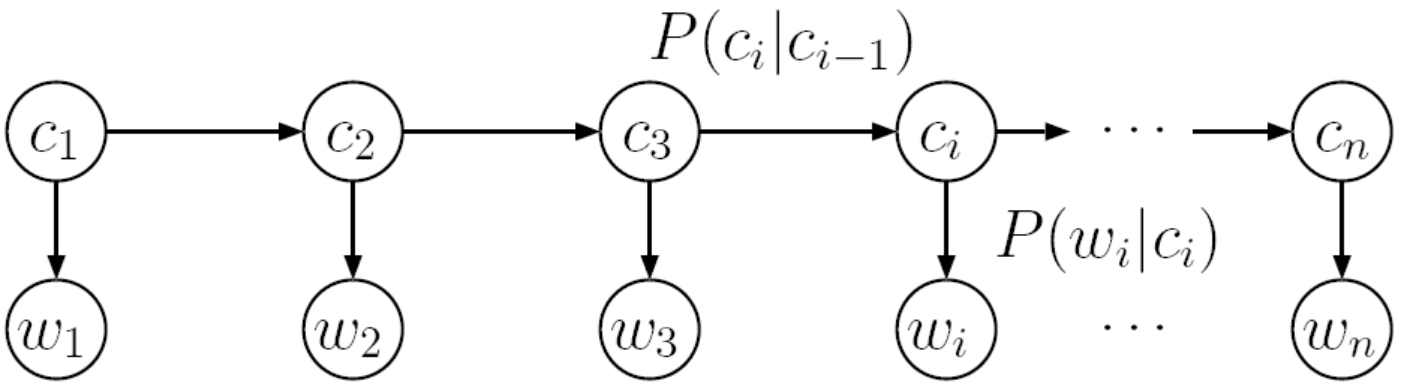
Example – NER (*)



- How to encode the additional features?

- We have chunks of PER/LOC/ORG separated by O chunks. This “breaks” the HMM to a bunch of independent optimizations

HMM-recap

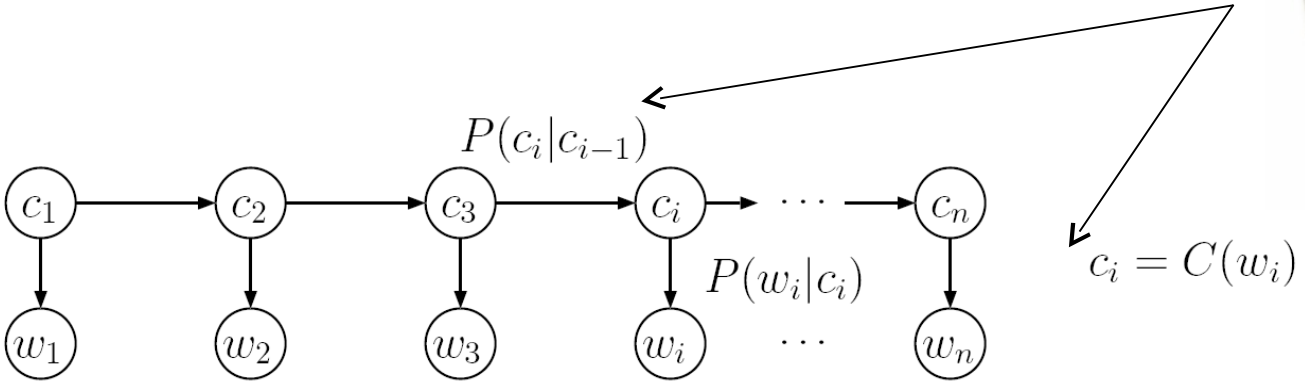


$$c_i = C(w_i)$$

HMM properties:
- Dynamic programming for inference and training.
- We can even efficiently learn the model in the absence of training data! We “find” the emission and the transition tables that maximize the of the observed data (EM). (For any task???)

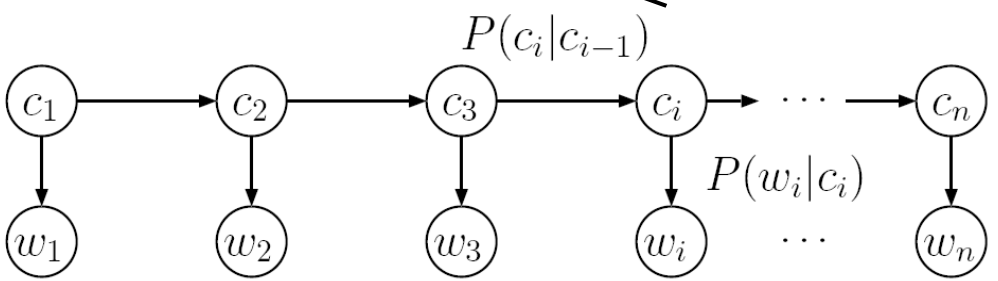
Fitting HMM with unlabeled data

There exists an algorithm (EM) that finds the parameters which maximize the likelihood of the data.
We prepare an HMM with prescribed number of hidden states trained on a large collection of unlabeled data.

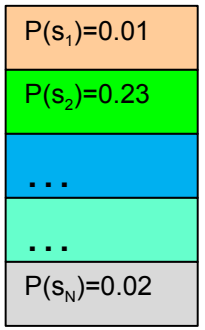


Fitting HMM with unlabeled data

When training or testing, we can use the Baum-Welch algorithm to get the probability distribution over the states.
This says that w_3 belongs to state s_2 with probability 0.23 in the given context .



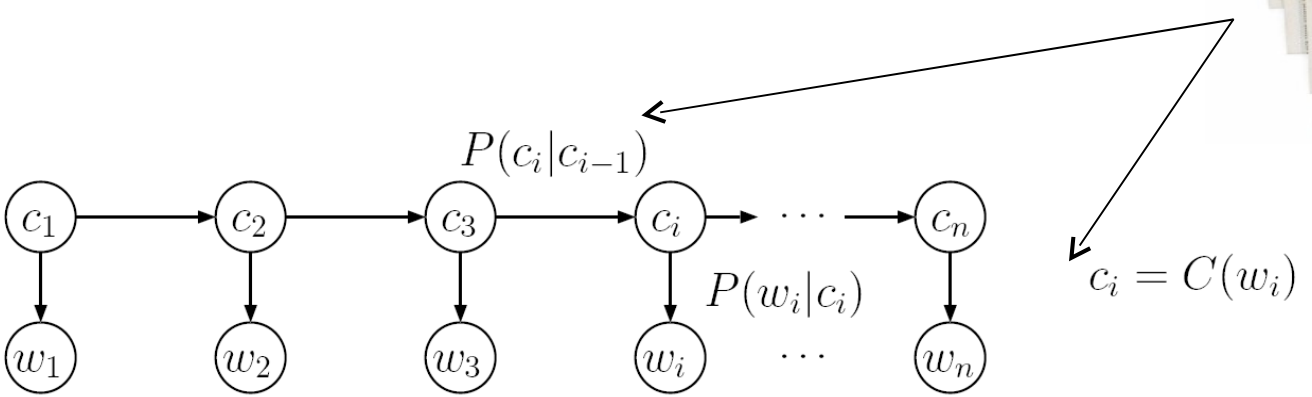
$c_i = C(w_i)$

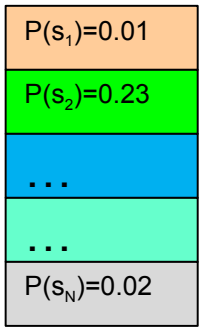


Fitting HMM with unlabeled data

Assuming that the hidden states correspond to some "semantic properties" of the words, we move to a dense representation and avoid data sparsity.

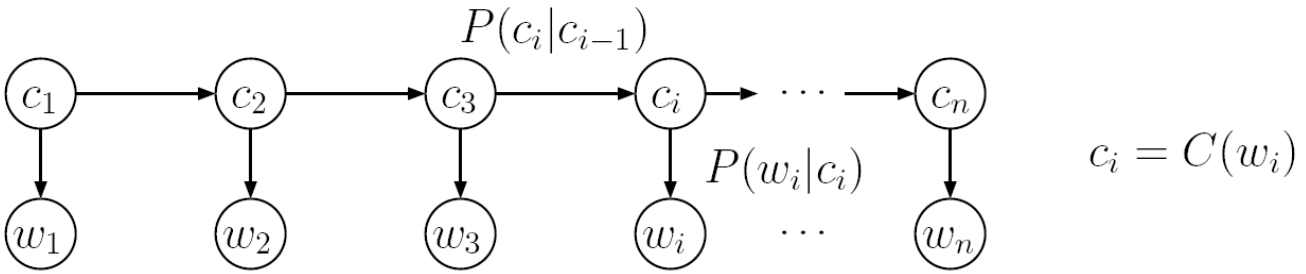
Huang&Yates [ACL09] show improvement in a variety of NLP tasks using this method.





Scales up to 20-100 hidden states

Word Class Models



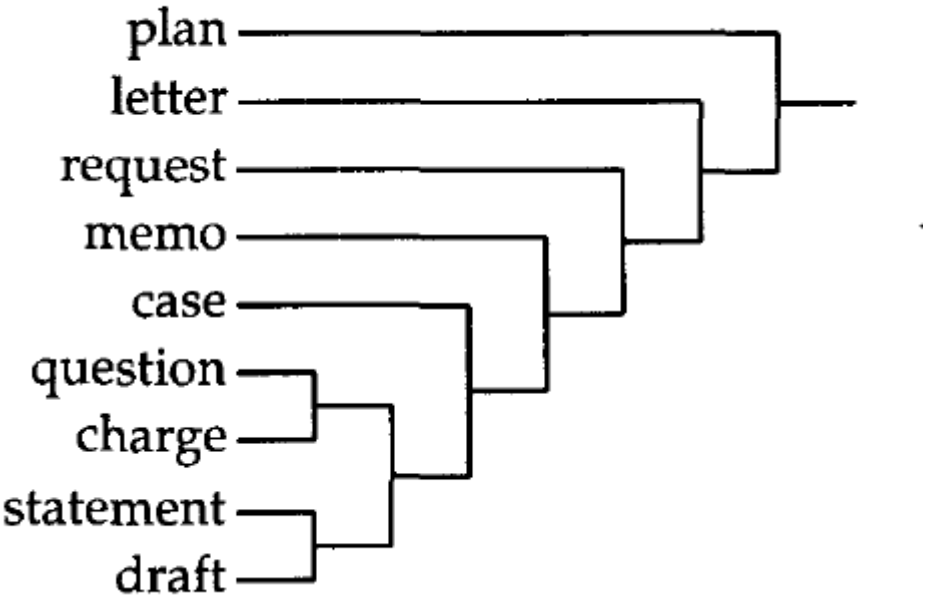
[Brown et. al. 1992] analyze the model constraint where each word can be generated by a single state. While their model is essentially an HMM, they develop a different training algorithm.

Advantages:

- Sparser, smaller models.
- When applying to new texts, no Baum-Welch inference is necessary. Each word can be deterministically be mapped to its class.

Word Class Models

By repeatedly inducing a word class model over the hidden state, they generate a hierarchical clustering of words

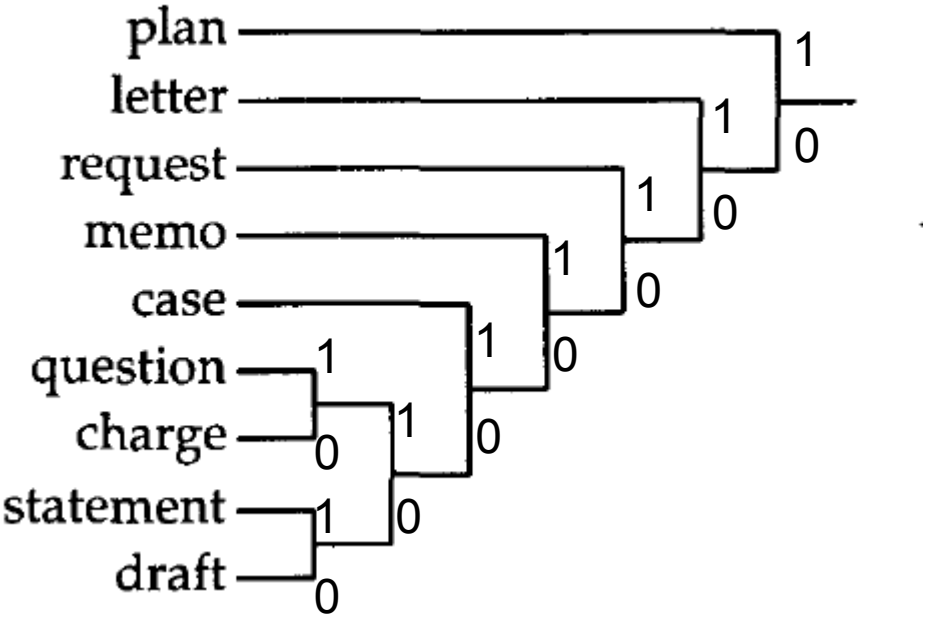


Using Word Class Models

Now we can assign each word a binary representation:

Question : 0000011

Statement: 0000000



Sample Clusters

Remember that we use prefixes of different length for different abstraction levels!

- *111111110110000* *slapped*
- 111111110110000 shattered
- 111111110110000 commissioned
- 111111110110000 drafted
- 111111110110000 authorized
- 111111110110000 authorised
- 111111110110000 imposed
- 111111110110000 established
- 111111110110000 developed

- *111111111100110* *officer*
- 111111111100110 acquaintance
- 111111111100110 policymaker
- 111111111100110 instructor
- 111111111100110 investigator
- 111111111100110 advisor
- 111111111100110 aide
- 111111111100110 expert
- 111111111100110 adviser

Sample Clusters

Remember that we use prefixes of different length for different abstraction levels!

- *101111000001* *bill*
- 101111000001 waiver
- 101111000001 protocol
- 101111000001 prospectus
- 101111000001 clause
- 101111000001 directive
- 101111000001 decree
- 101111000001 declaration
- 101111000001 document
- 101111000001 resolution
- 101111000001 proposal

- *111111100* *Bill*
- 111111100 Boris
- 111111100 Warren
- 111111100 Fidel
- 111111100 Yasser
- 111111100 Kenneth
- 111111100 Viktor
- 111111100 Benjamin
- 111111100 Jacques
- 111111100 Bob
- 111111100 Alexander

Sample Clusters

Remember that we use prefixes of different length for different abstraction levels!

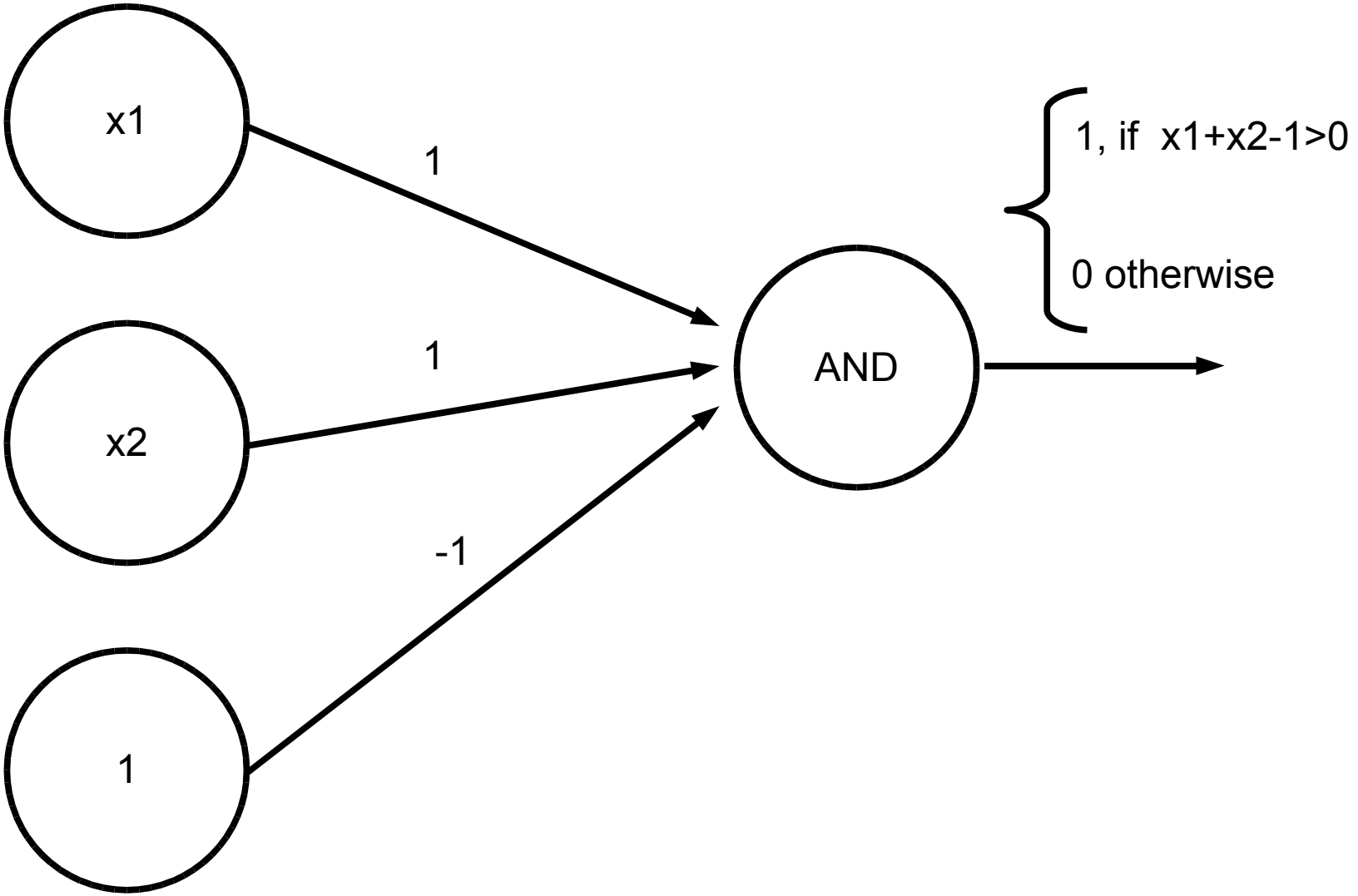
- *111110100 Clinton 15073*
- 111110100 Aleman 380
- 111110100 Zeroual 398
- 111110100 Sampras 424
- 111110100 Barzani 477
- 111110100 Cardoso 558
- 111110100 Kim 1257
- 111110100 King 1816
- 111110100 Saddam 2256
- 111110100 Netanyahu 5436
- 111110100 Dole 6106

- *111111100 Bill*
- 111111100 Boris
- 111111100 Warren
- 111111100 Fidel
- 111111100 Yasser
- 111111100 Kenneth
- 111111100 Viktor
- 111111100 Benjamin
- 111111100 Jacques
- 111111100 Bob
- 111111100 Alexander

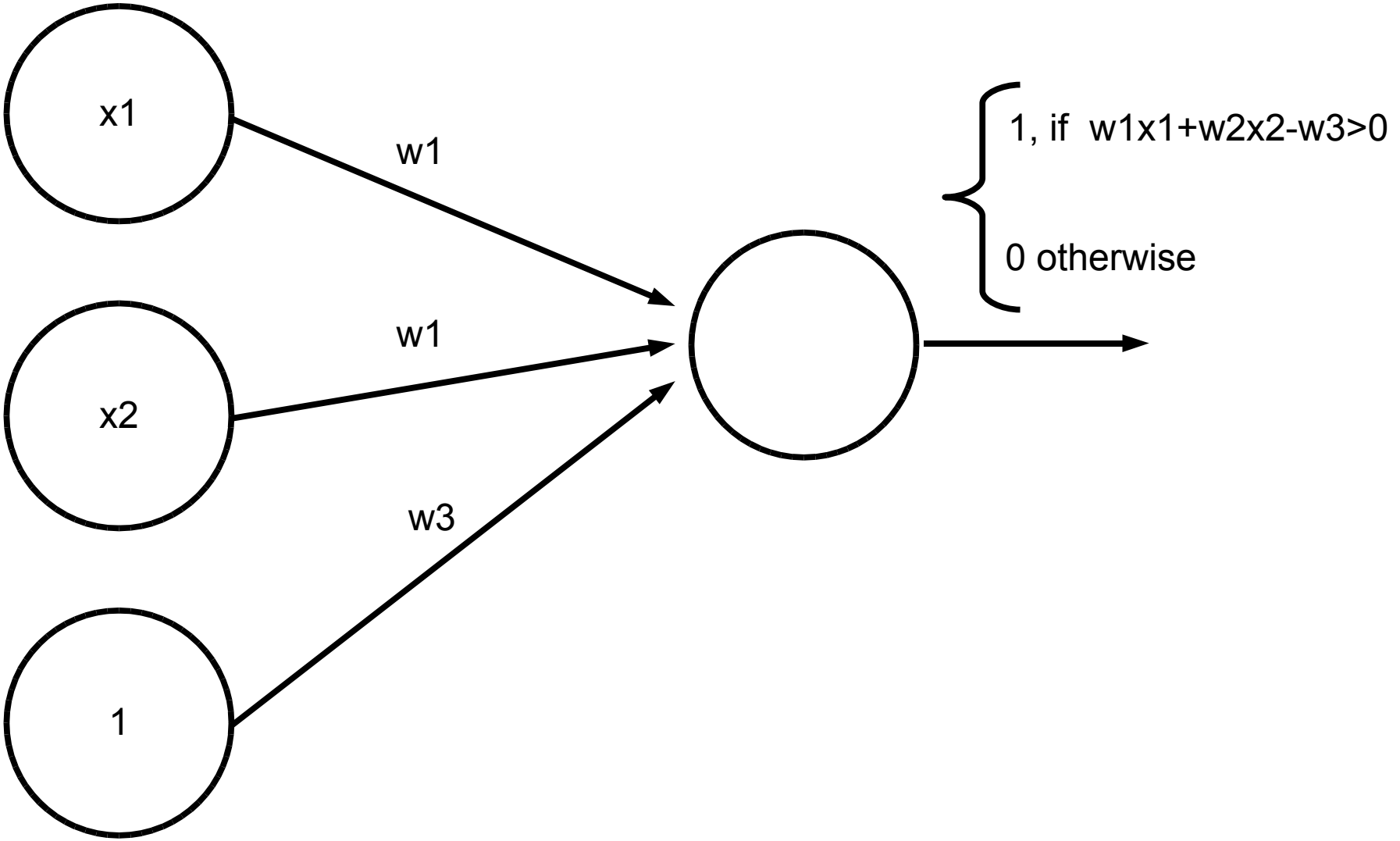
Outline of this talk

- Contributions.
- Induction of word representations from unlabeled text.
  - Preliminaries.
  - HMM-based representations.
  - NN-based representations.
- Using the word representations in NER
- Results & Conclusions.

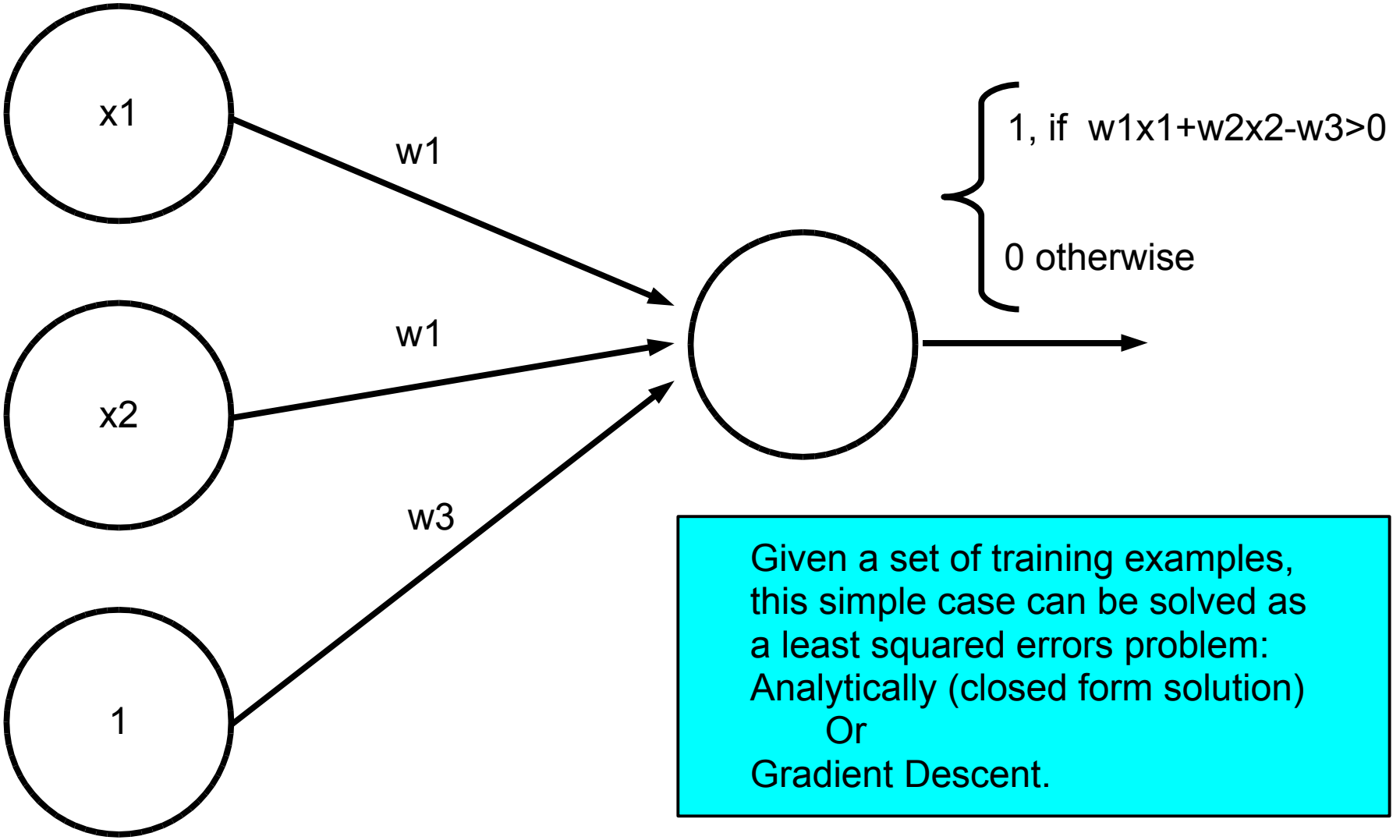
Neural Networks



Training Neural Networks

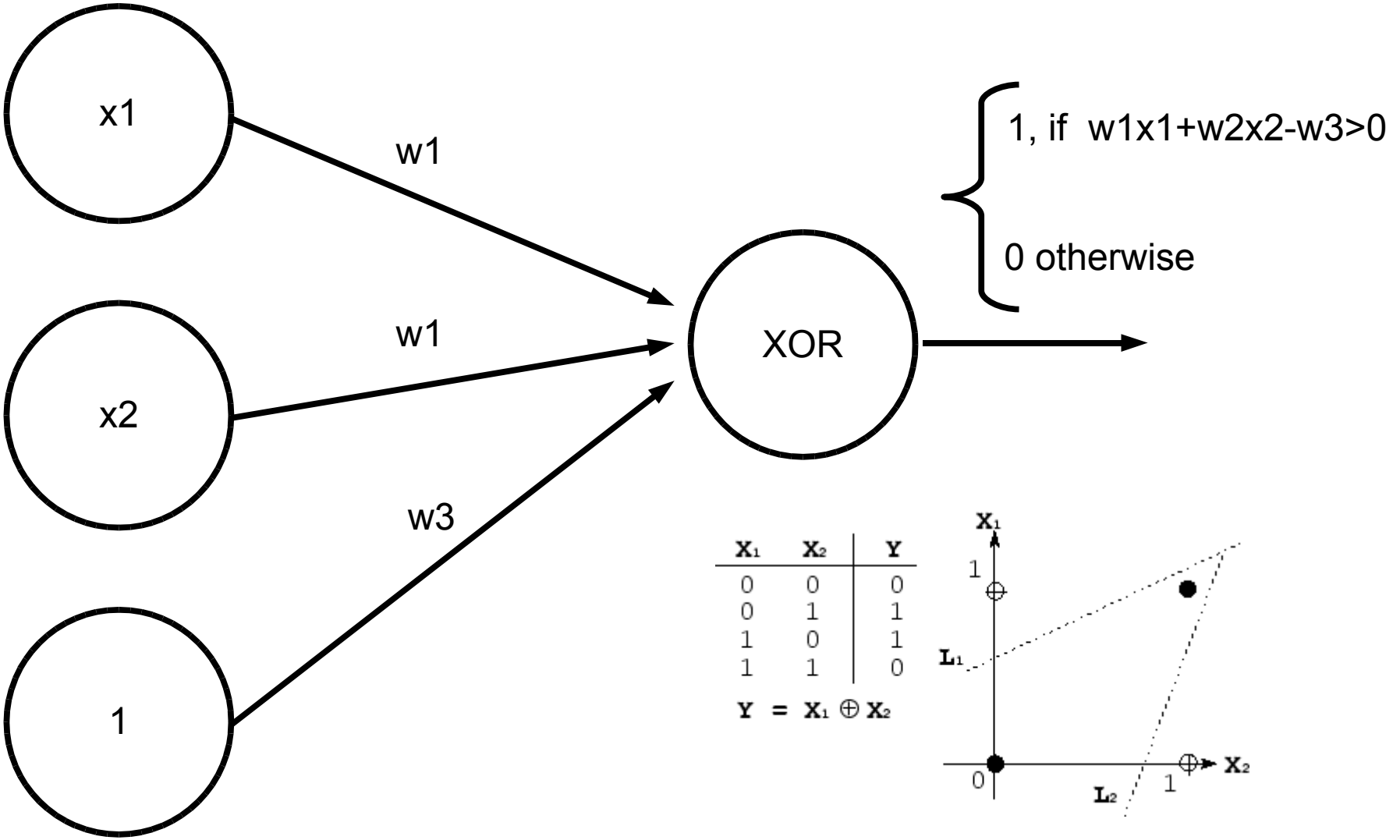


Training Neural Networks



Given a set of training examples,
this simple case can be solved as
a least squared errors problem:
Analytically (closed form solution)
Or
Gradient Descent.

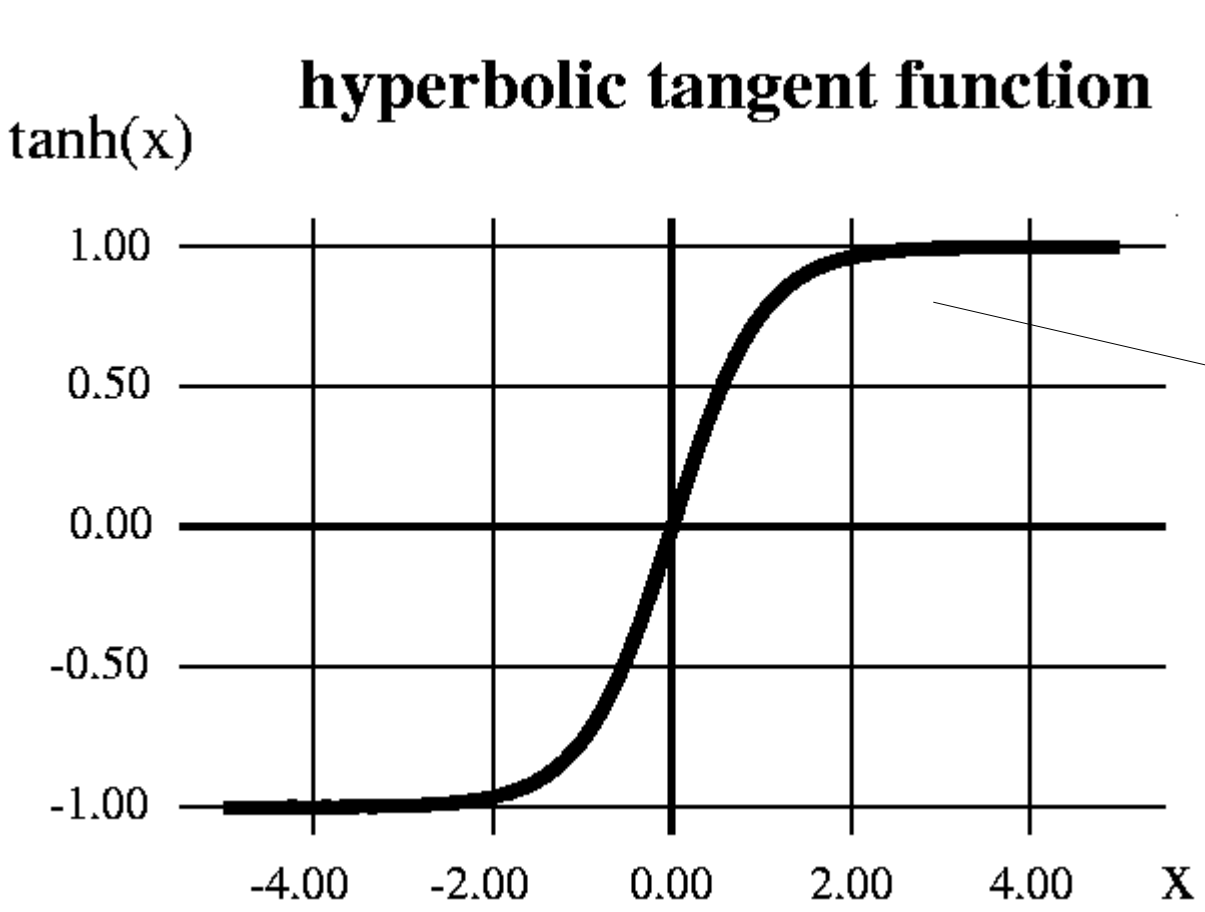
Neural Networks-expressivity

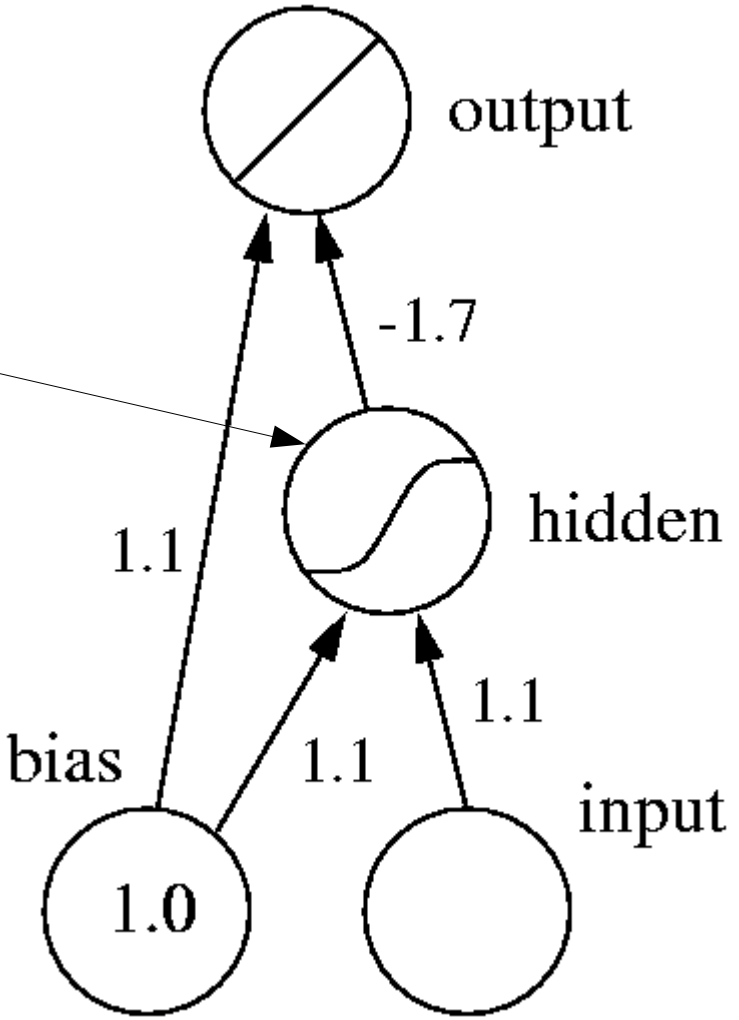


X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

$Y = X_1 \oplus X_2$

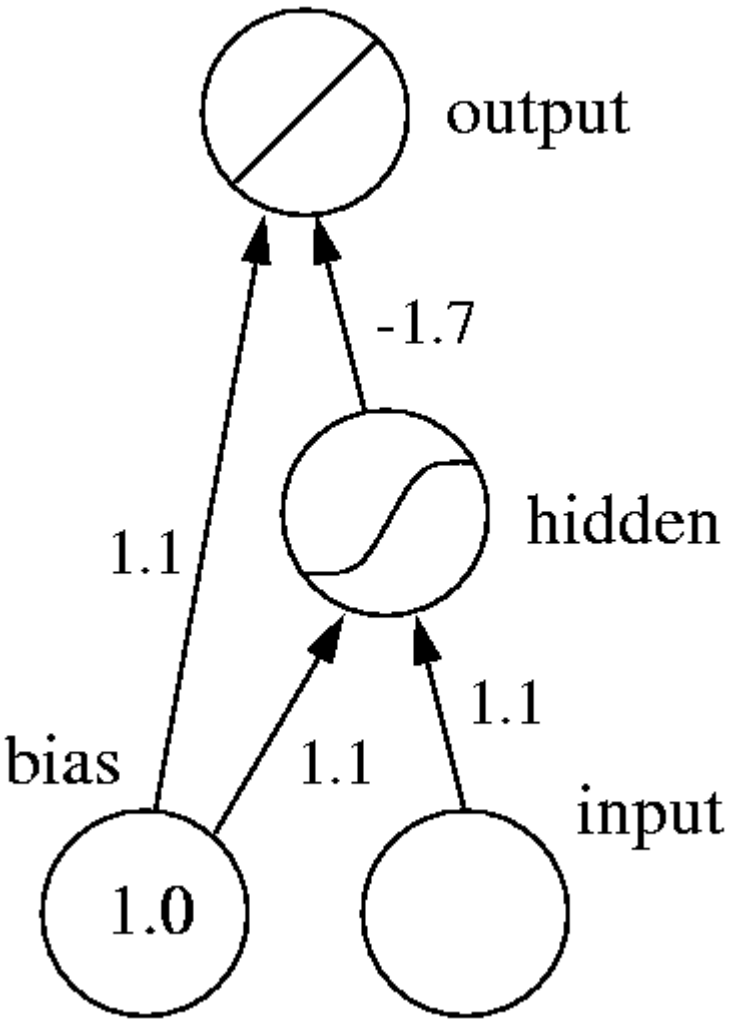
Multilayered NN





Multilayered NNs

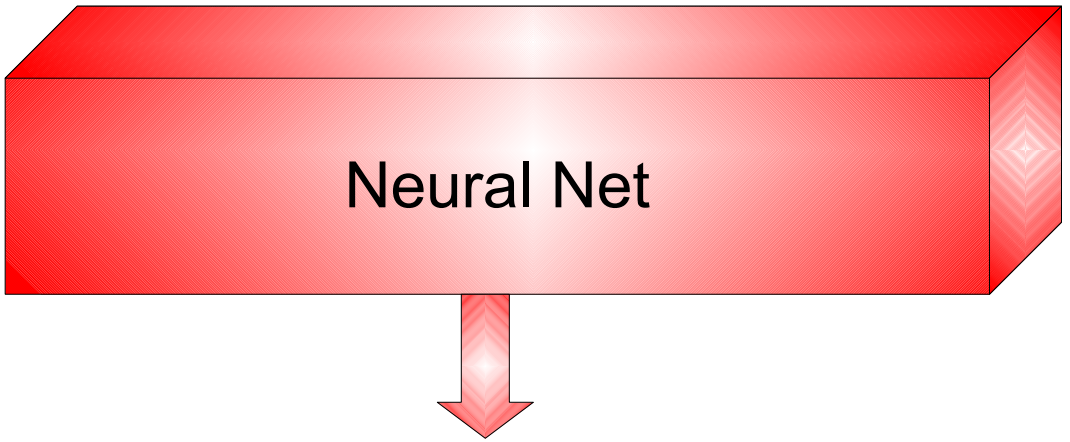
- Hidden layers must be nonlinear (else, no additional expressivity)
- Training, harder but possible- gradient descent technique known as backpropagation.



Modeling NLP With NNs

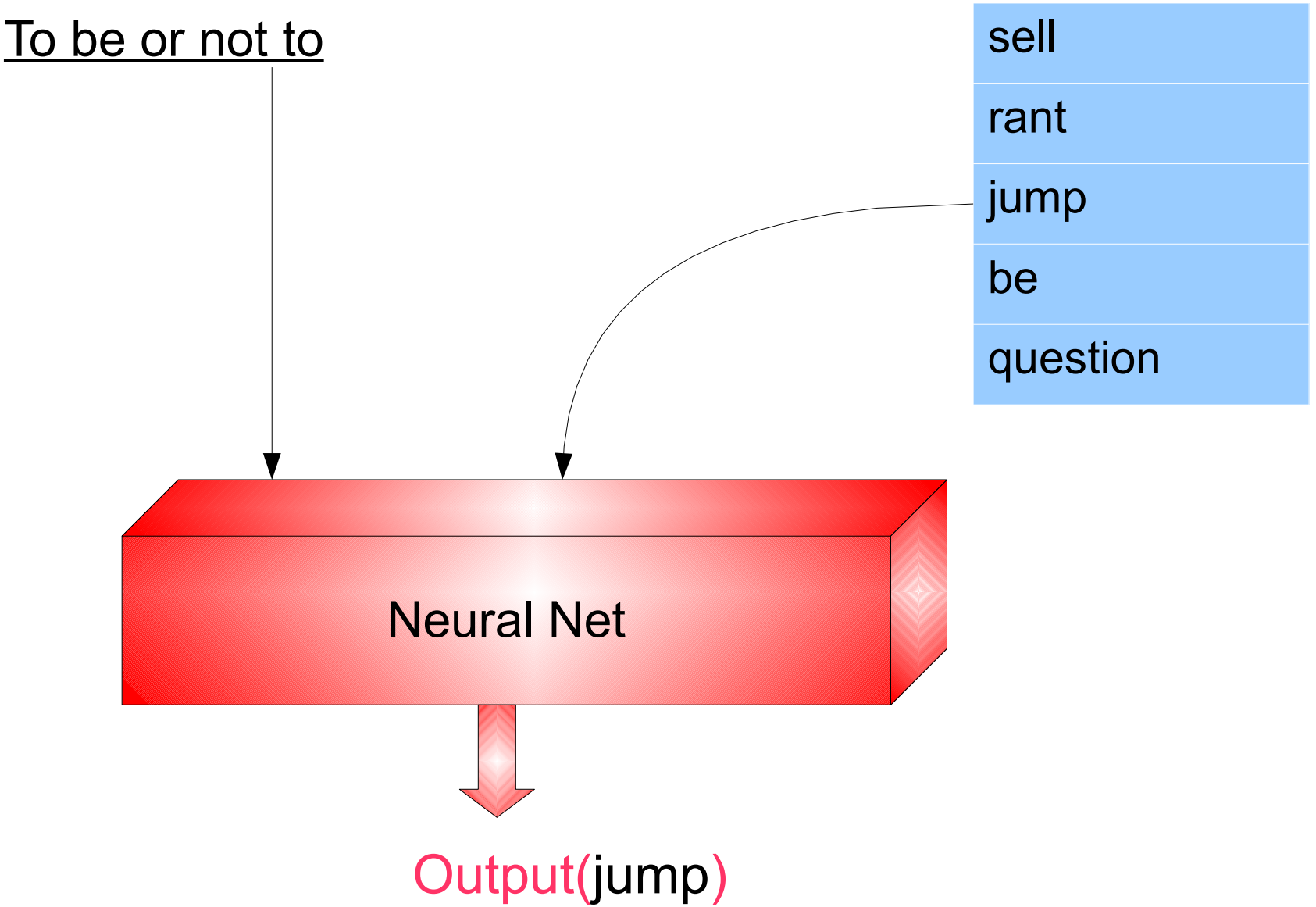
To be or not to

sell
rant
jump
be
question

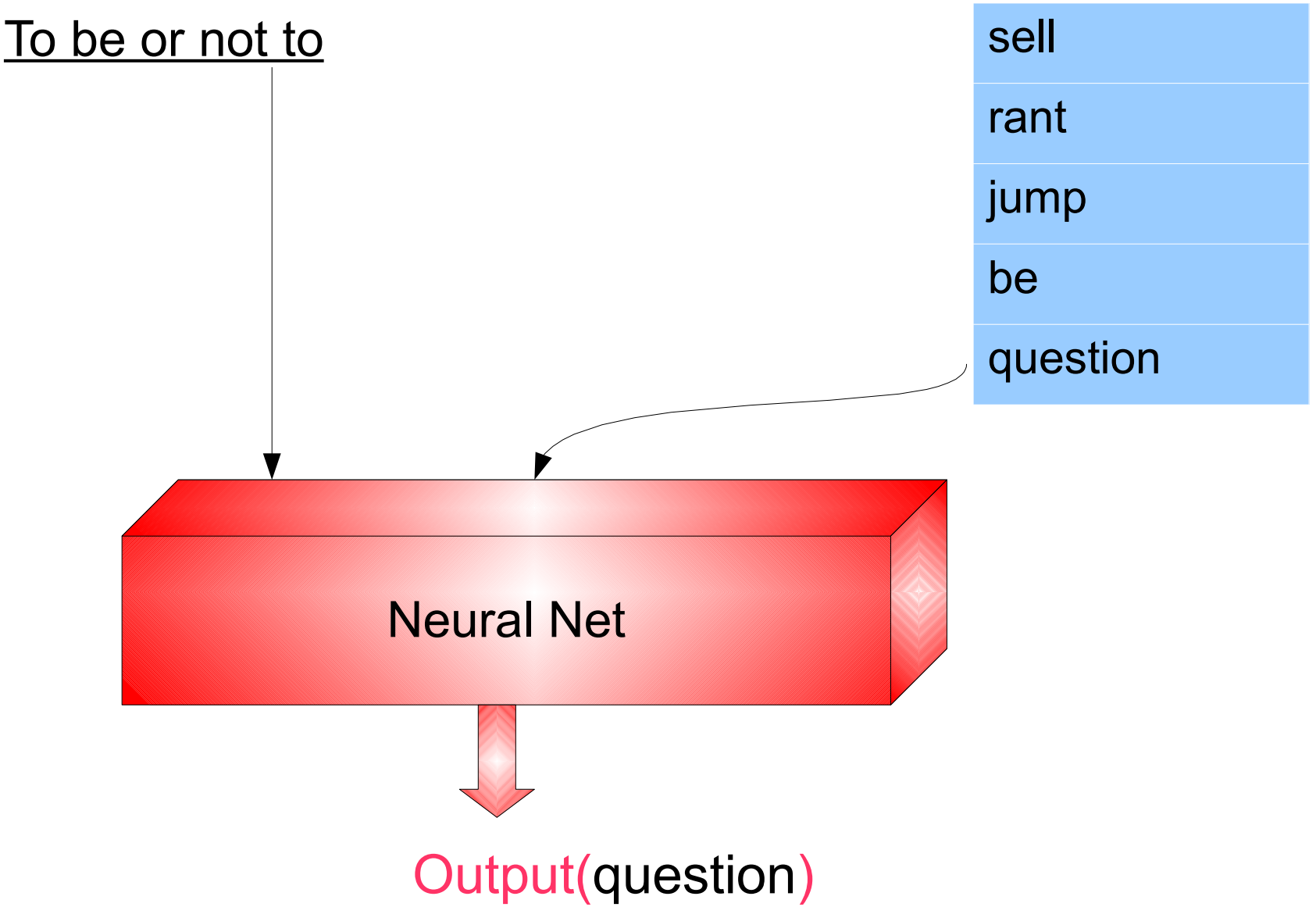


Output

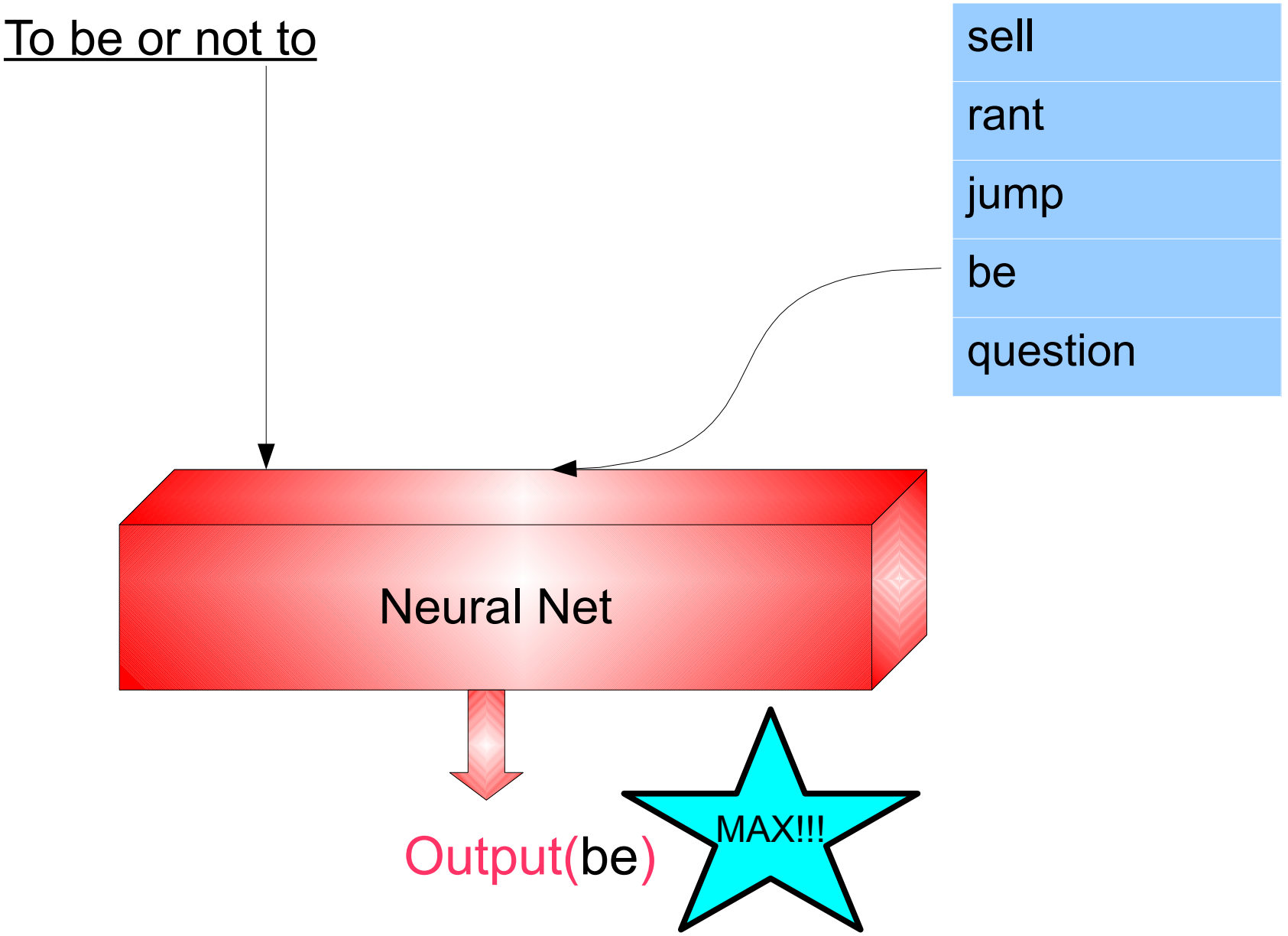
Modeling NLP With NNs



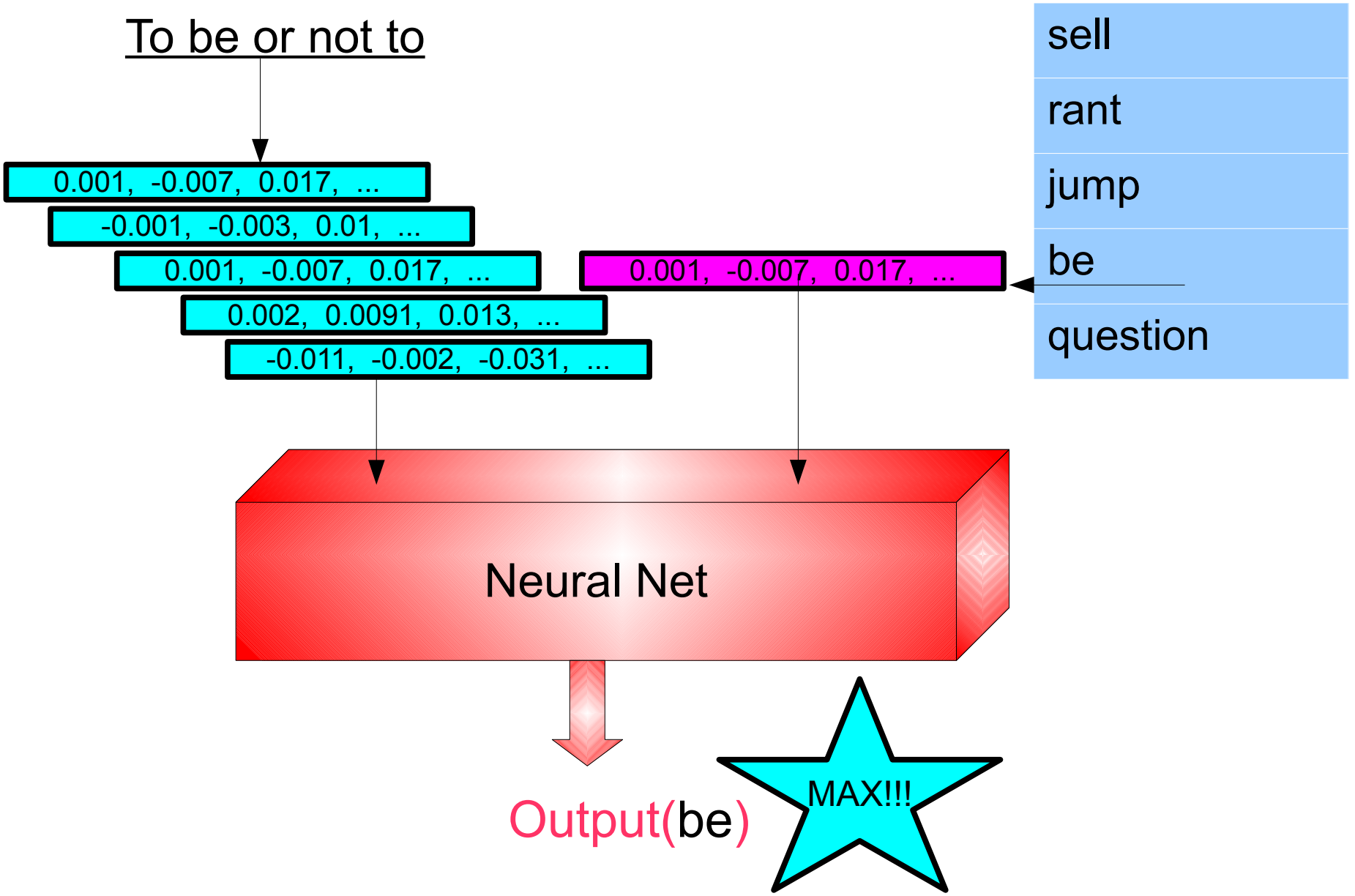
Modeling NLP With NNs



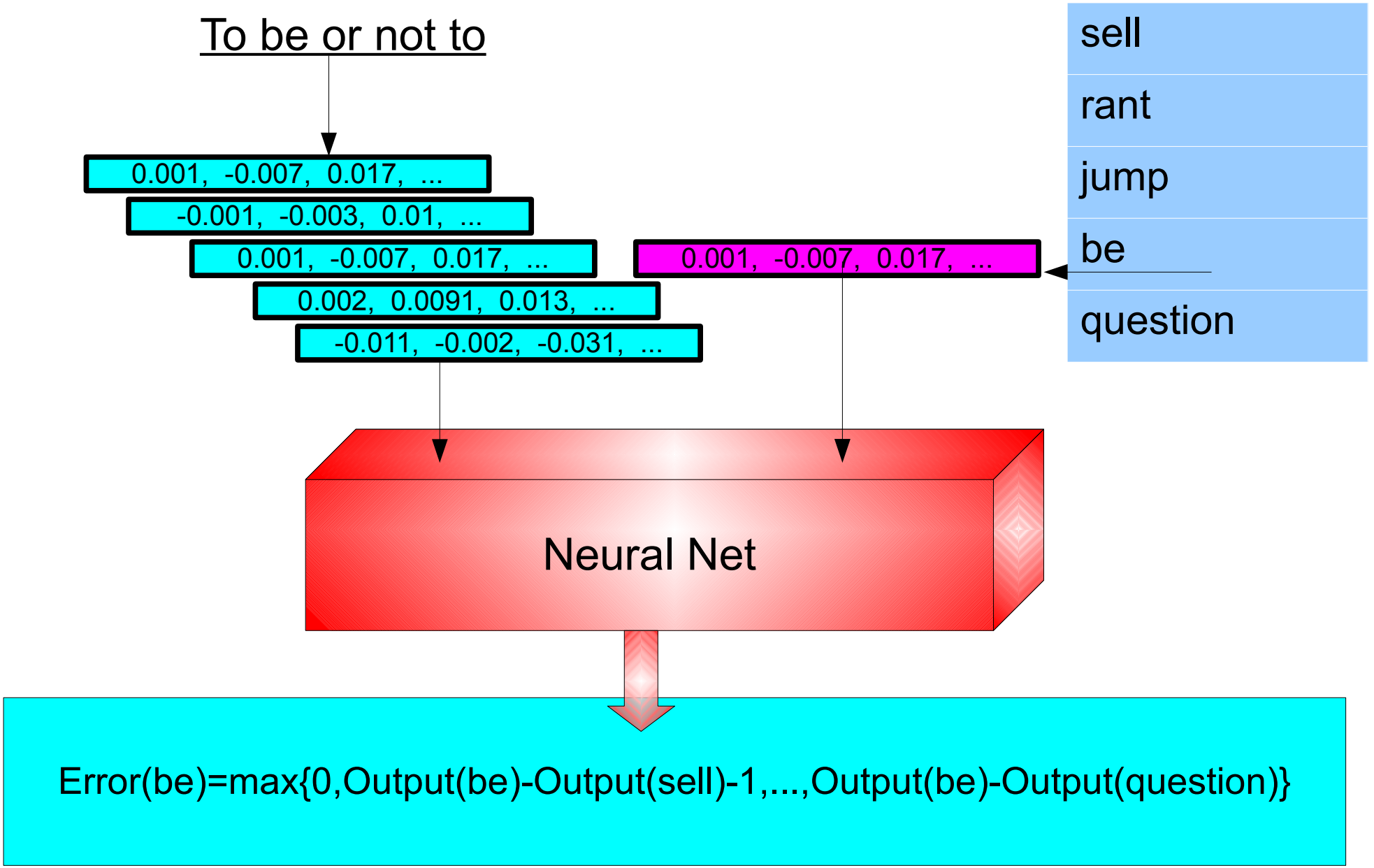
Modeling NLP With NNs



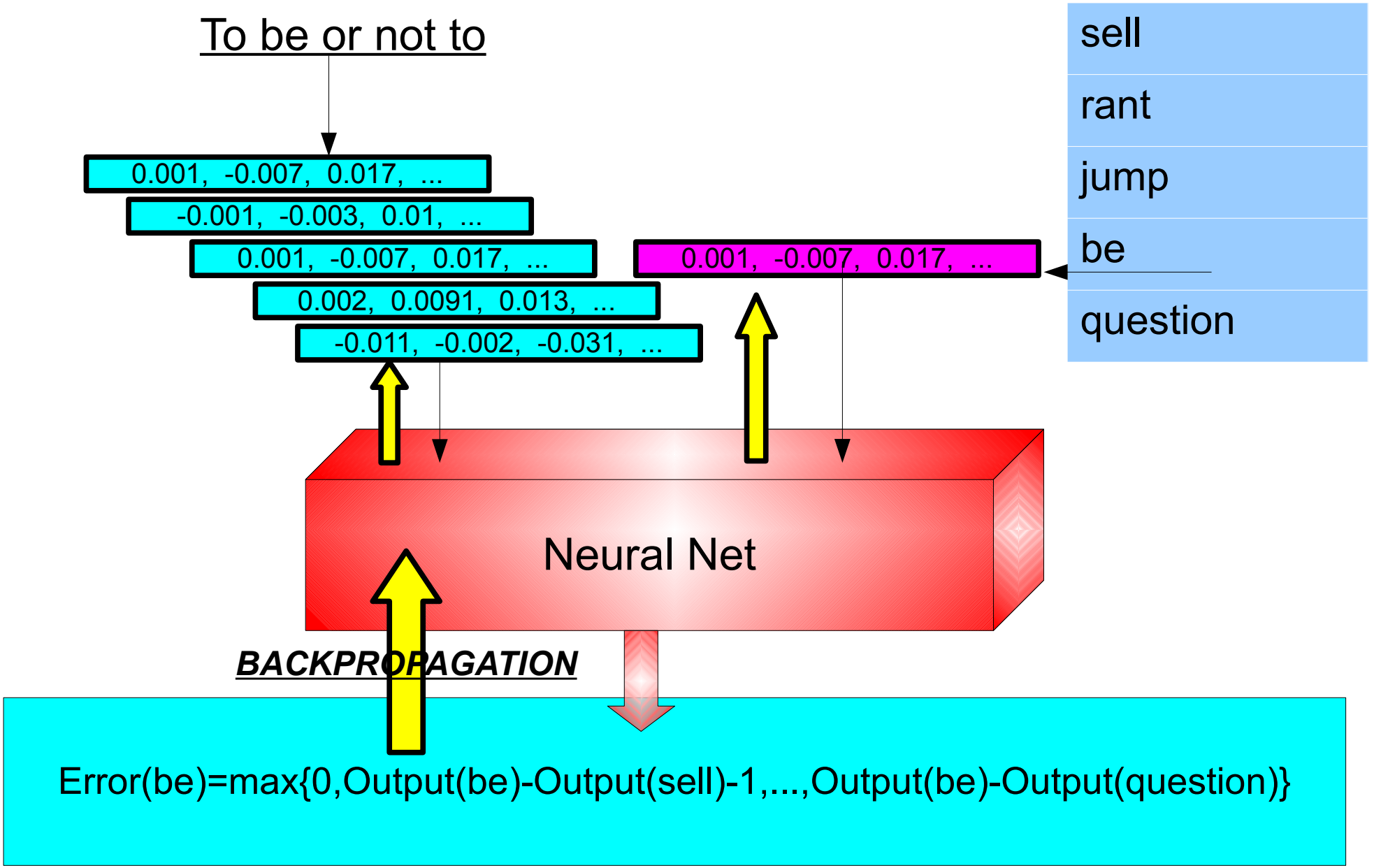
Modeling NLP With NNs



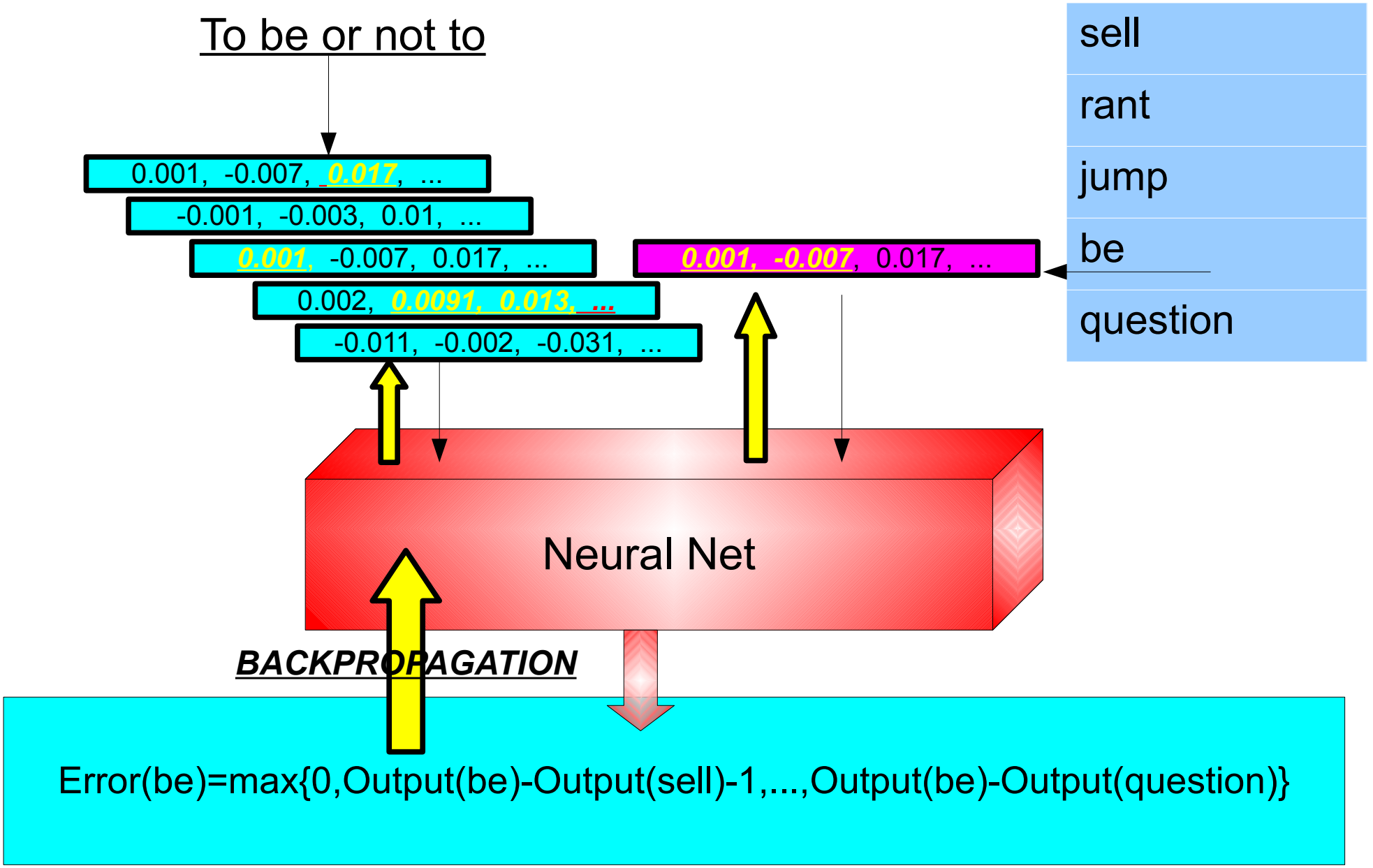
Modeling NLP With NNs



Modeling NLP With NNs



Modeling NLP With NNs



What do we have now?

- A NN, which given 4 tokens, can predict the 5-th by comparing all possibilities.
  - Very slow inference.
  - State of the art performance [Mnih&Hinton, 2009]
- More importantly- if the 50-dimensional vectors help to predict the next word, they carry useful information.
  - Use as additional features in my favorite HMM.
  - “Fast” lookup tables.

Summary Of NN embeddings

- We have implemented another approach for learning the embeddings (in a different model), HLBL [Mnih&Hinton2009]
  - 50 and 100 dimensions.
- The advantage of the embeddings is that all words are represented with 50/100-dimensional vectors.
  - Less data sparsity
  - We can say something about the word even if we have not seen it in training.

Summary So Far

- We have discussed 3 approaches to represent words:
  - Brown clusters
    - 01000111010 congregations
    - 01000111010 masterminds
    - 01000111010 blockers
    - 01000111010 columnists
    - 01000111010 molecules
    - 01000111010 journals
    - 01000111010 watchdogs

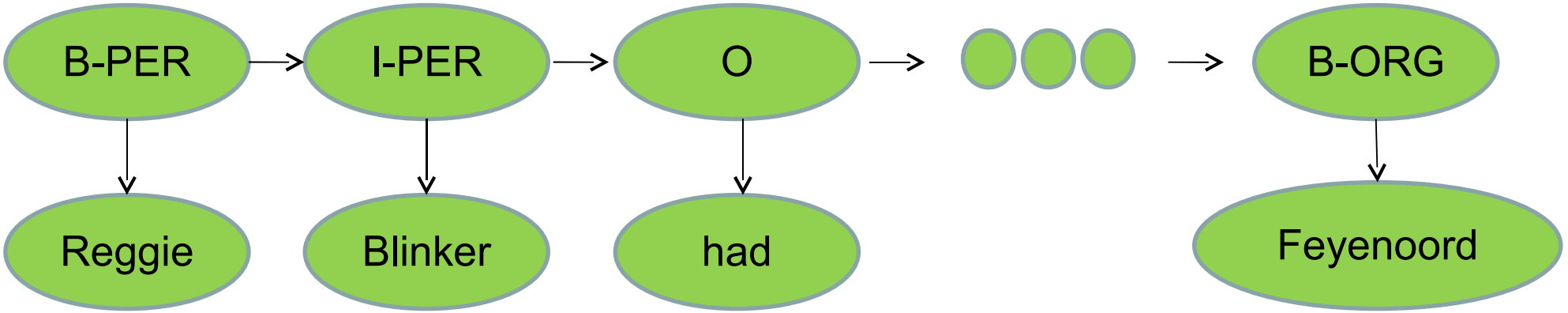
Summary So Far

- We have discussed 3 approaches to represent words:
  - Neural Networks: (C&W, HLBL)
    - require: (7.22e-03, -4.52-02, 6.83e-03, …)
    - Times: (-2.88e-01, 3.49e-01, -8.19e-02, …)
    - Office: (-1.58e-01, 5.52e-02, 9.89e-02, …)
    - ...

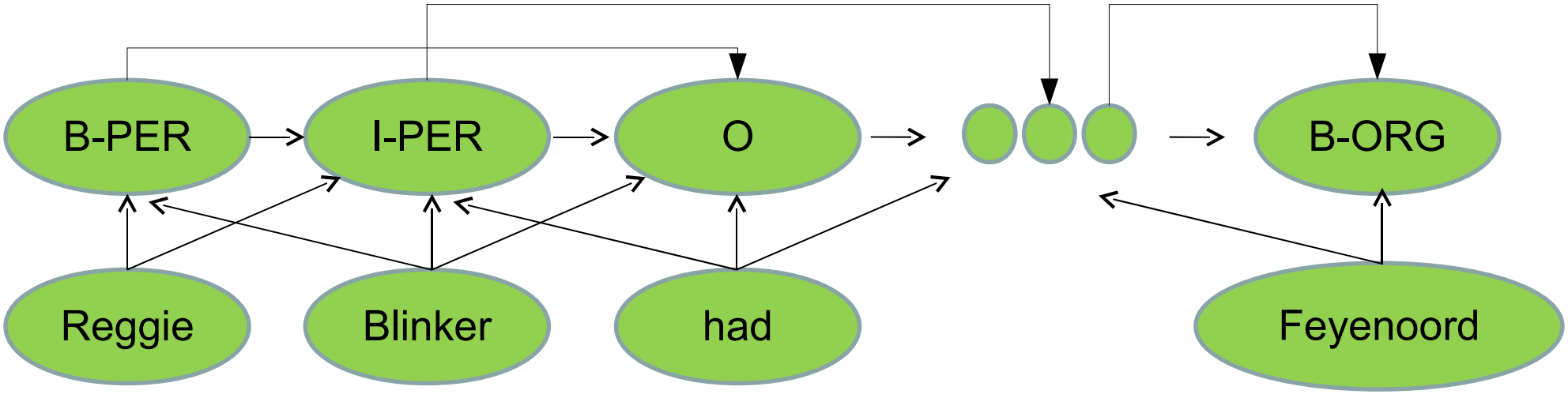
Outline of this talk

- Contributions.
- Induction of word representations from unlabeled text.
  - Preliminaries.
  - HMM-based representations.
  - NN-based representations.
- Using the word representations in NER
- Results & Conclusions.

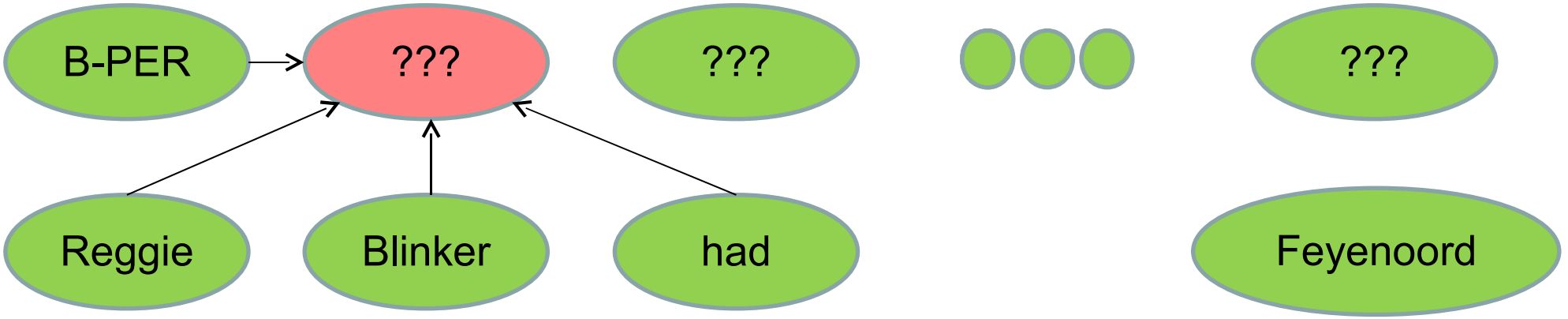
Modeling NER.



Modeling NER.

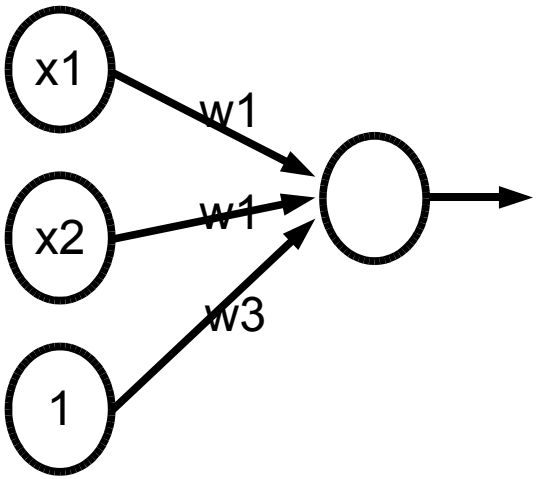


Modeling NER.

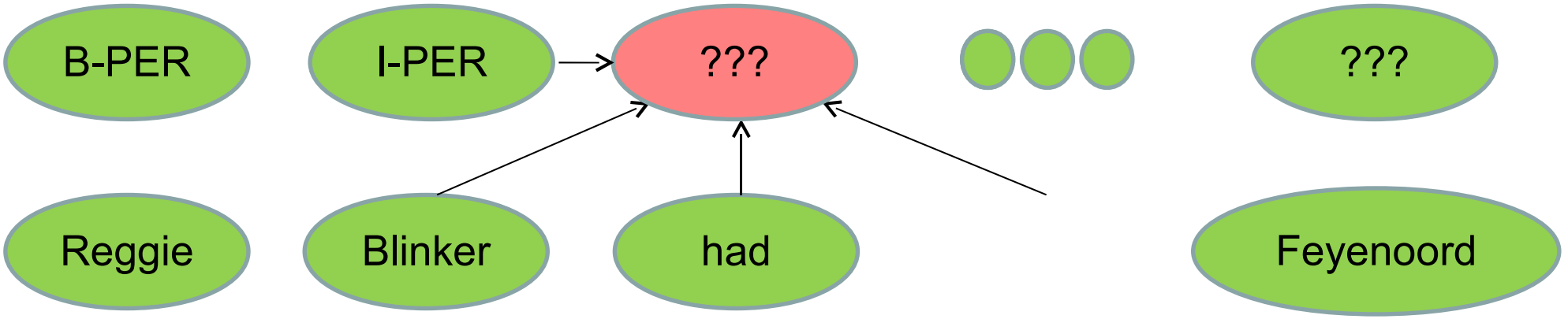


Use Preceptron to assign label to “Blinker” with thee following features:

- Prediction for prev word is: B-Per
- Prev word is “Reggie”
- Prev word is capitilized
- Current word is “Blinker”
- Current word is capitilized
- Next word is “had”
- ...

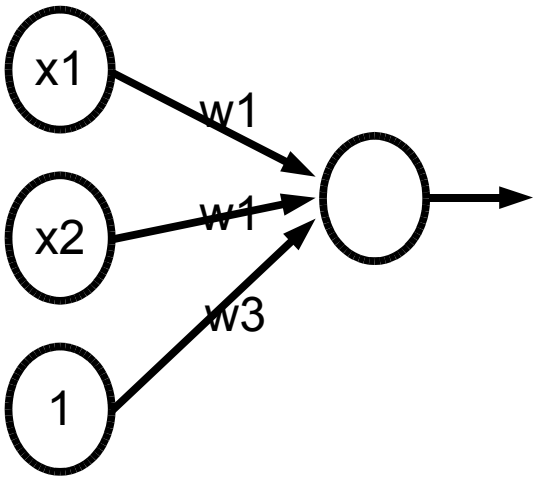


Modeling NER.



Use Preceptron to assign label to “had” with thee following features:

- Prediction for prev word is: I-Per
- Prev word is “Blinker”
- Prev word is capitilized
- Current word is “had”
- Current word is not capitilized
- Next word is “his”



Complete list of baseline features

- Tokens in the window C=[-2,+2]
- Capitalization of tokens in C.
- Previous 2 predictions
- Conjunction of previous prediction and C.
- Prefixes and suffixes of the current token.
- Normalized digits (22/12/2009 ---> *DD*/*DD*/*DDDD*)
- Overall around 15 active features per sample.

Adding Word Representations

- Brown clusters:
  - Prefixes of length 4, 6, 10, 20 for words in the window C=[-2,+2] tokens
  - Conjunctions of the above prefixes and the previous prediction.
- NN embeddings.
  - The 50/100 dimension embedding vectors for words in the window C=[-2,+2] tokens.
  - Conjunctions of the embeddings and the previous prediction.
  - Normalization needed.

Results

Resources	CoNLL Test	CoNLL Dev	Muc7 Dry	Muc7 Formal	Webpages
Baseline	84.58	89.85	69.86	67.29	54.67
Reuters C&W, 50 dim	87.36	91.52	75.27	74.16	56.11
Reuters HLBL, 50 dim	87.43	91.51	74.51	74.33	54.24
Reuters HLBL, 100 dim	88.20	91.49	75.53	75.68	54.52
Reuters Brown Cluster	88.74	92.19	80.11	80.08	56.29
Reuters+Wall Street Journal+Wikipedia Brown Clusters	89.49	92.57	82.61	78.93	58.70

Conclusions

- Word representation extracted from large amounts of unlabeled data improve the performance
- Brown clusters are faster and result in better performance. Maybe due to the sparsity.
- We have a state of the art NER system that achieves 90.8F1 on CoNLL test. It's publicly available. Use it: http://l2r.cs.uiuc.edu/~cogcomp/LbjNer.php