~~Specializing Word Embeddings (for Parsing) by Information Bottleneck~~

Threshing Word Embeddings

Xiang Lisa Li and Jason Eisner

JOHNS HOPKINS UNIVERSITY

CENTER FOR LANGUAGE AND SPEECH PROCESSING

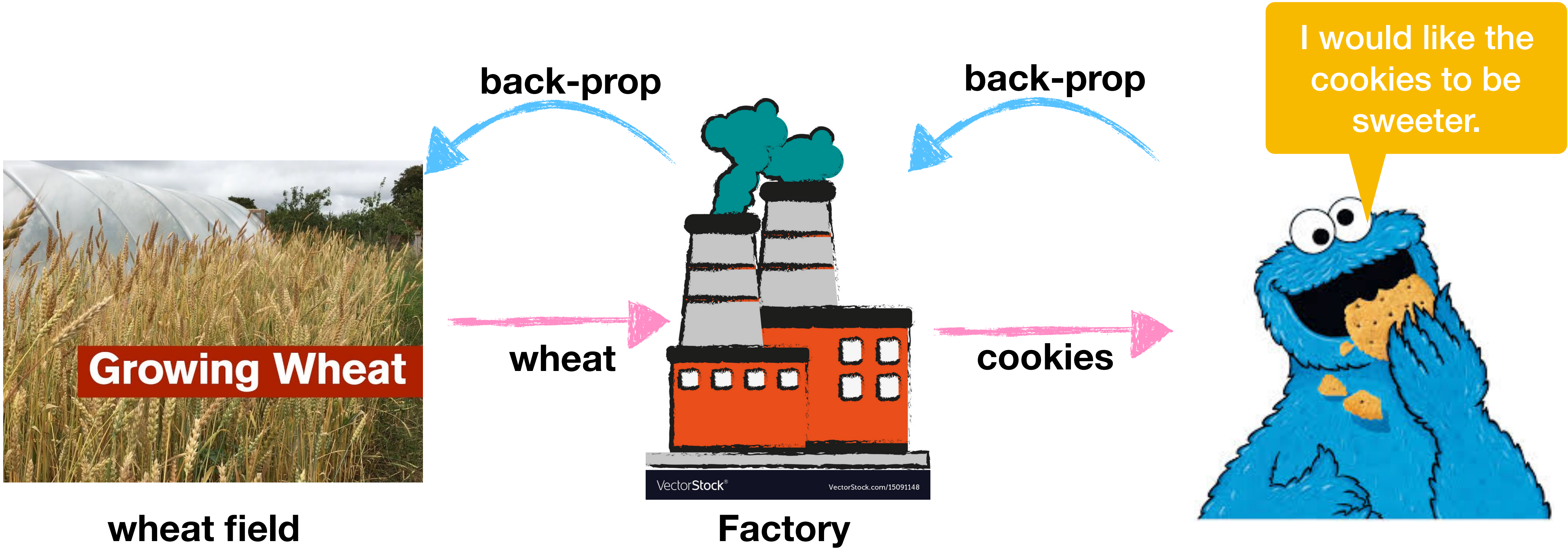
Pre-trained Word Embeddings

Syntactic Knowledge

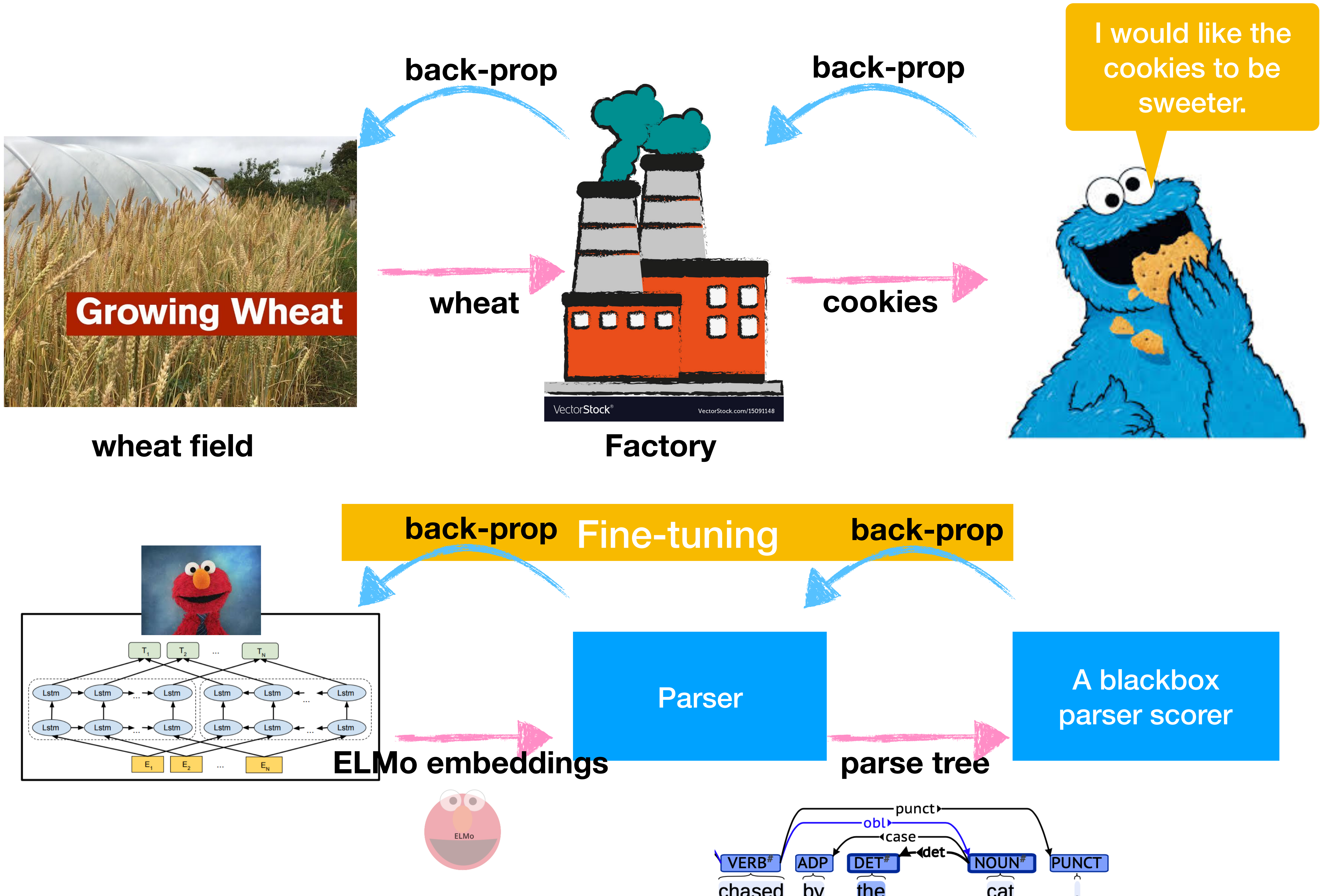
How to extract syntax automatically?

Specialize the Embeddings!

for the task (parsing)


back-prop
back-prop
I would like the cookies to be sweeter.
Growing Wheat
wheat
cookies
VectorStock®
VectorStock.com/15091148
wheat field
Factory

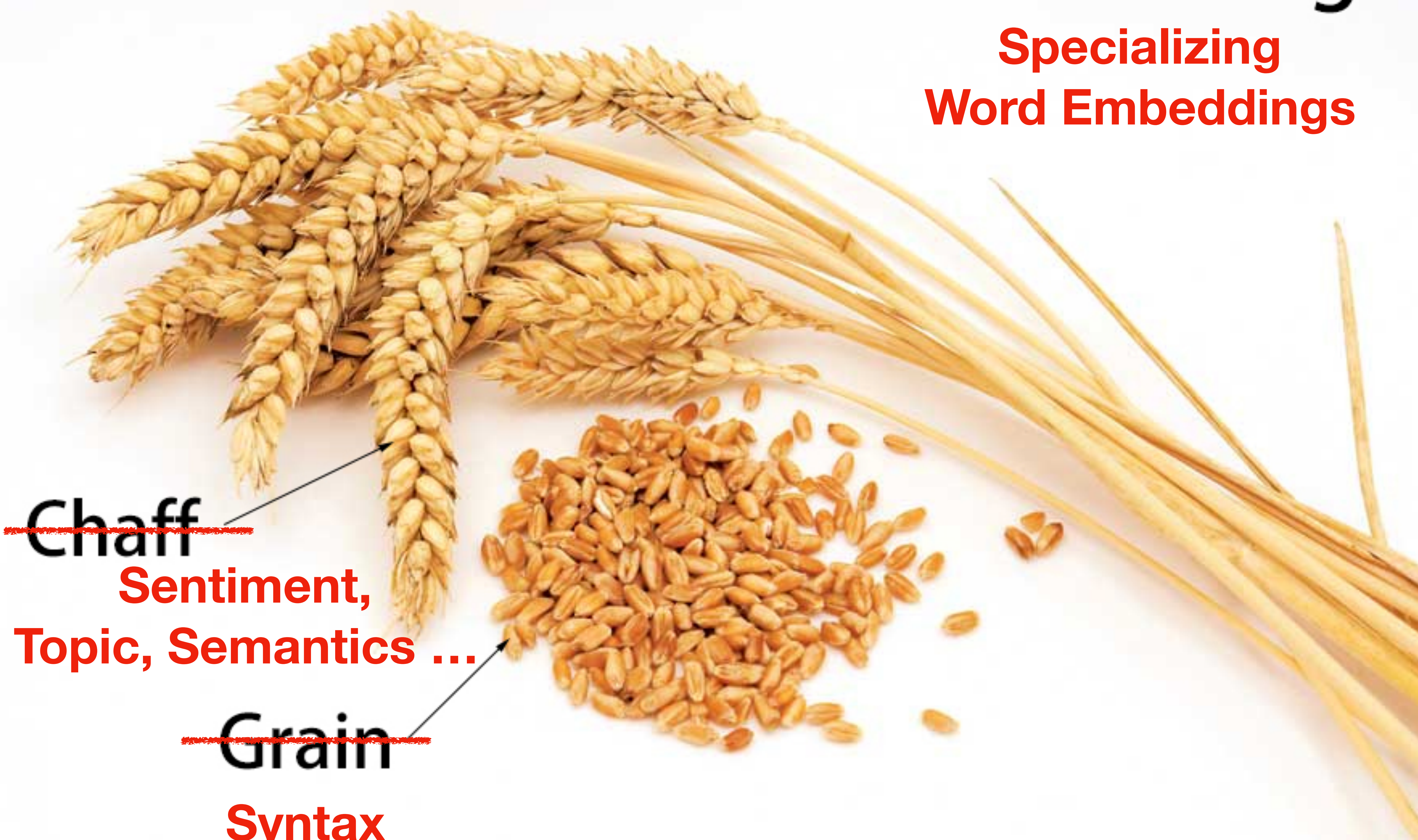
Fine-tuning



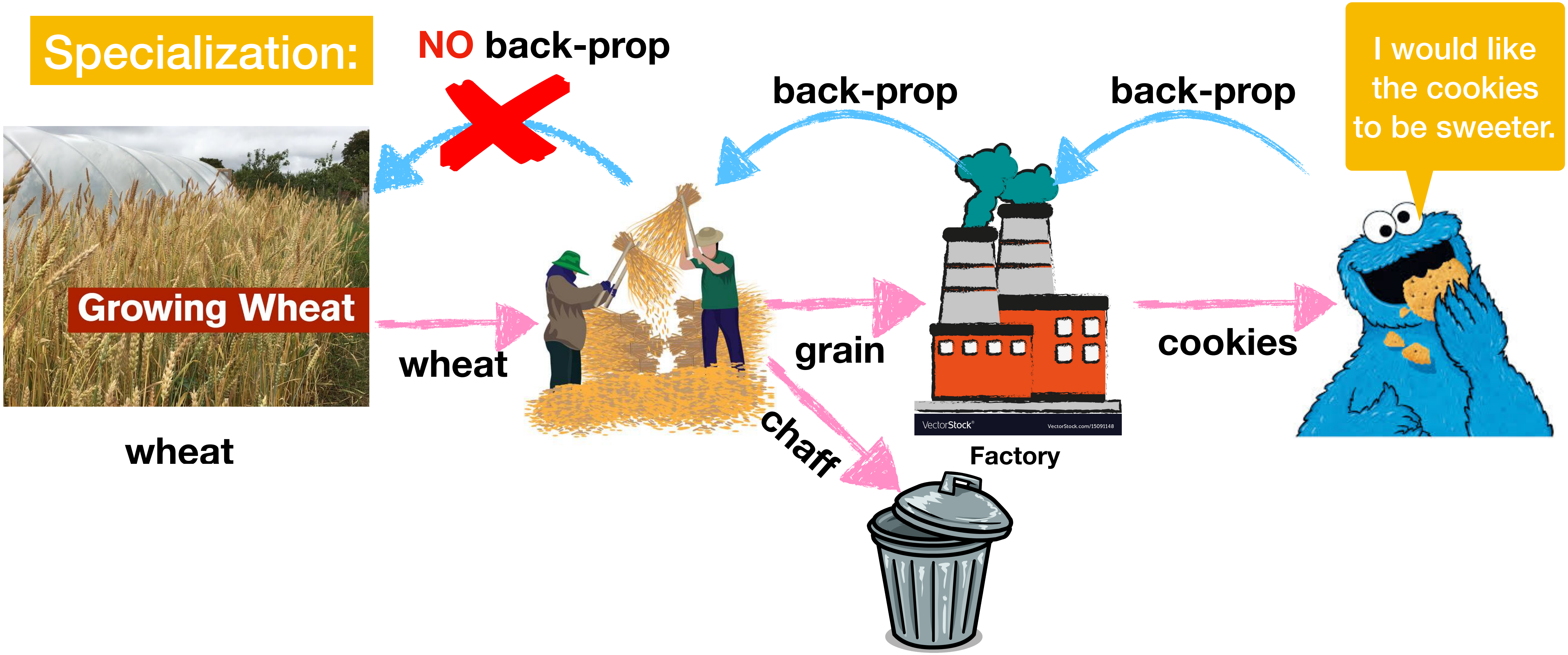
back-prop
back-prop
I would like the cookies to be sweeter.
Growing Wheat
wheat
cookies
VectorStock
wheat field
Factory
back-prop
Fine-tuning
back-prop
Lstm
ELMo embeddings
ELMo
Parser
parse tree
A blackbox parser scorer
punct
obl
case
det
VERB
ADP
DET
NOUN
PUNCT
chased
by
the
cat
.

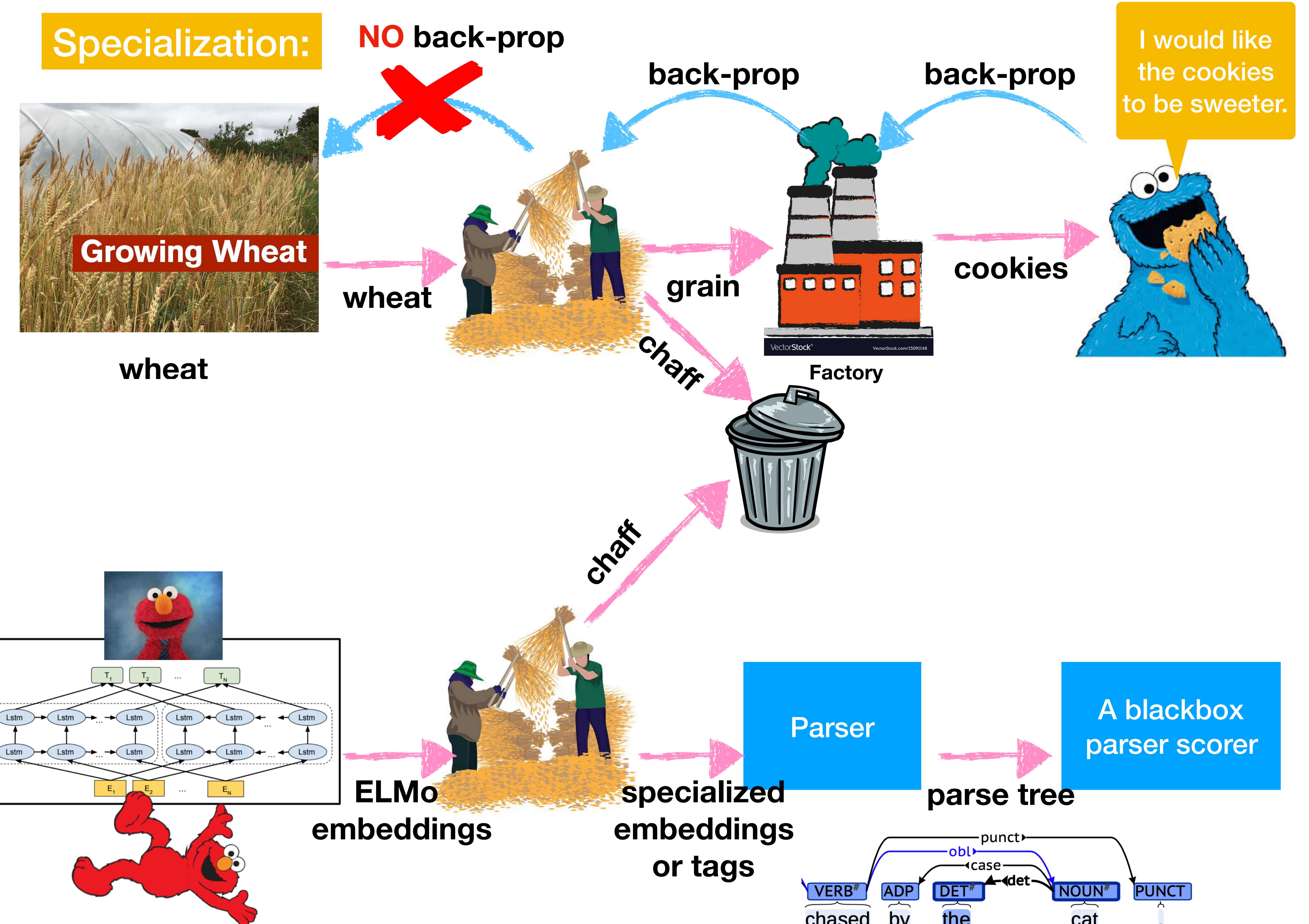
What Is ~~Threshing~~?

Specializing Word Embeddings

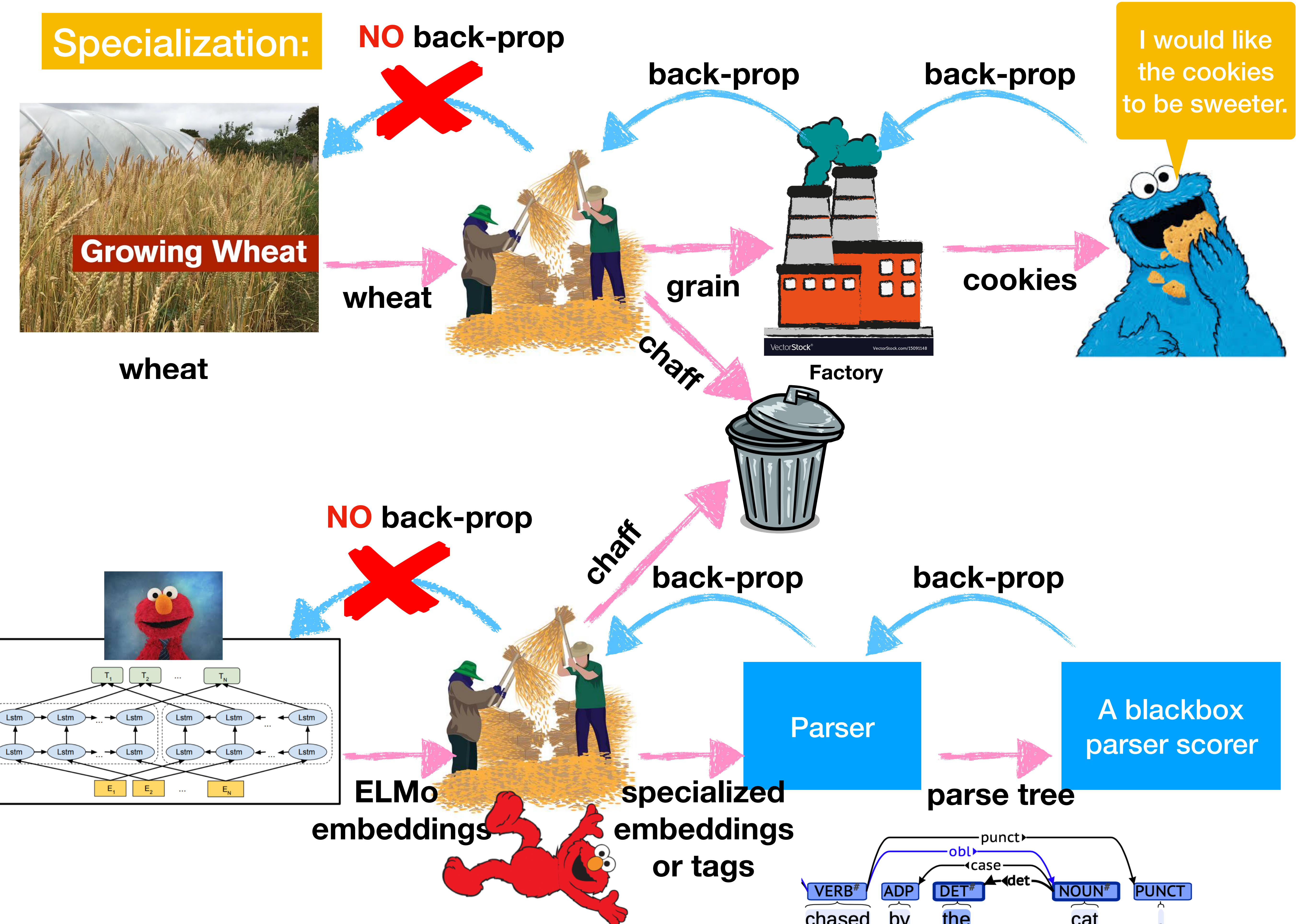


Specialization:

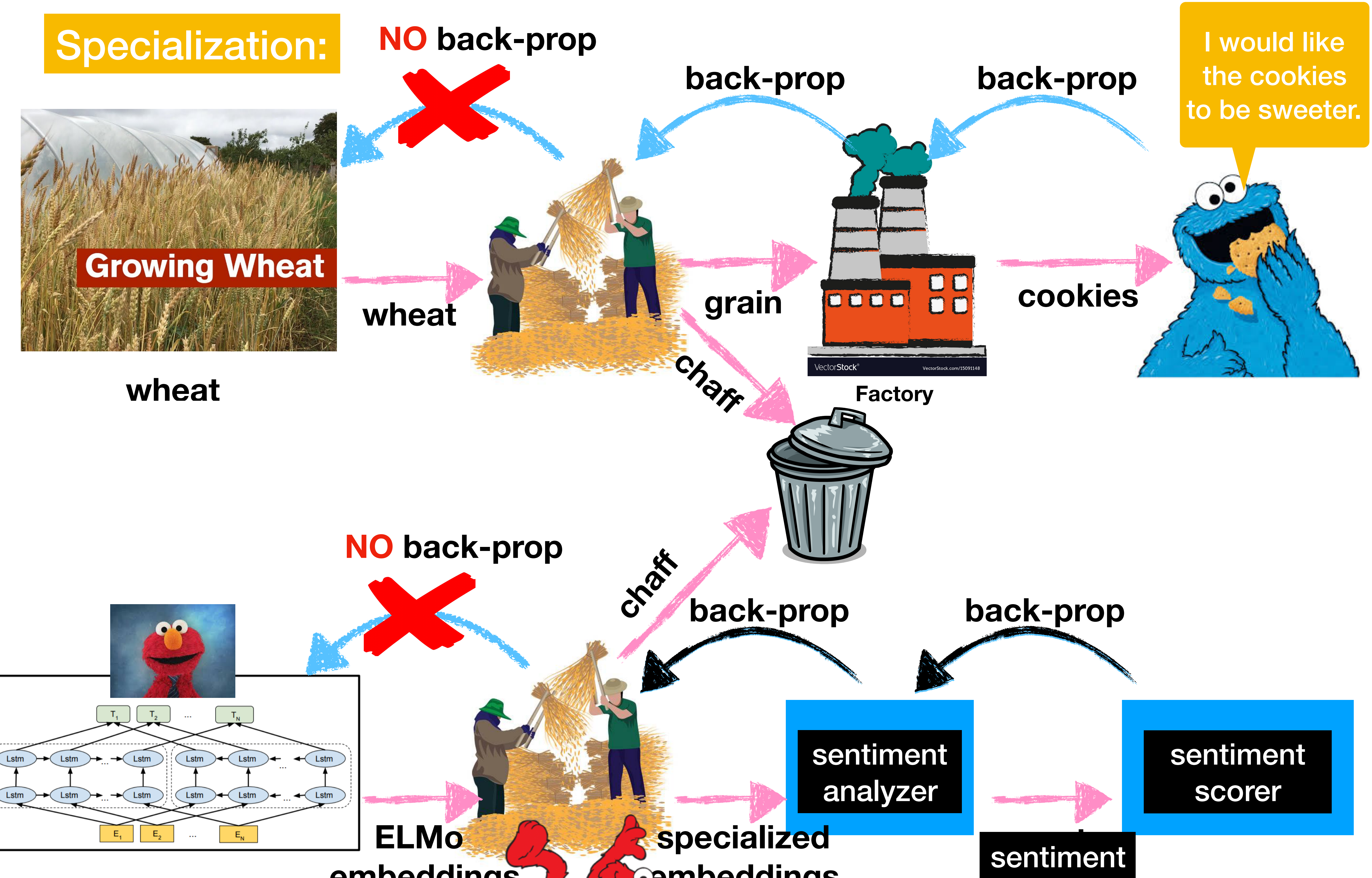




Specialization:
NO back-prop
back-prop
back-prop
I would like the cookies to be sweeter.
Growing Wheat
wheat
wheat
grain
cookies
chaff
Factory
VectorStock
chaff
Lstm
ELMo embeddings
specialized embeddings or tags
Parser
parse tree
A blackbox parser scorer
punct
obl
case
det
VERB
ADP
DET
NOUN
PUNCT
chased
by
the
cat
.


Specialization:
NO back-prop
back-prop
back-prop
I would like the cookies to be sweeter.
Growing Wheat
wheat
grain
cookies
wheat
chaff
Factory
NO back-prop
chaff
back-prop
back-prop
Parser
A blackbox parser scorer
ELMo embeddings
specialized embeddings or tags
parse tree
punct
obl
case
det
VERB
ADP
DET
NOUN
PUNCT
chased
by
the
cat
.

Specialization:

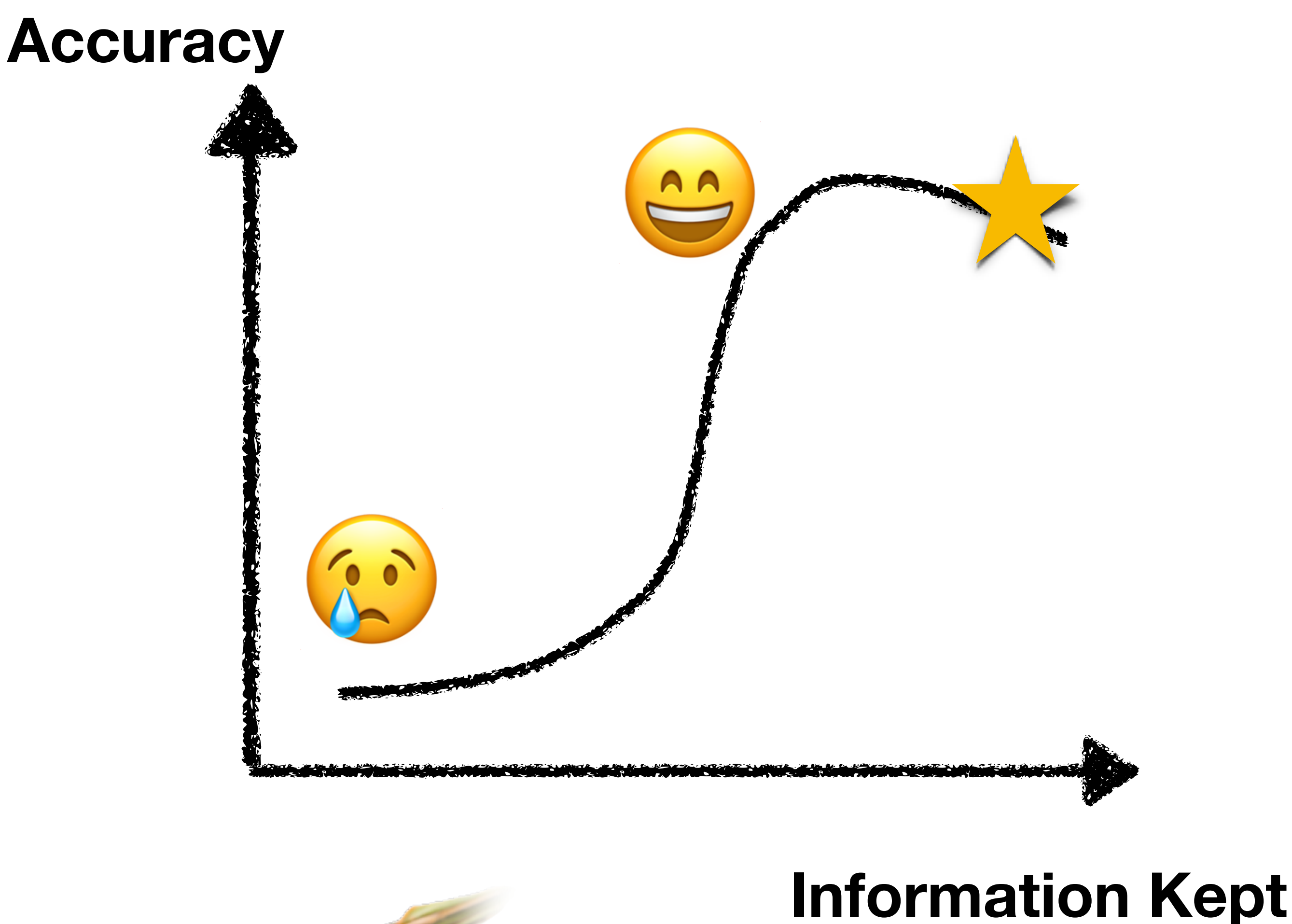


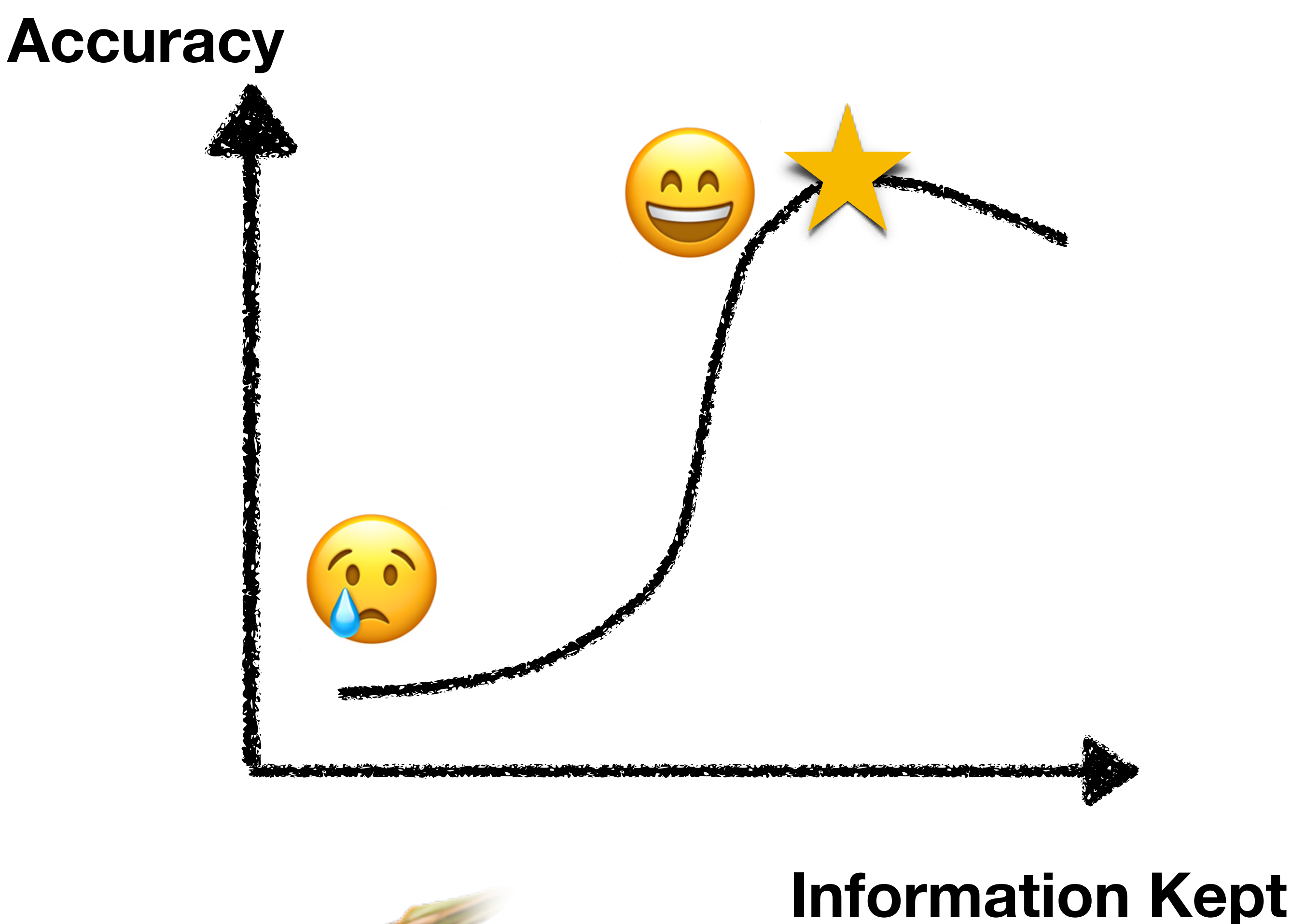
Our specializa… …ique is easily adaptable by swapping blackboxes !

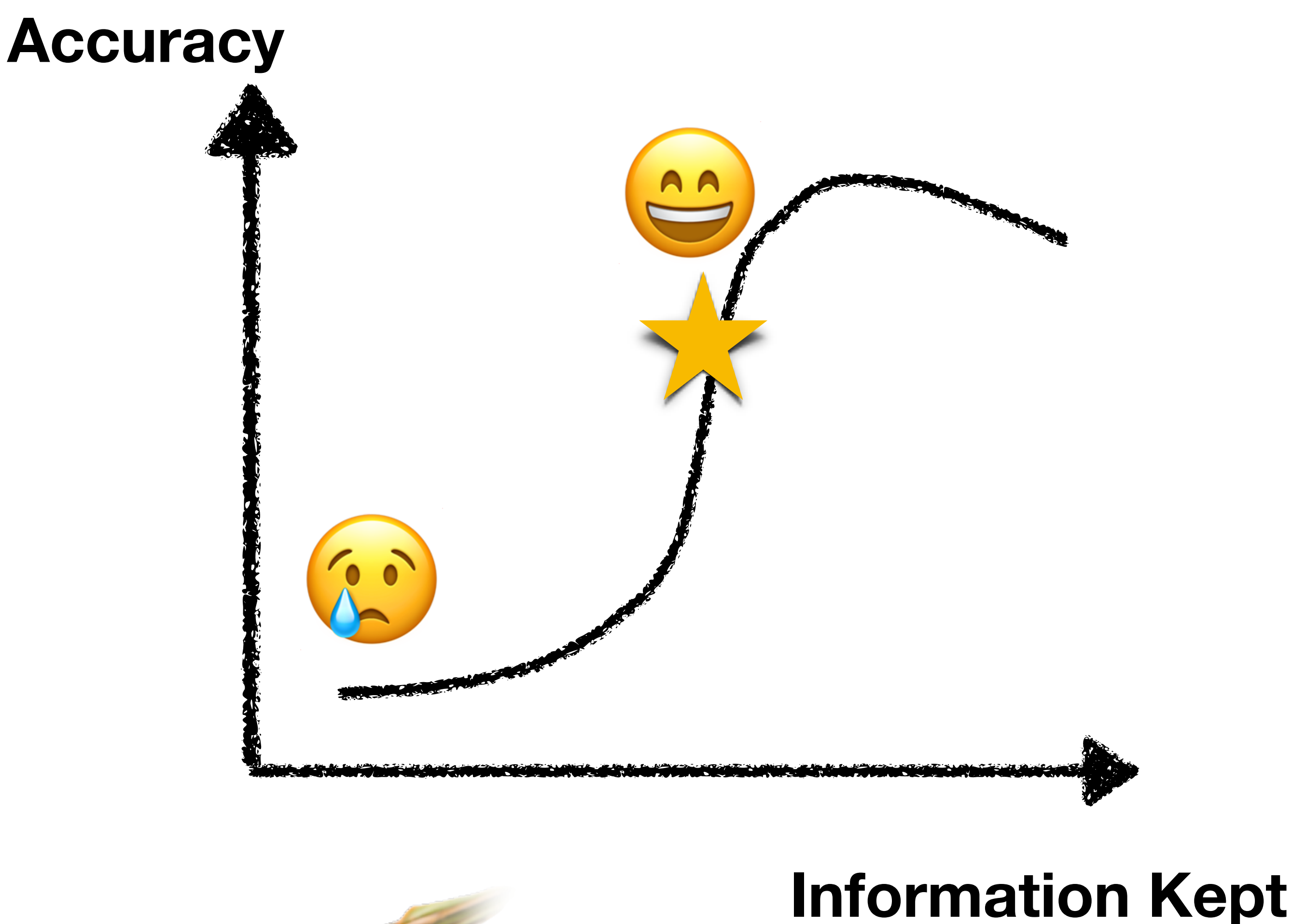
Pros of Specialization

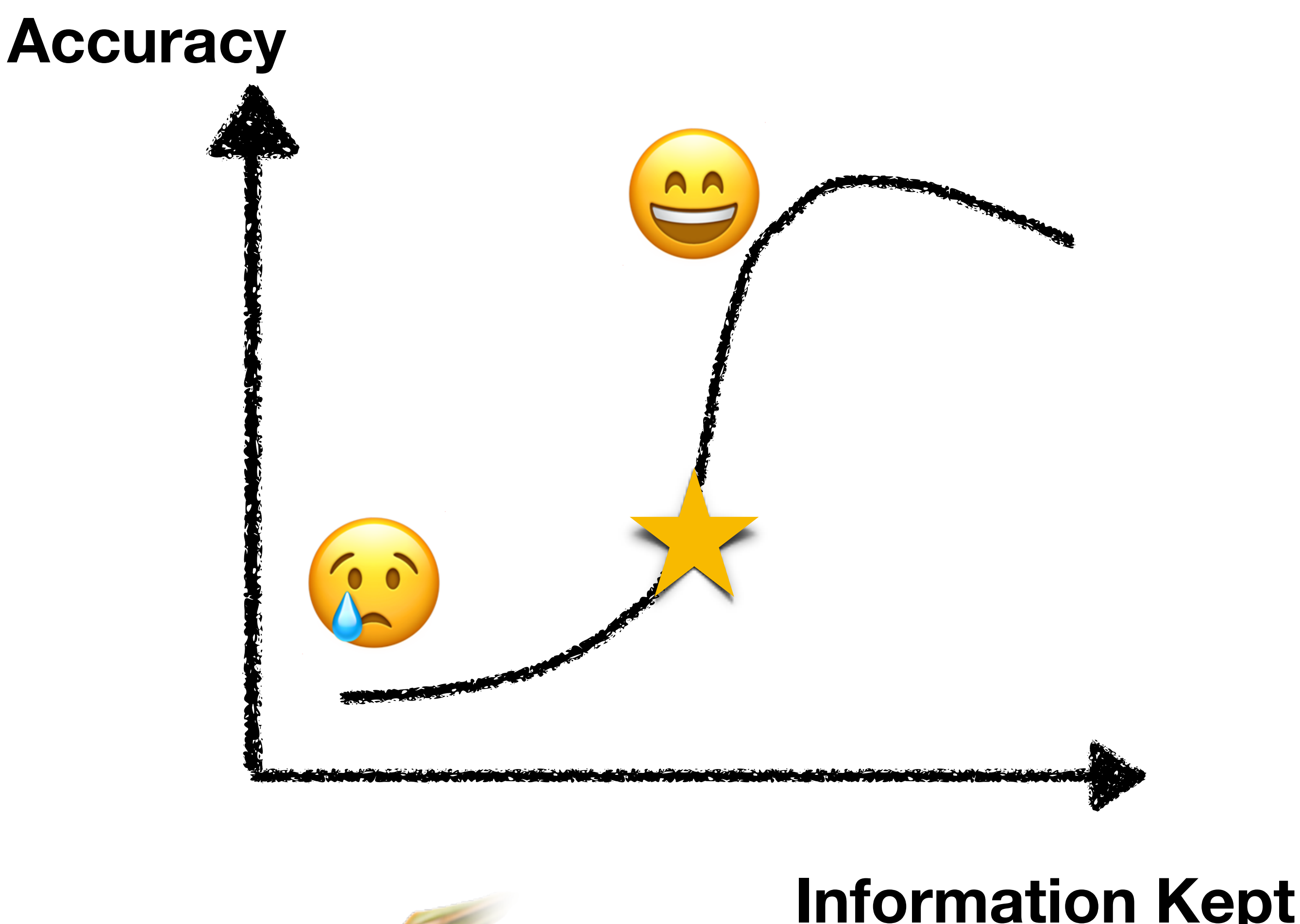
vs. Fine Tuning

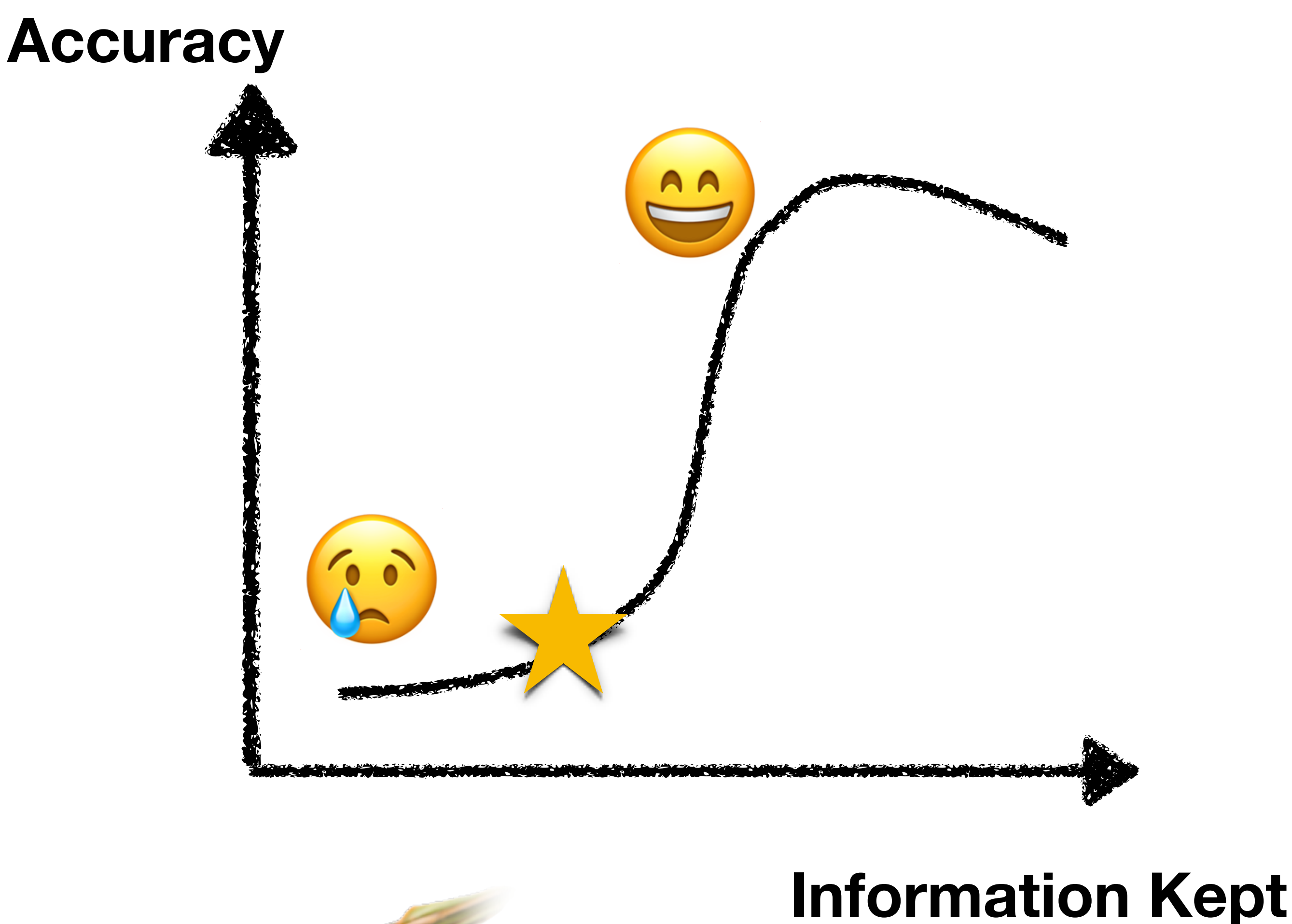
- Faster — trains on 100 sentences per second
- Better generalization — few parameters, so harder to overfit.

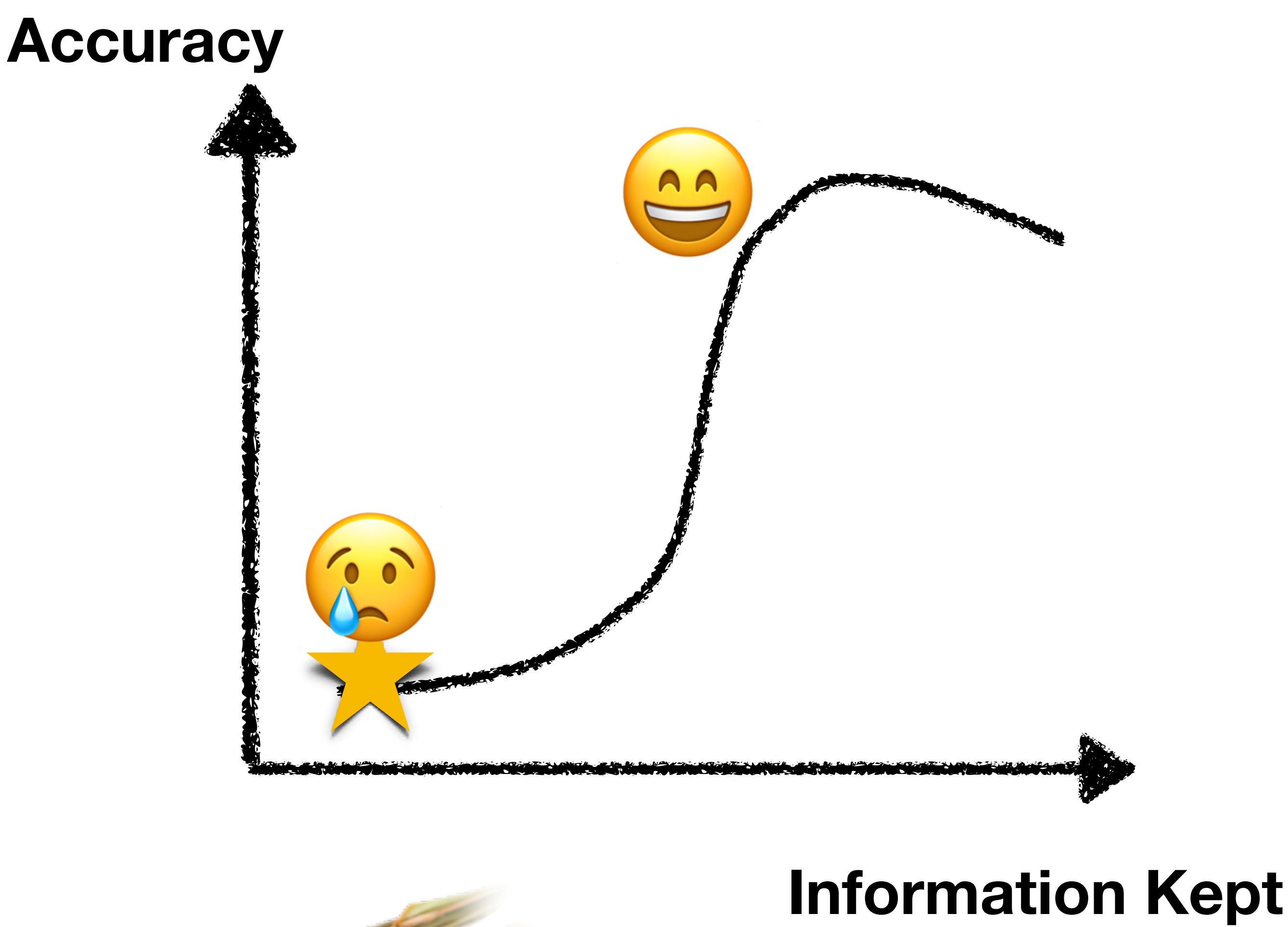

Accuracy
Information Kept

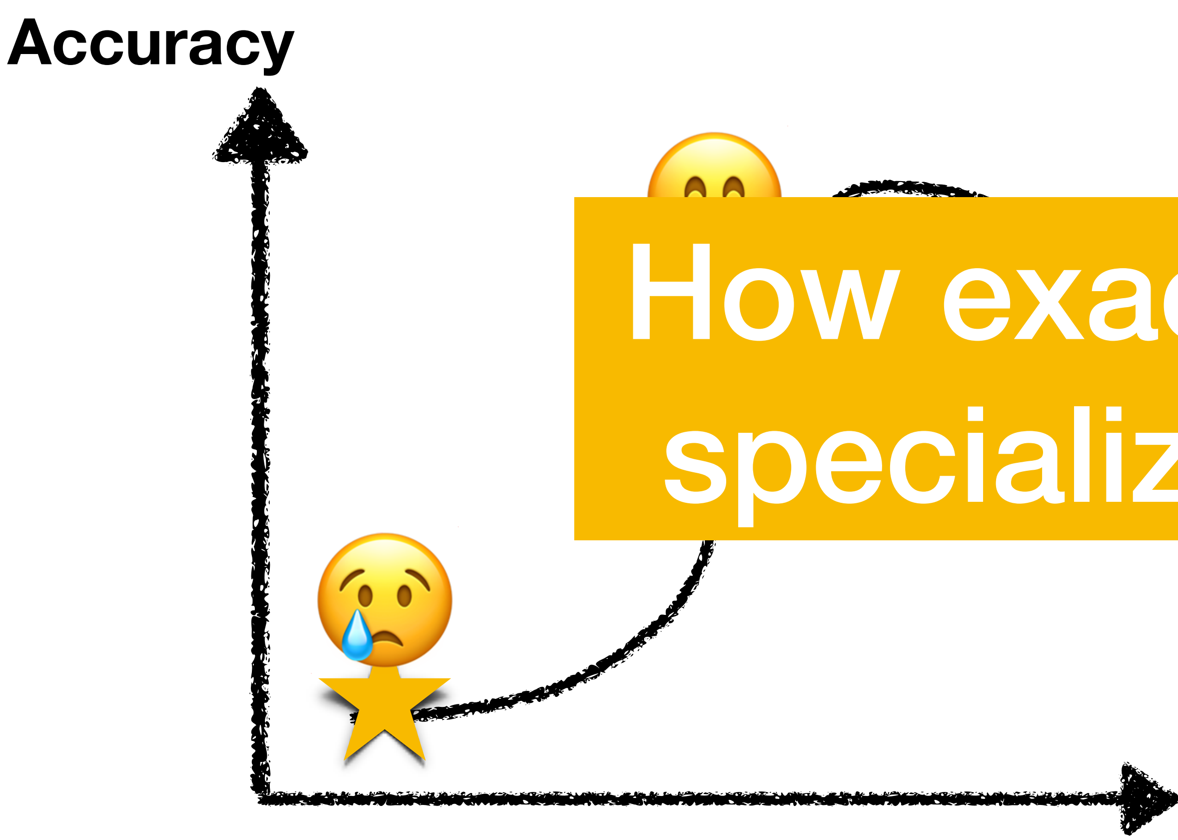

Accuracy
Information Kept


Accuracy
Information Kept


Accuracy
Information Kept


Accuracy
Information Kept


Accuracy
Information Kept

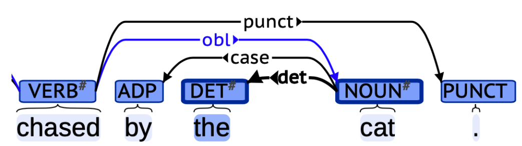


How exactly does the specialization work?

Information Bottleneck


back-prop
chaff
NO back-prop
back-prop
back-prop
Parser
A blackbox parser scorer
ELMo embeddings
specialized taggings
parse tree
I ♥ Trash
punct
obl
case
det
VERB
ADP
DET
NOUN
PUNCT
chased
by
the
cat
.

Information Bottleneck Objective

$\mathcal{L} = -\,\mathrm{I}($ 

$;$  $) + \beta \cdot \mathrm{I}($  $;$  $)$

mutual information between dependency parses and specialized representation.

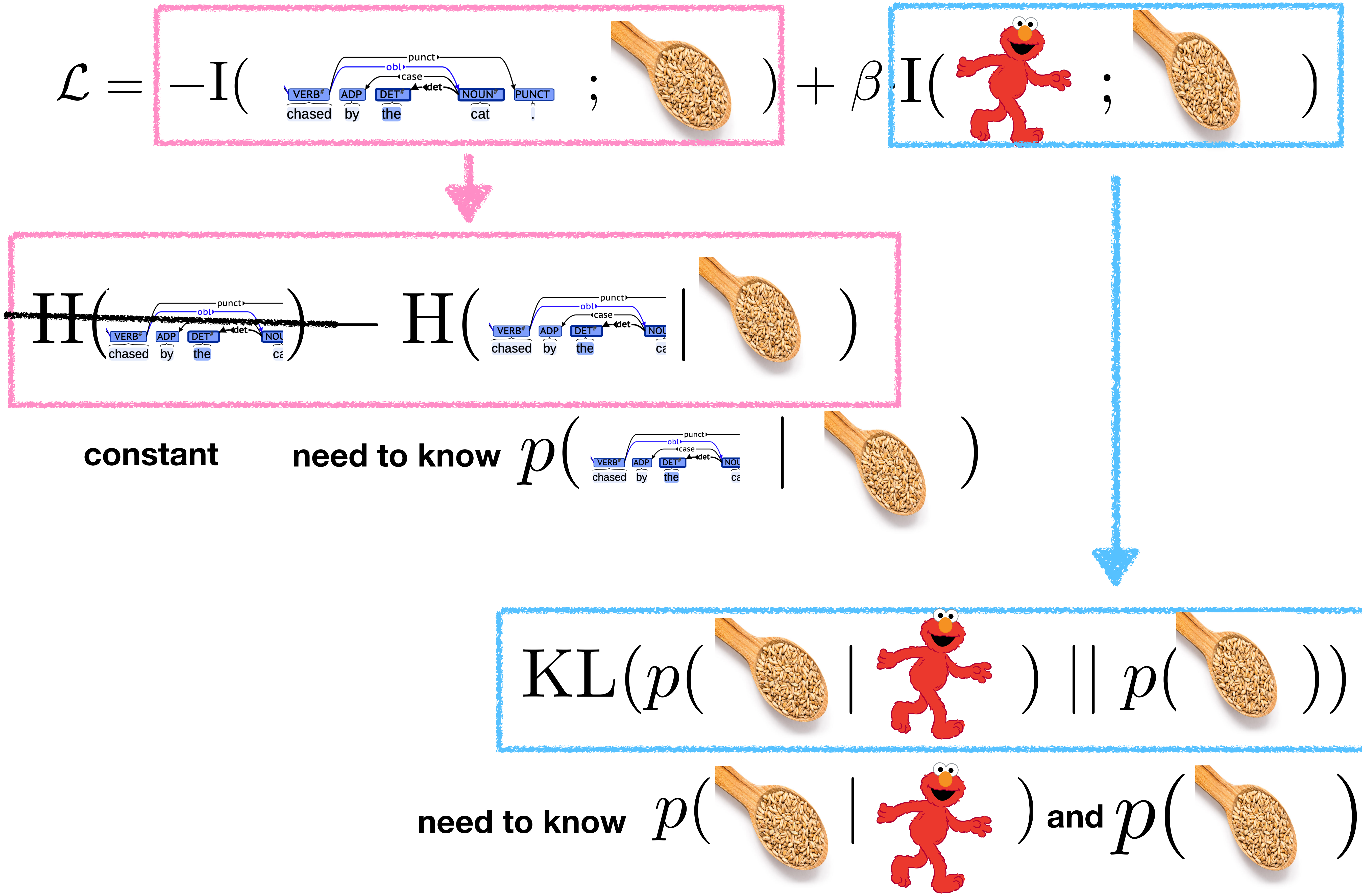
mutual information between pre-trained ELMo embeddings and specialized representation.

keep more syntactic information

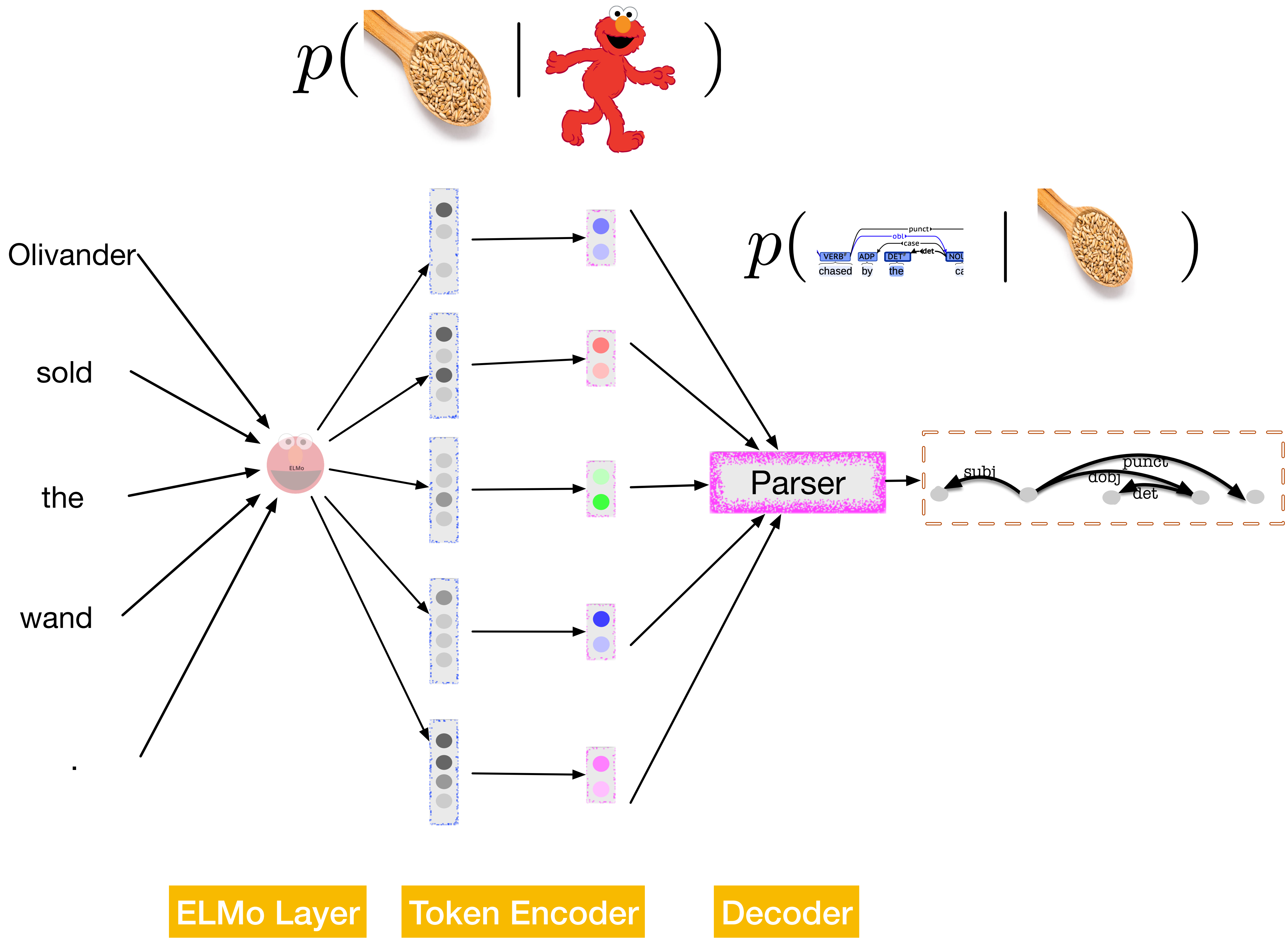
discard more information

bigger β –

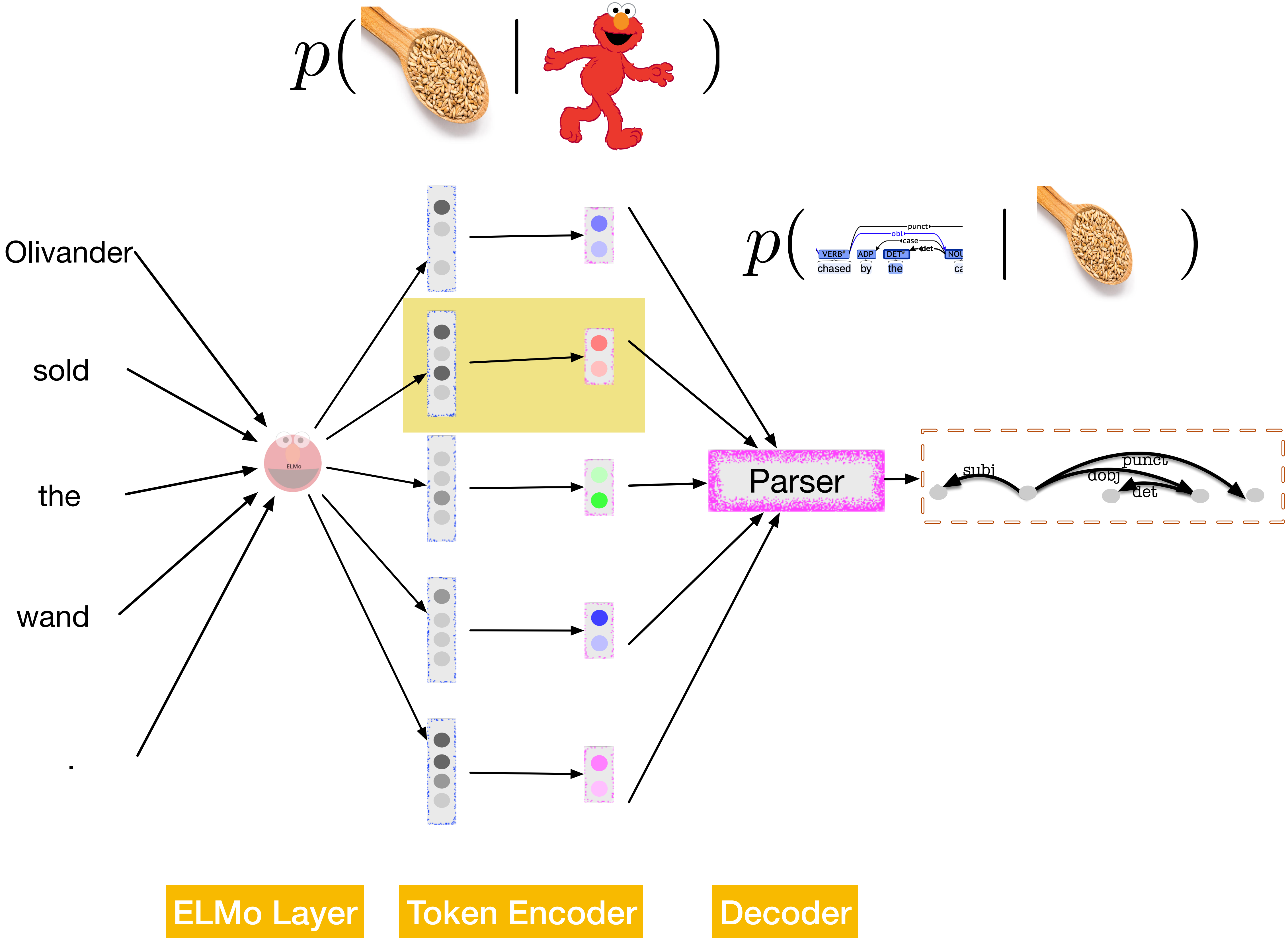
ompression
scar happy



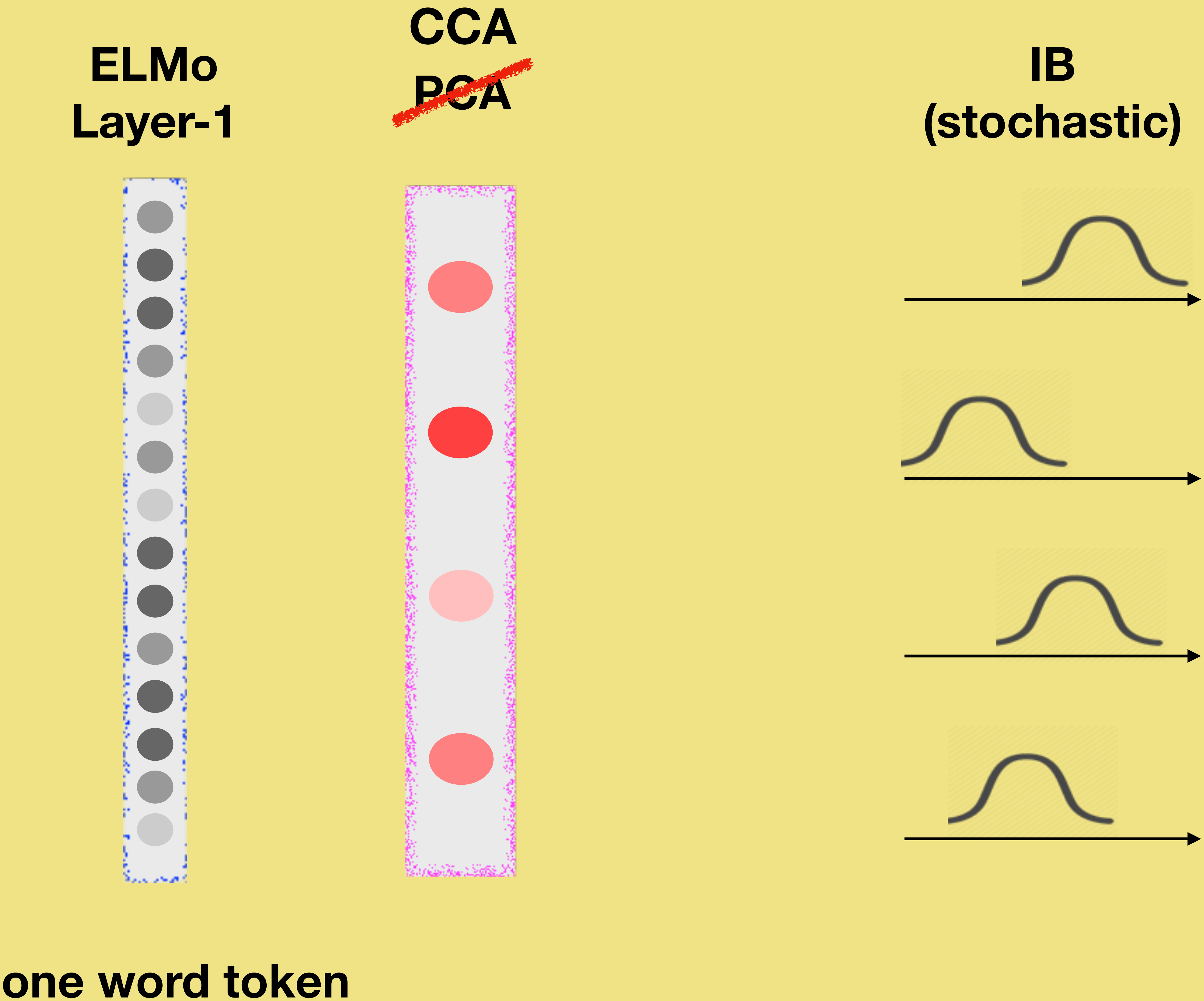
$\mathcal{L} = -\mathrm{I}(\;;\;) + \beta\,\mathrm{I}(\;;\;)$
punct
obl
case
det
VERB ADP DET NOUN PUNCT
chased by the cat .
$\mathrm{H}(\;) - \mathrm{H}(\;|\;)$
constant
need to know $p(\;|\;)$
$\mathrm{KL}(p(\;|\;)\,||\,p(\;))$
need to know $p(\;|\;)$ and $p(\;)$



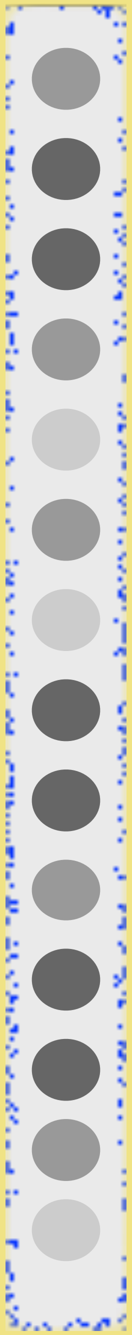
p(|)
Olivander
sold
the
wand
.
ELMo
p(|)
punct
obl
case
det
VERB
ADP
DET
chased
by
the
Parser
subj
punct
dobj
det
ELMo Layer
Token Encoder
Decoder



p(|)
p(|)
punct
obl
case
det
VERB
ADP
DET
chased
by
the
Olivander
sold
the
wand
.
ELMo
Parser
subj
punct
dobj
det
ELMo Layer
Token Encoder
Decoder

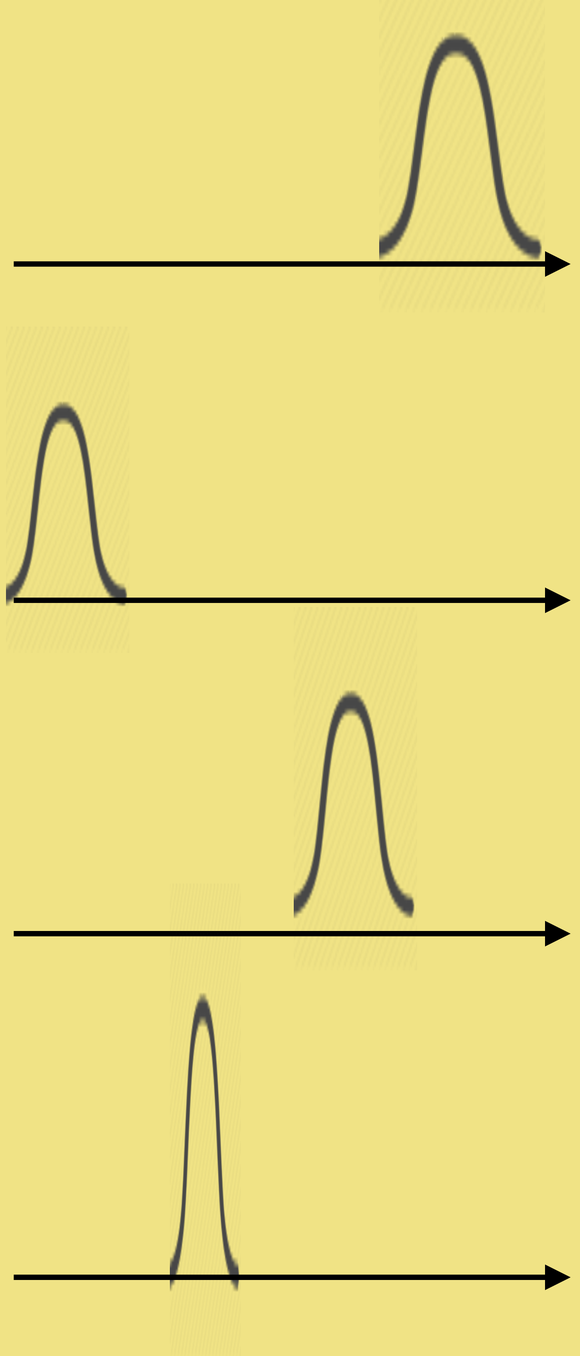

ELMo
Layer-1
CCA
~~PCA~~
IB
(stochastic)
one word token

ELMo Layer-1



CCA

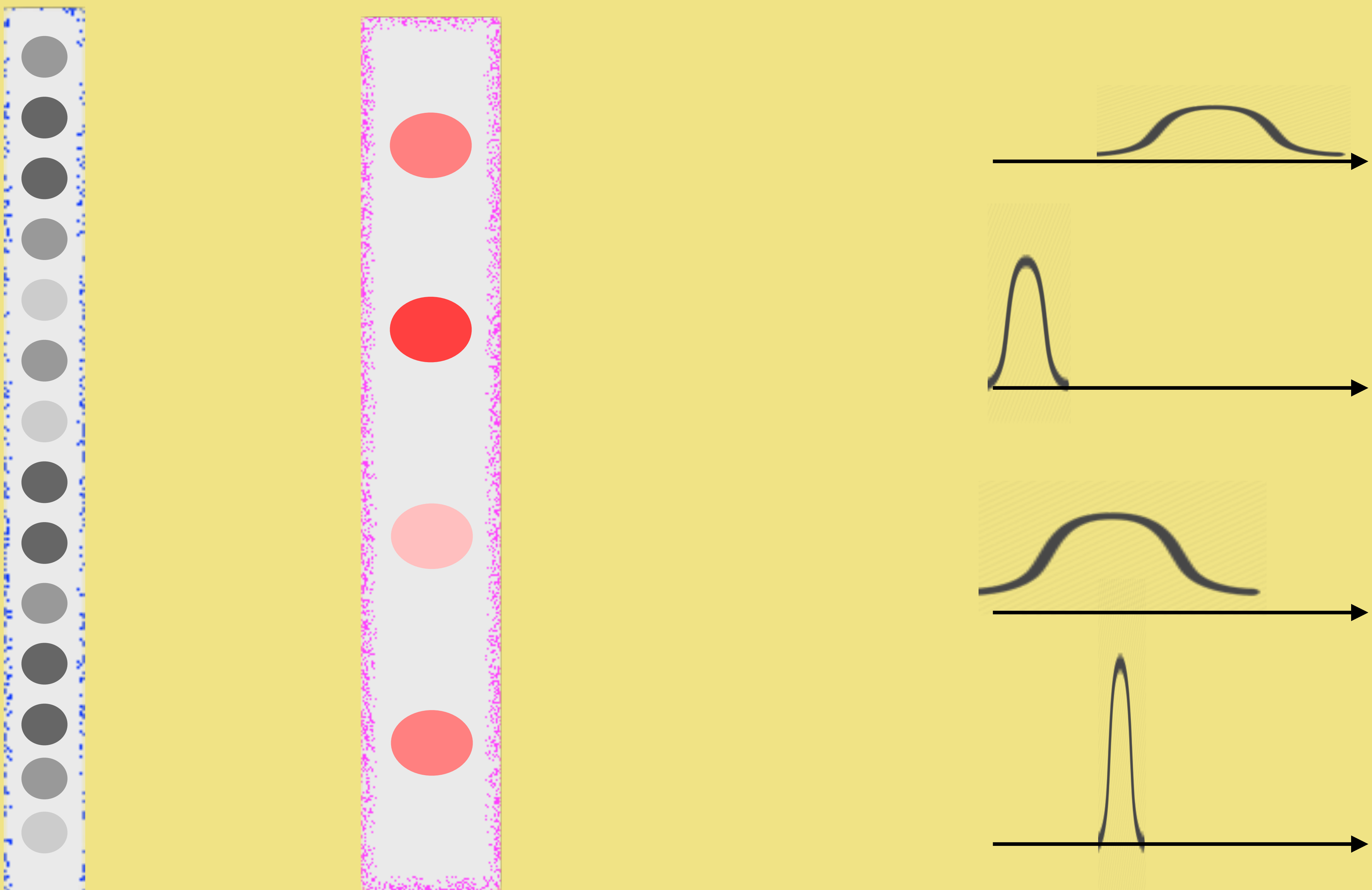
IB (stochastic)



one word token

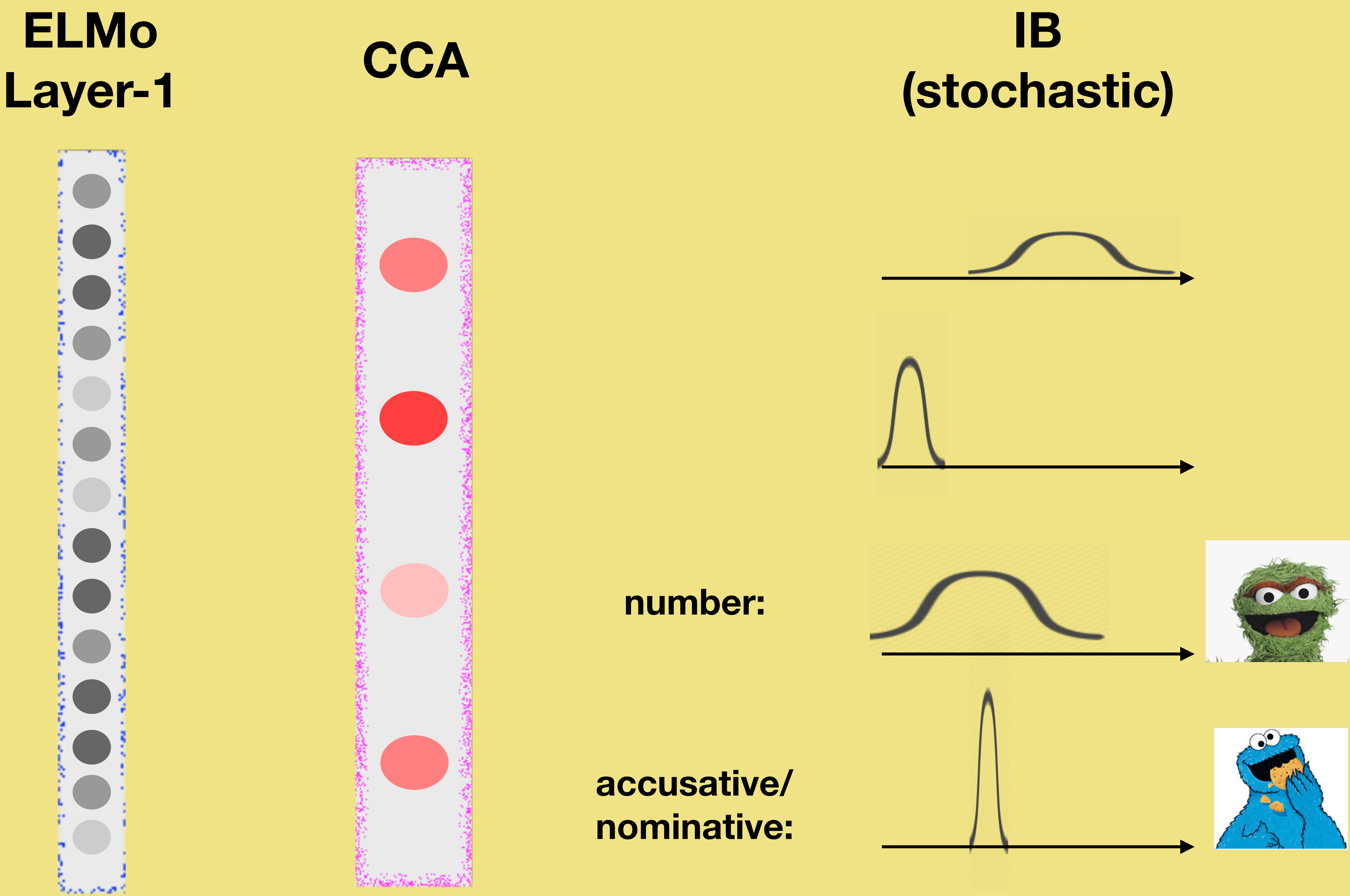
For some tokens, we need all 4 dimensions

ELMo Layer-1 **CCA** **IB (stochastic)**

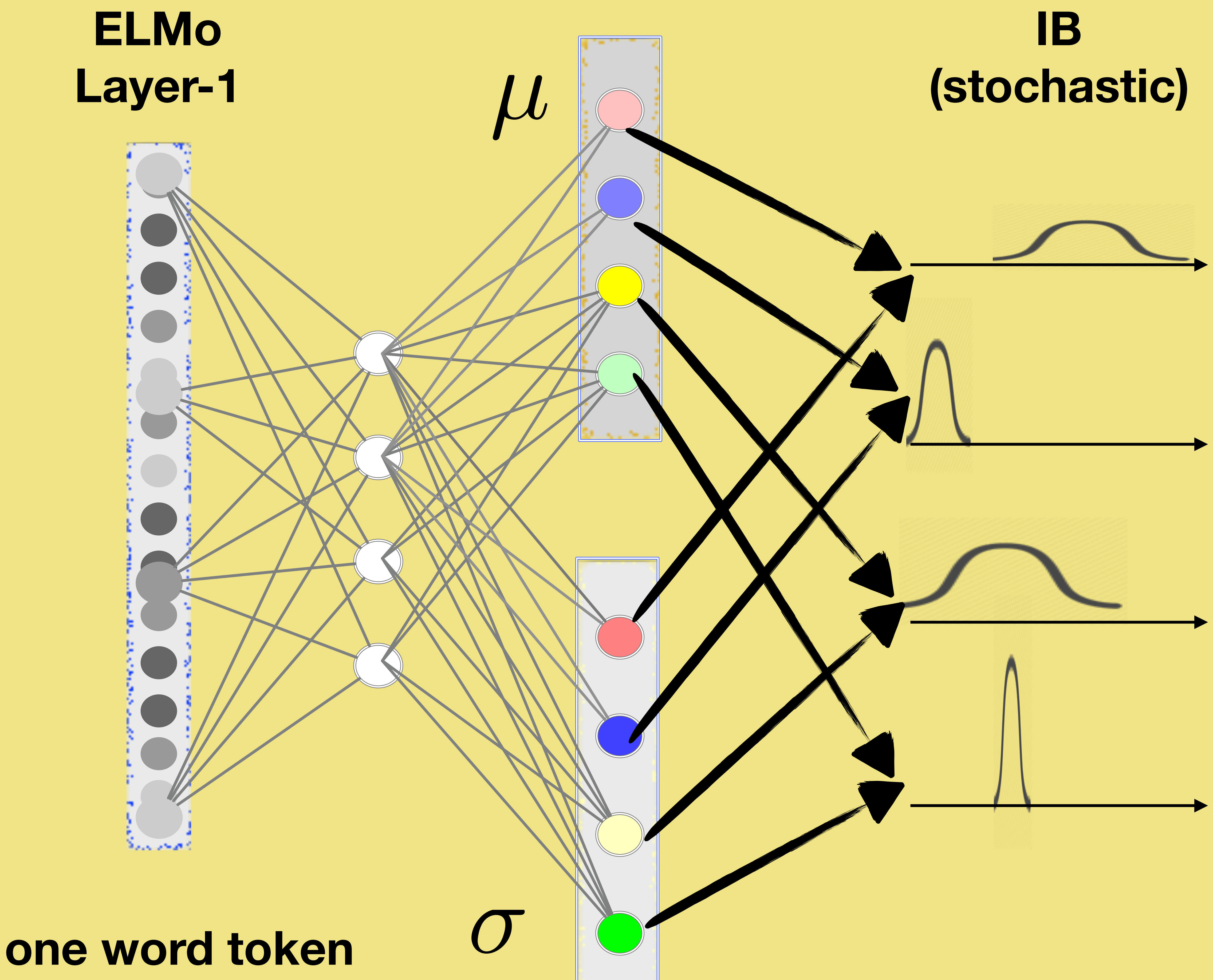


one word token

For some tokens, retaining less information does not hurt parsing.



Example: In English, we don't care whether an object pronoun is singular or plural.

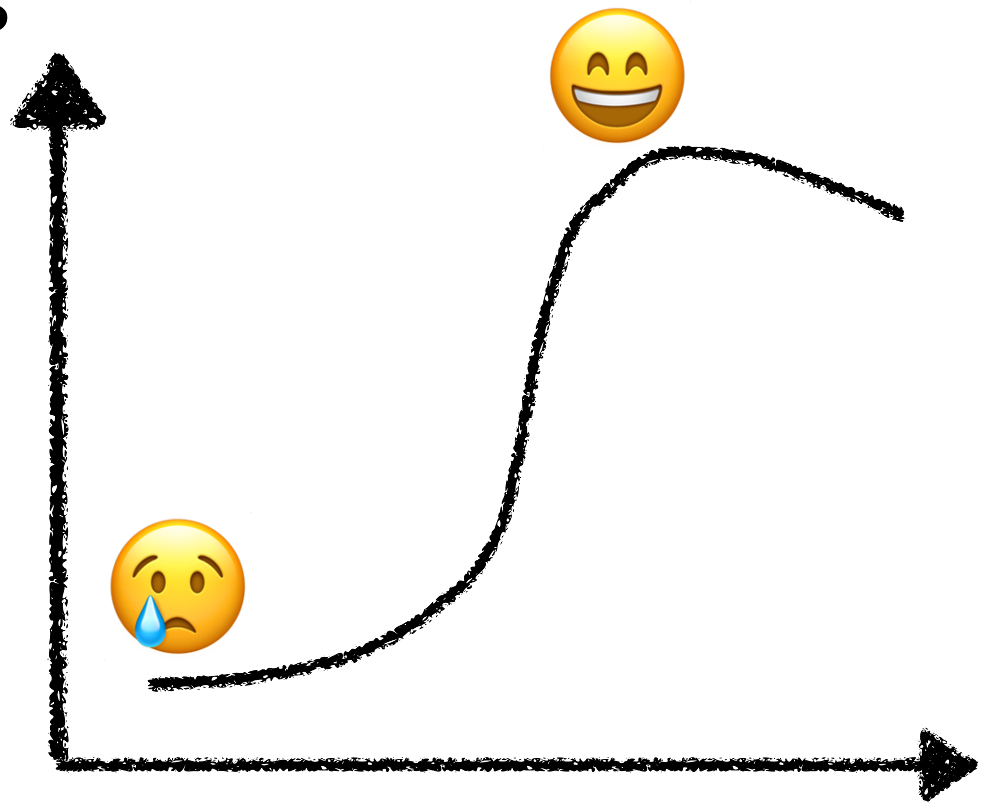


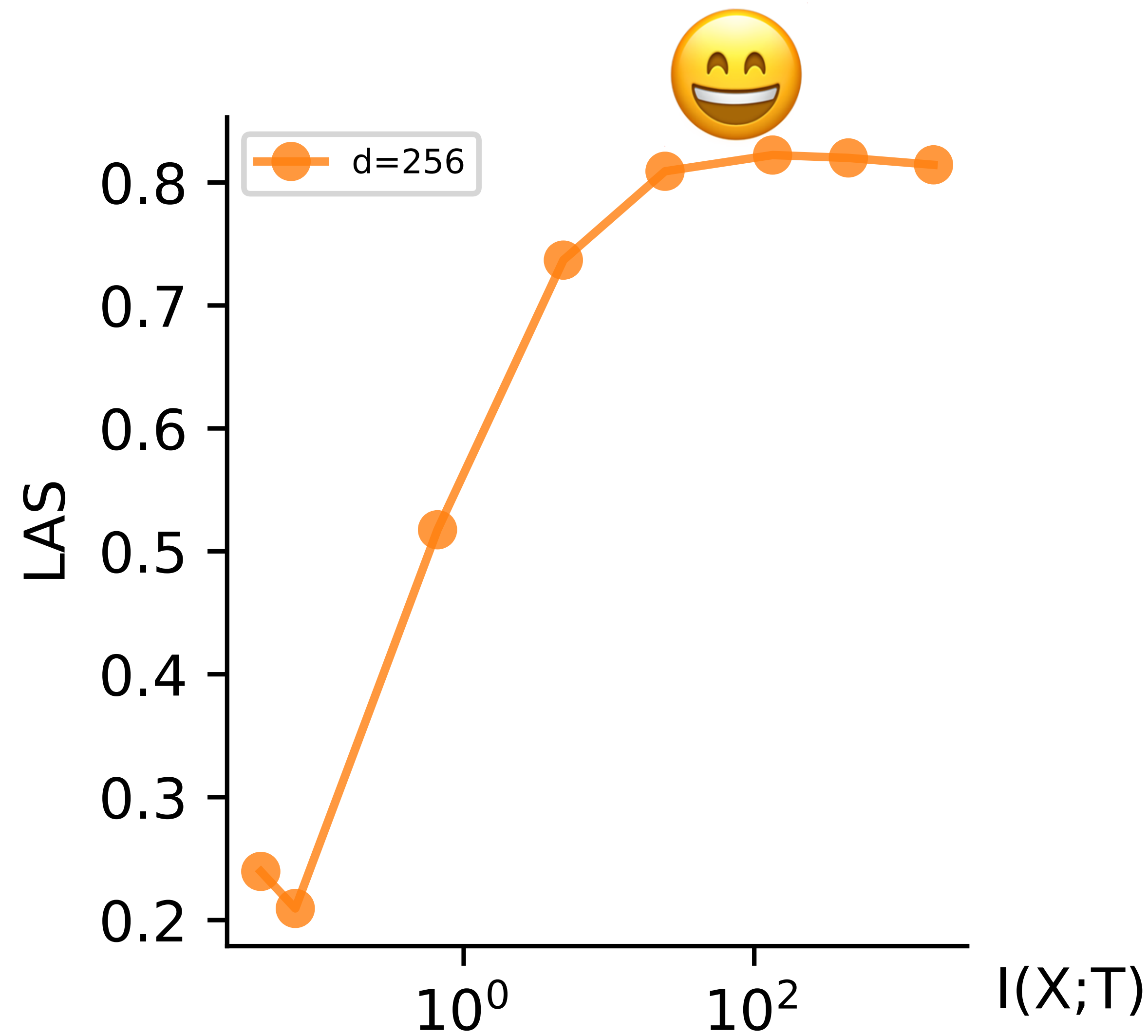
ELMo
Layer-1
μ
IB
(stochastic)
σ
one word token

Cartoon

English

LAS



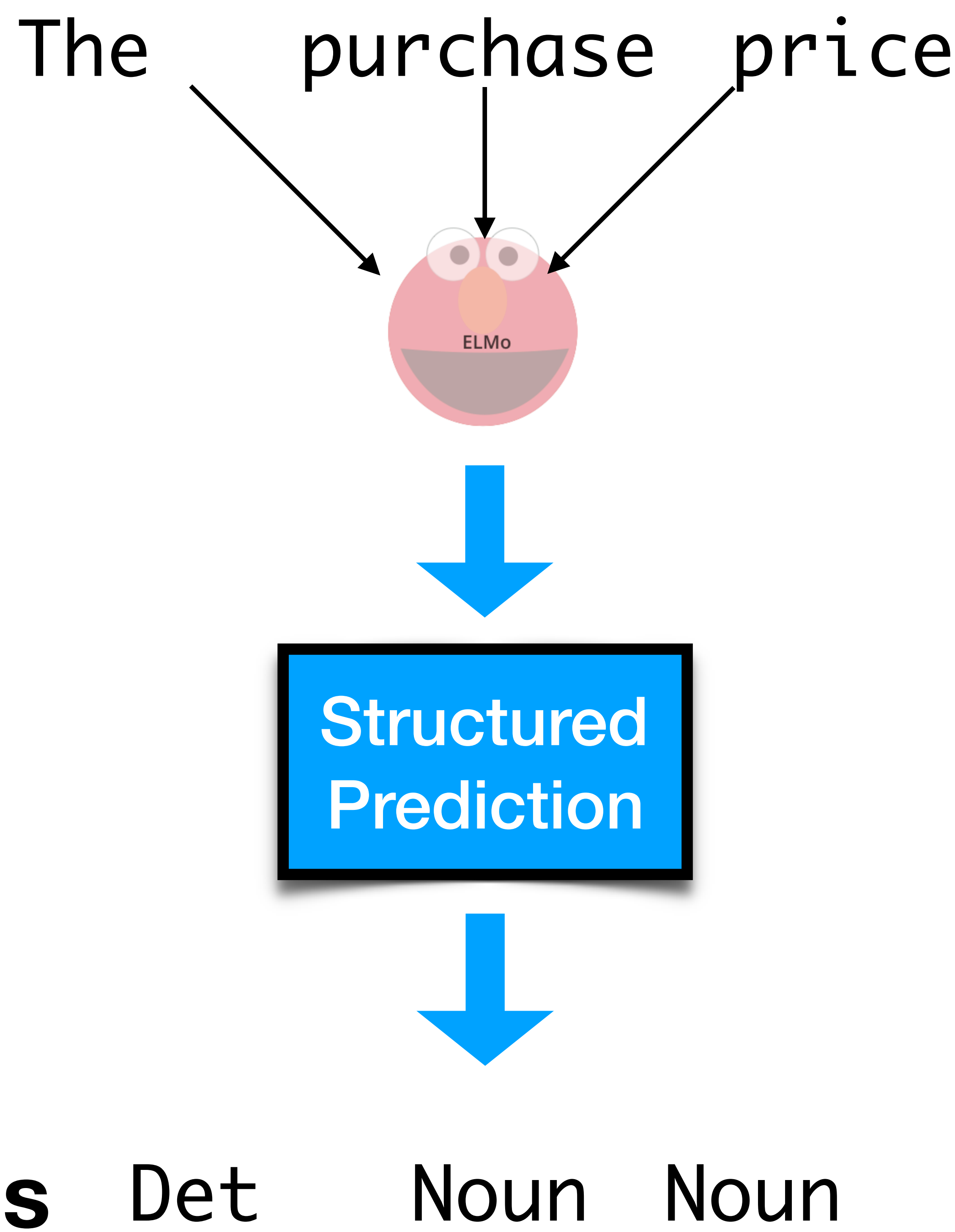


Information Kept

Q: Is our specialization contextual?

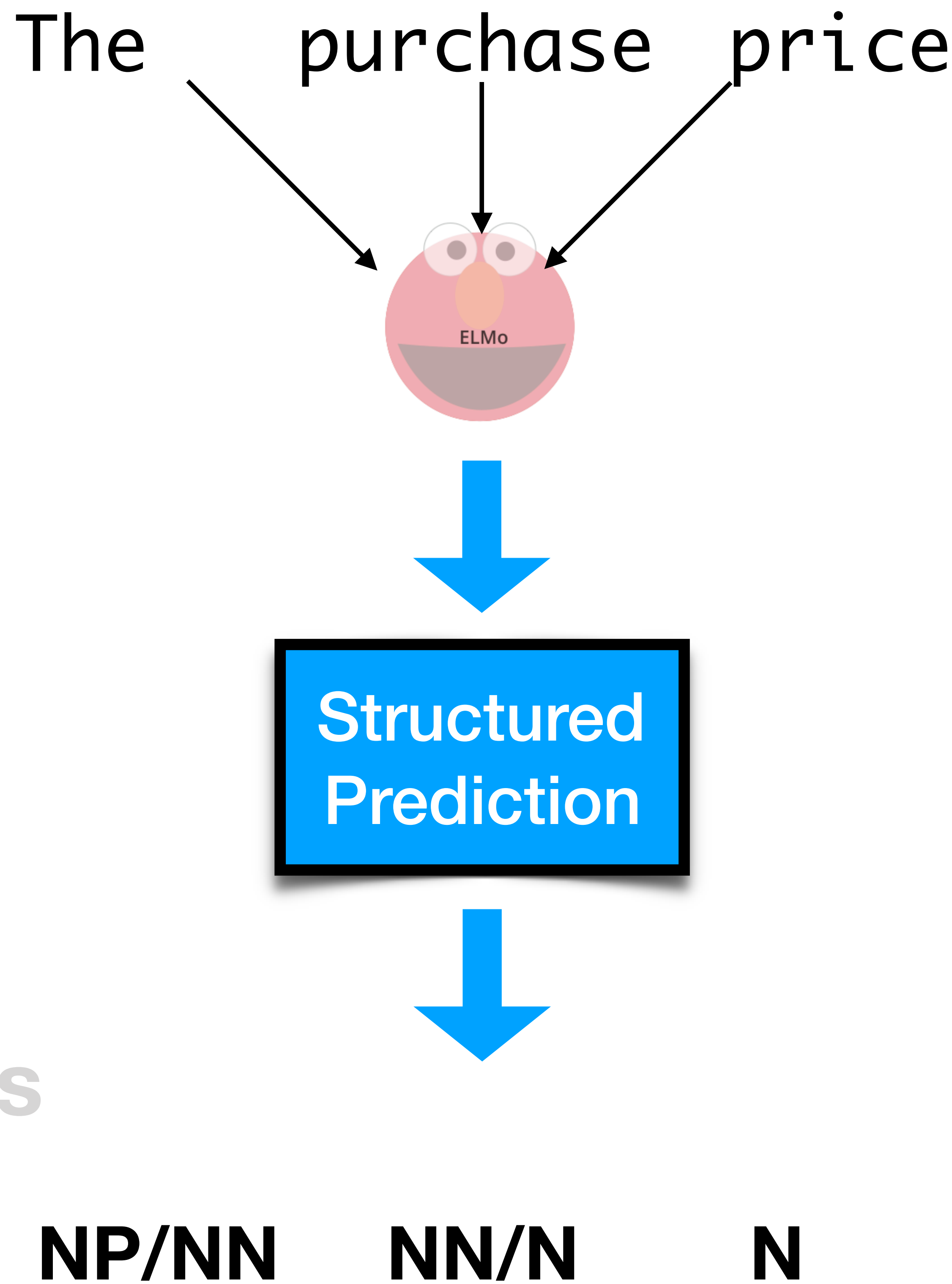
A: YES, because we compress ELMo layer 1, which depends on the context.

ELMo Embeddings



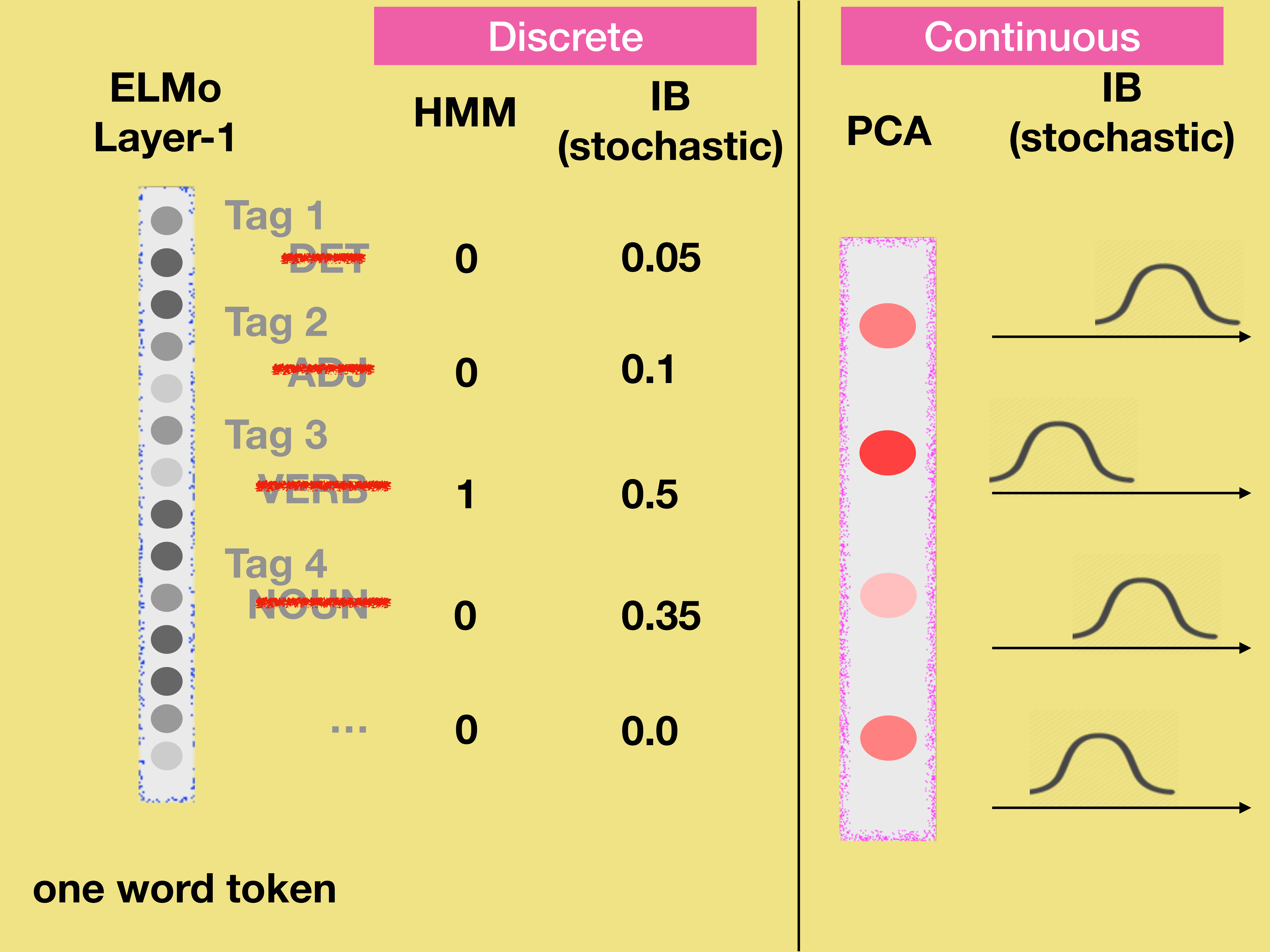
part-of-speech taggings

ELMo Embeddings



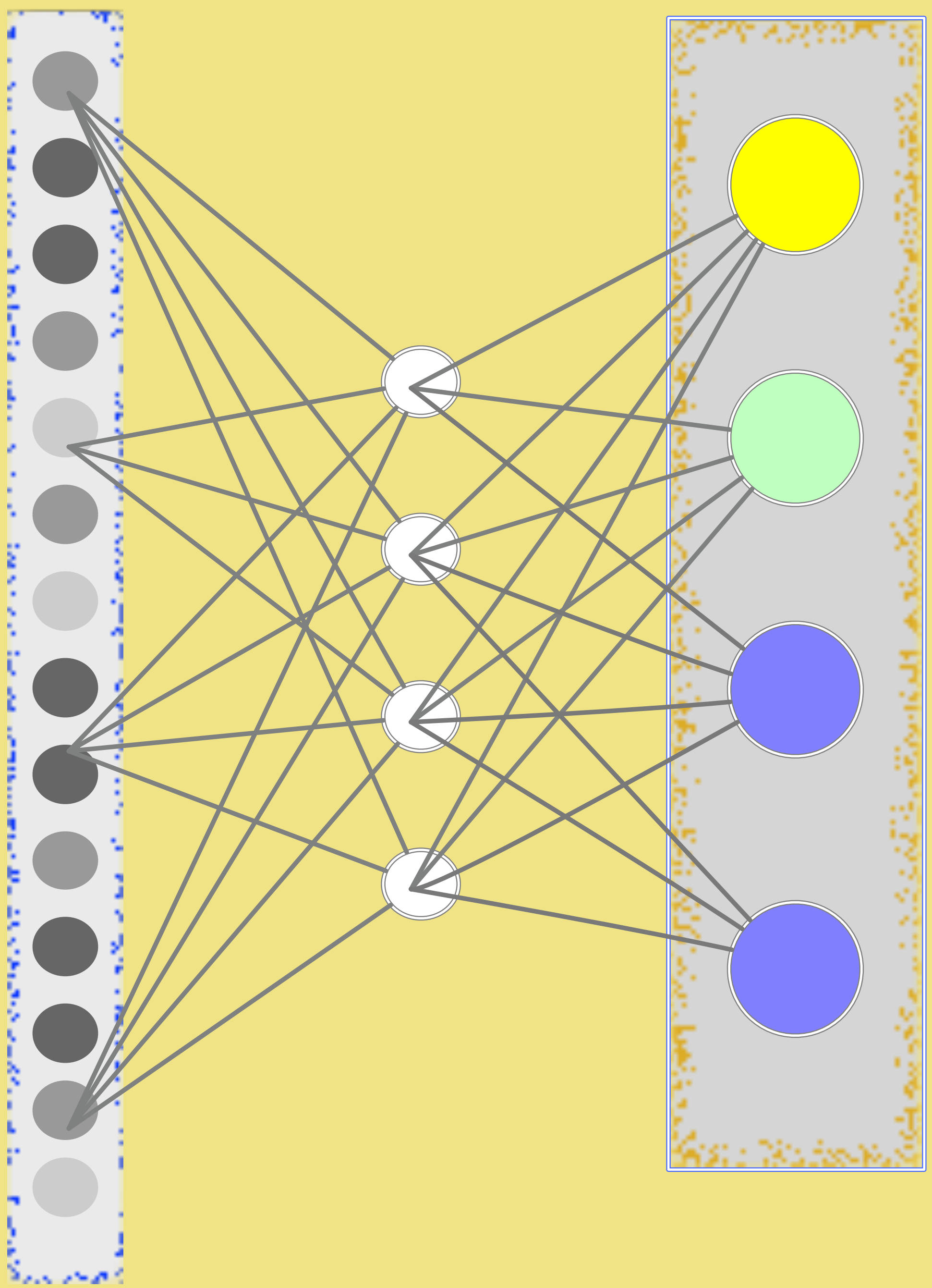
part-of-speech taggings

CCG taggings


Discrete
Continuous
ELMo
Layer-1
HMM
IB
(stochastic)
PCA
IB
(stochastic)
Tag 1
~~DET~~
0
0.05
Tag 2
~~ADJ~~
0
0.1
Tag 3
~~VERB~~
1
0.5
Tag 4
~~NOUN~~
0
0.35
...
0
0.0
one word token

ELMo
Layer-1

IB
(stochastic)



Softmax

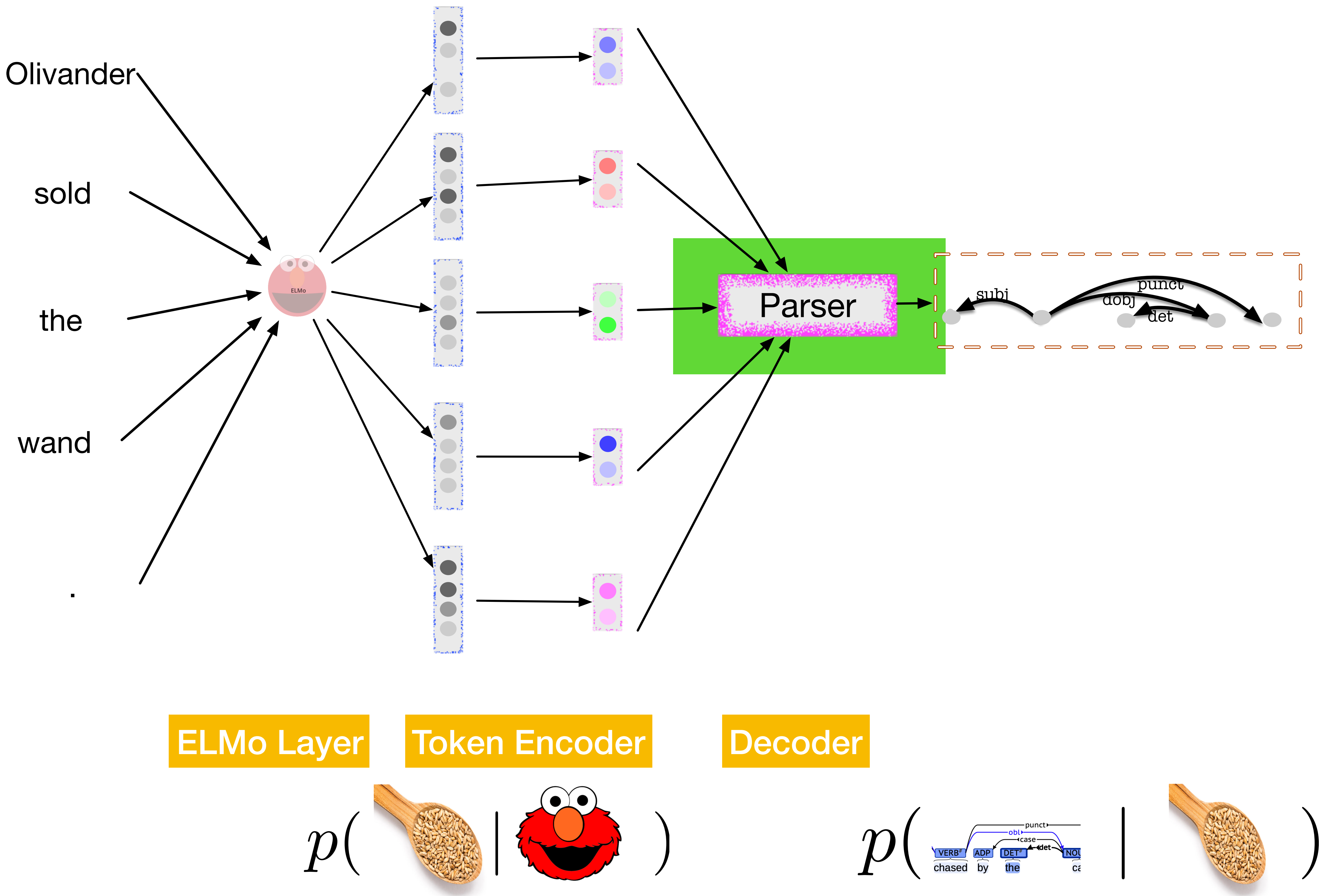
Tag 1 0.05

Tag 2 0.1

Tag 3 0.7

Tag 4 0.15

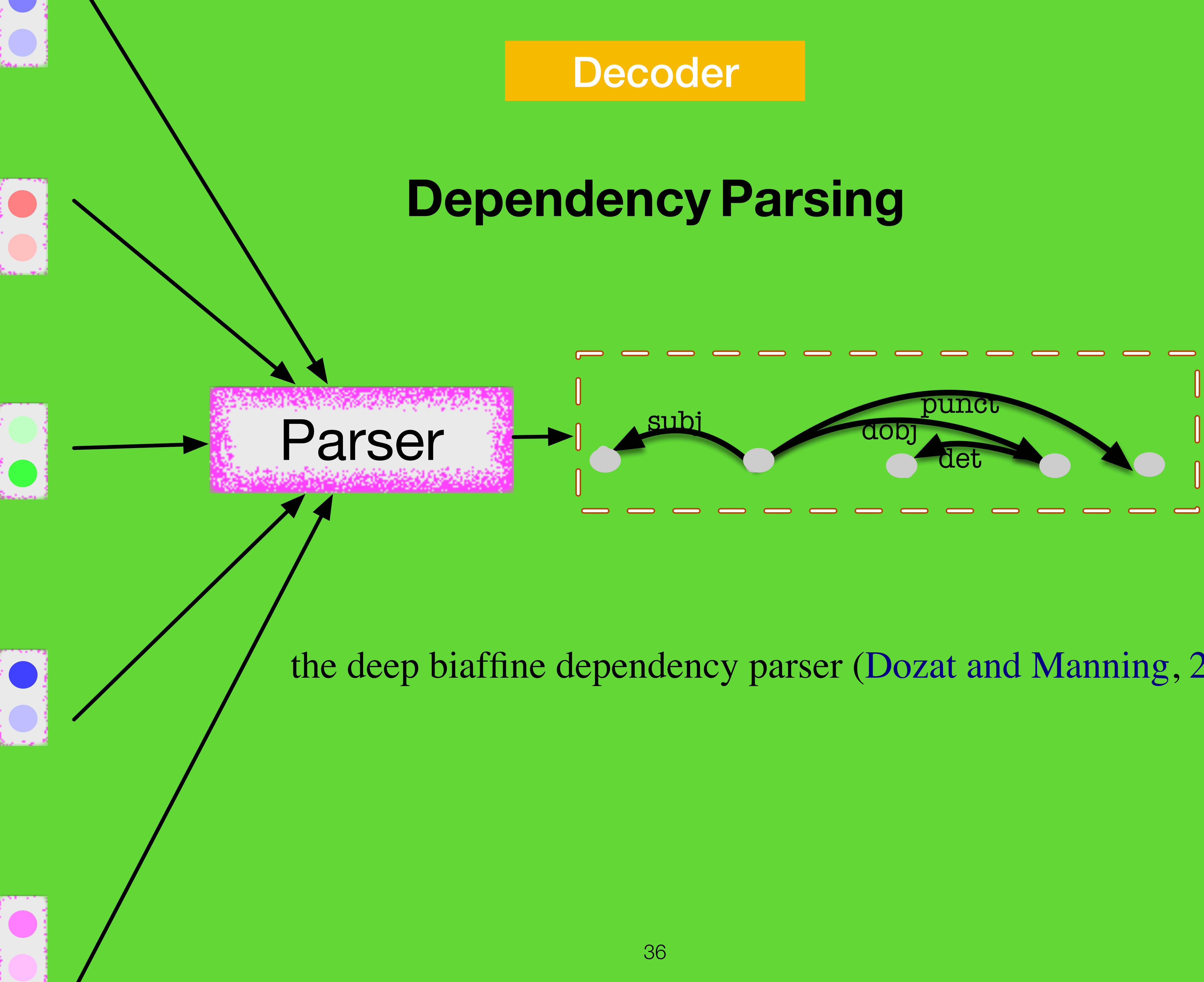
one word token



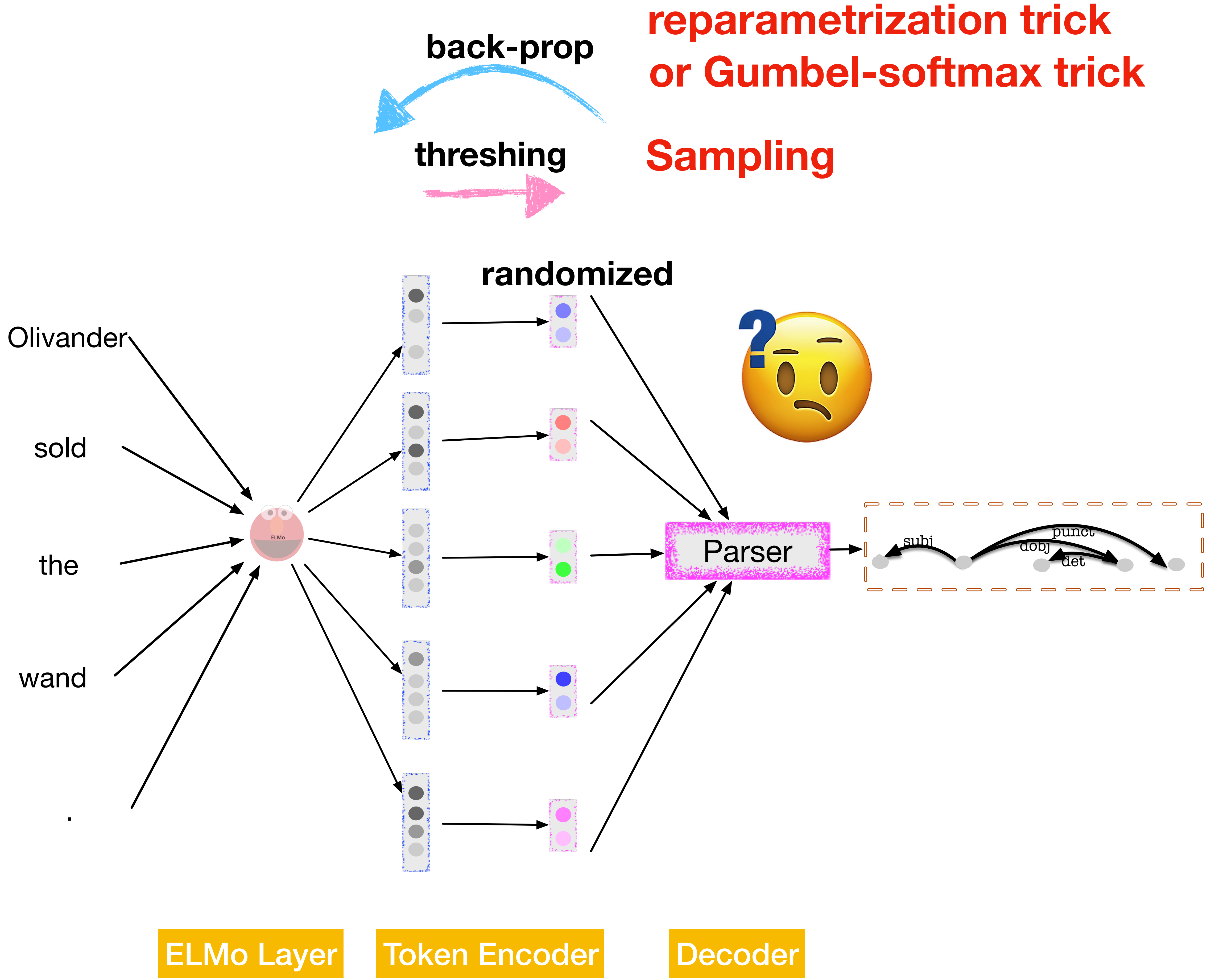
Olivander
sold
the
wand
.
Parser
subj
punct
dobj
det
ELMo Layer
Token Encoder
Decoder
p(
|
)
p(
chased
by
the
|
)

Decoder

Dependency Parsing

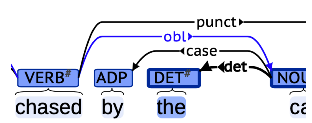


the deep biaffine dependency parser (Dozat and Manning, 2016)



reparametrization trick
or Gumbel-softmax trick
back-prop
threshing
Sampling
randomized
Olivander
sold
the
wand
.
Parser
subj
punct
dobj
det
ELMo Layer
Token Encoder
Decoder

Variational Upper Bound

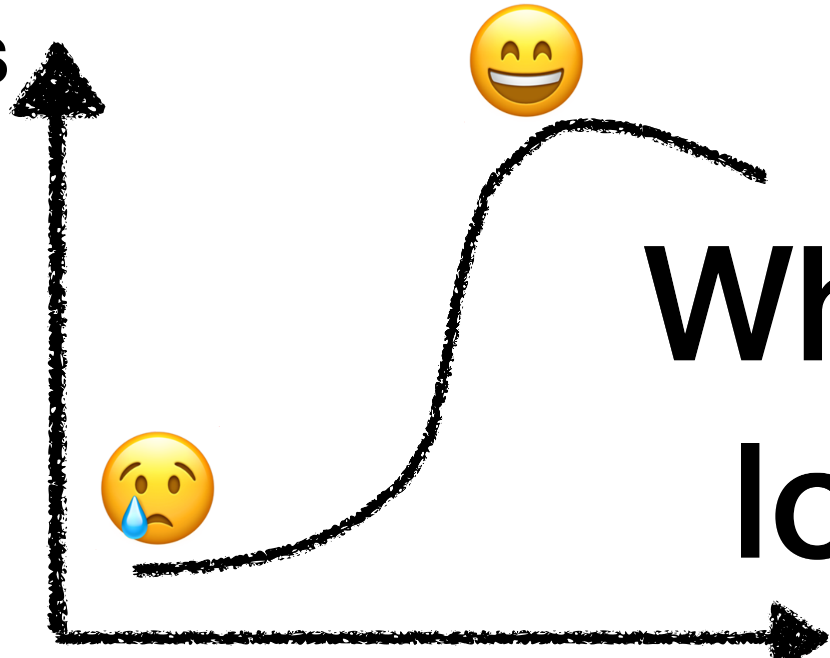
- To minimize the IB objective directly, some terms are intractable.
- $p($ 

$|$  $)$ **and** $p($  $)$
- So we instead minimize a variational upper bound.
  - Which is what the previous slide estimated by sampling
- For mathematical details, please refer to our paper ;)

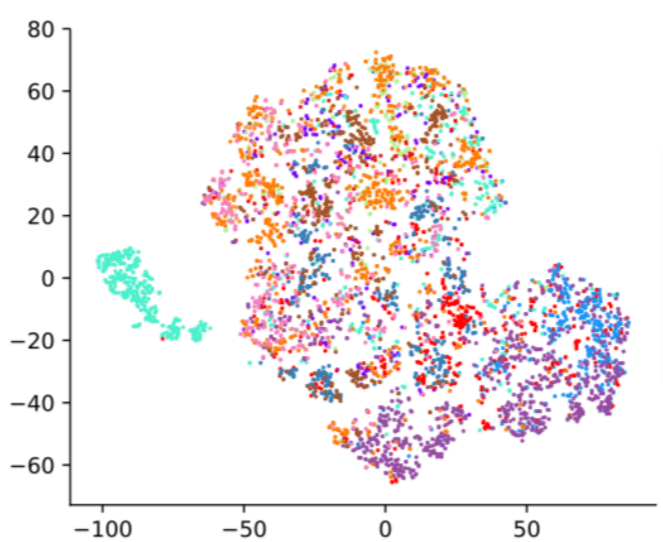
Do our specialized tags correlate with traditional POS tags?

Yes!

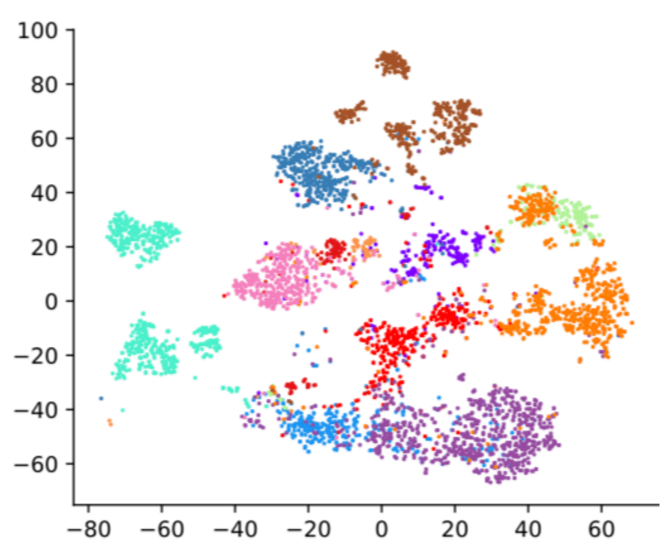
Results



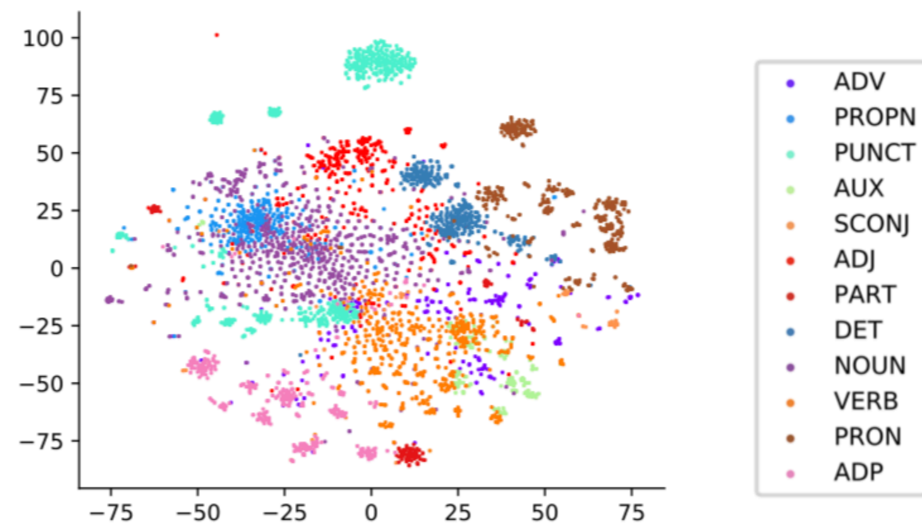
What do specialized tags look like? [Continuous]



(c) $\mathrm{I}(X;T) \approx 0.069$



(b) $\mathrm{I}(X;T) \approx 24.3$



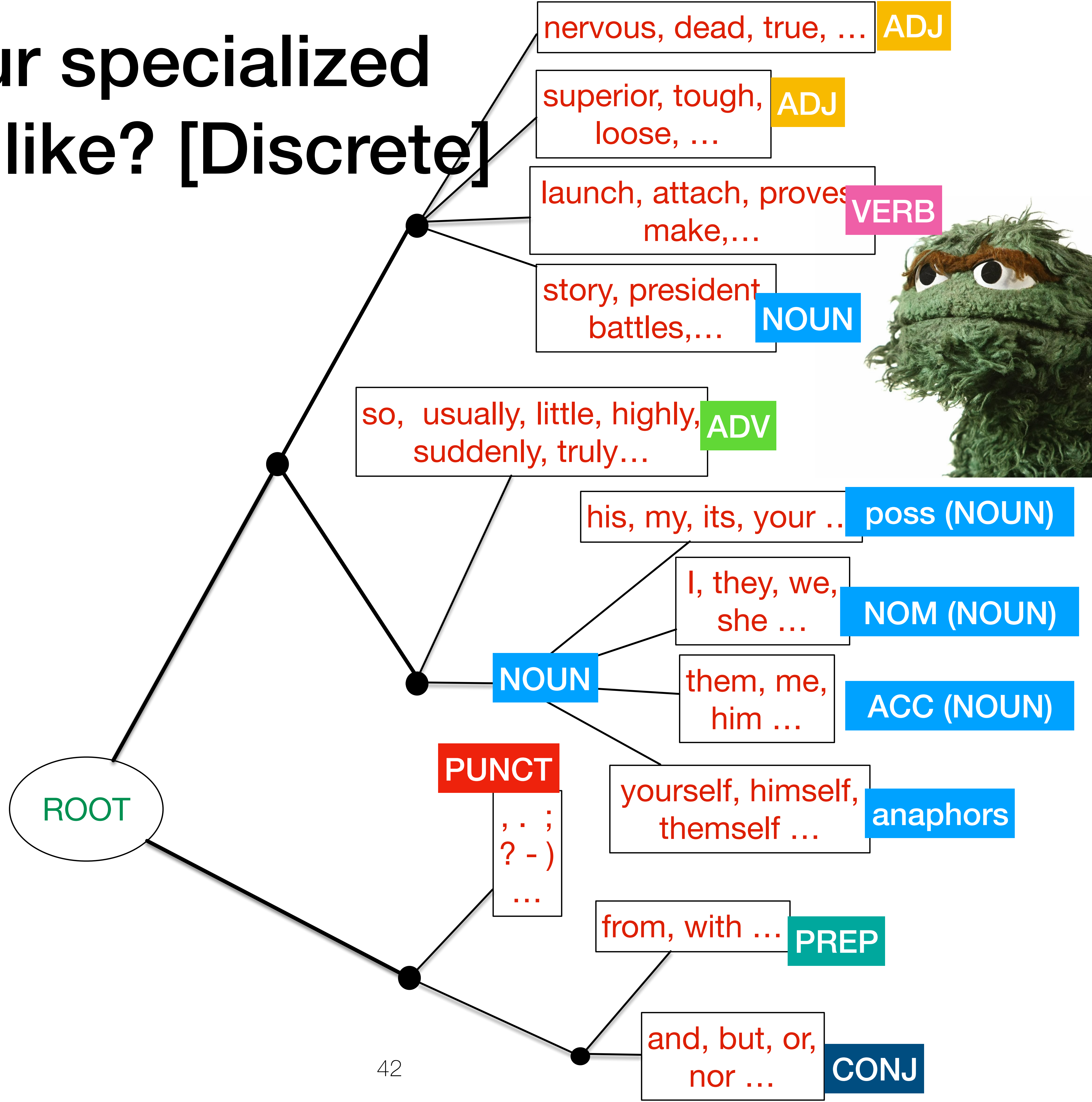
(a) ELMo, $\mathrm{I}(X;T) = H(X) \approx 400.6$

Too much Compression

Moderate Compression

All Information Kept

What our specialized tags look like? [Discrete]



Do our specialized tags correlate with POS tags?

Yes!

- Our results agree with the intuition that POS keeps information about parsing.
- POS is contextual. Our taggings are contextual as well !!!

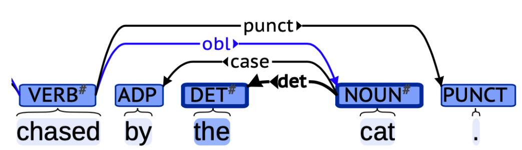
WAIT!? The word type (without context) should also be a strong predictor of POS.

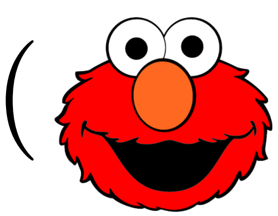
How do we specialize so that we mostly depend on type information?

1. We could use ELMo Layer-0.

ELMo layer-0 is based on a character-level convolutional network. Thus, it inherently does not contain contextual information. Specializing is to extract from existing embeddings, which is also non-contextual.

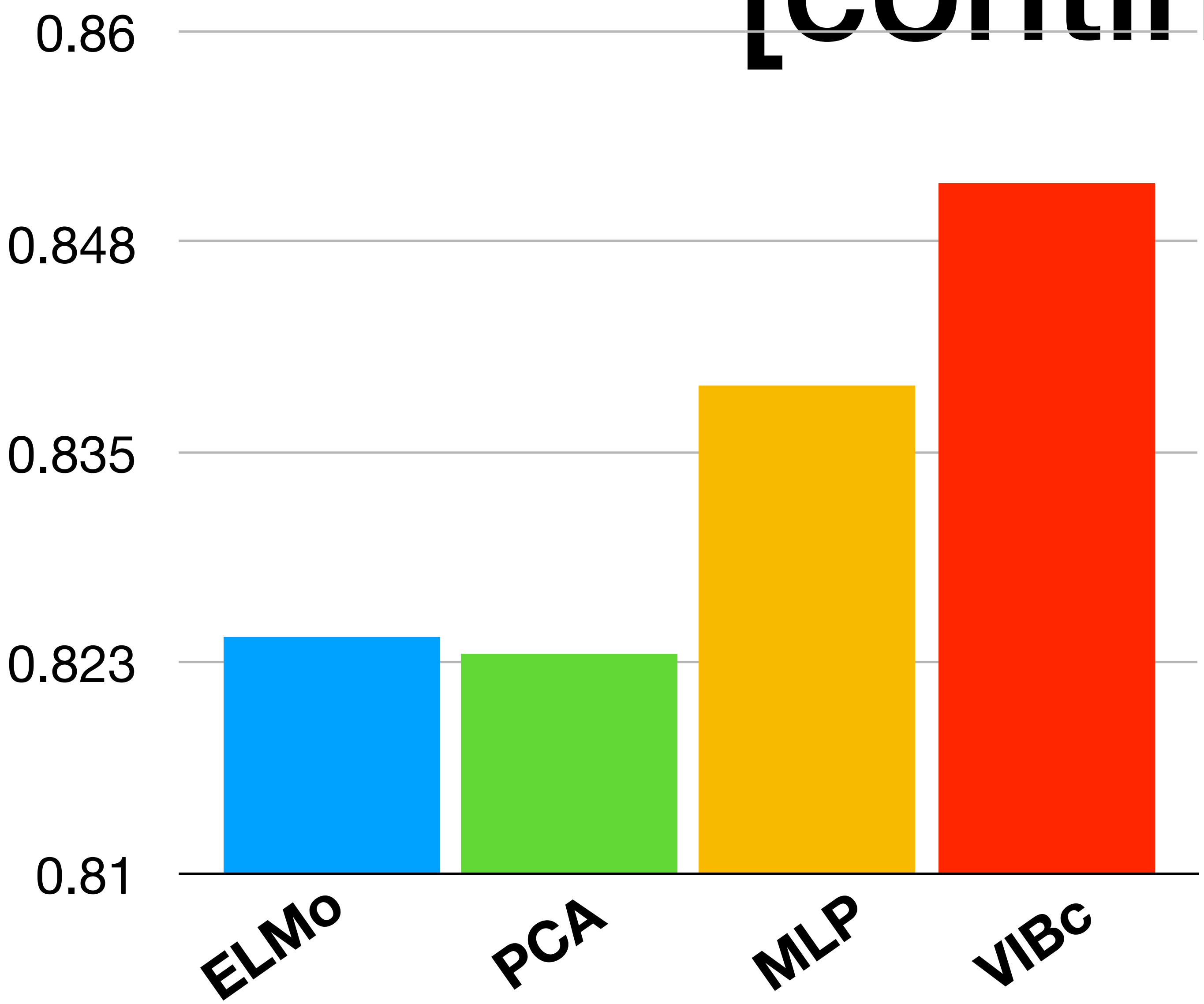
2. We could do a softer version — make the specialized tagging depend “mostly” on its word type.

$\mathcal{L} = -\mathrm{I}($ 

$;$  $) + \beta \cdot \mathrm{I}($  $;$  $)$

$+\gamma \cdot \sum_{\text{word } i \text{ in sentence}} \mathrm{I}($  $_i \; ;$  $_i \mid$  $_{\text{layer-0}, i})$

Parsing Performance [continuous]



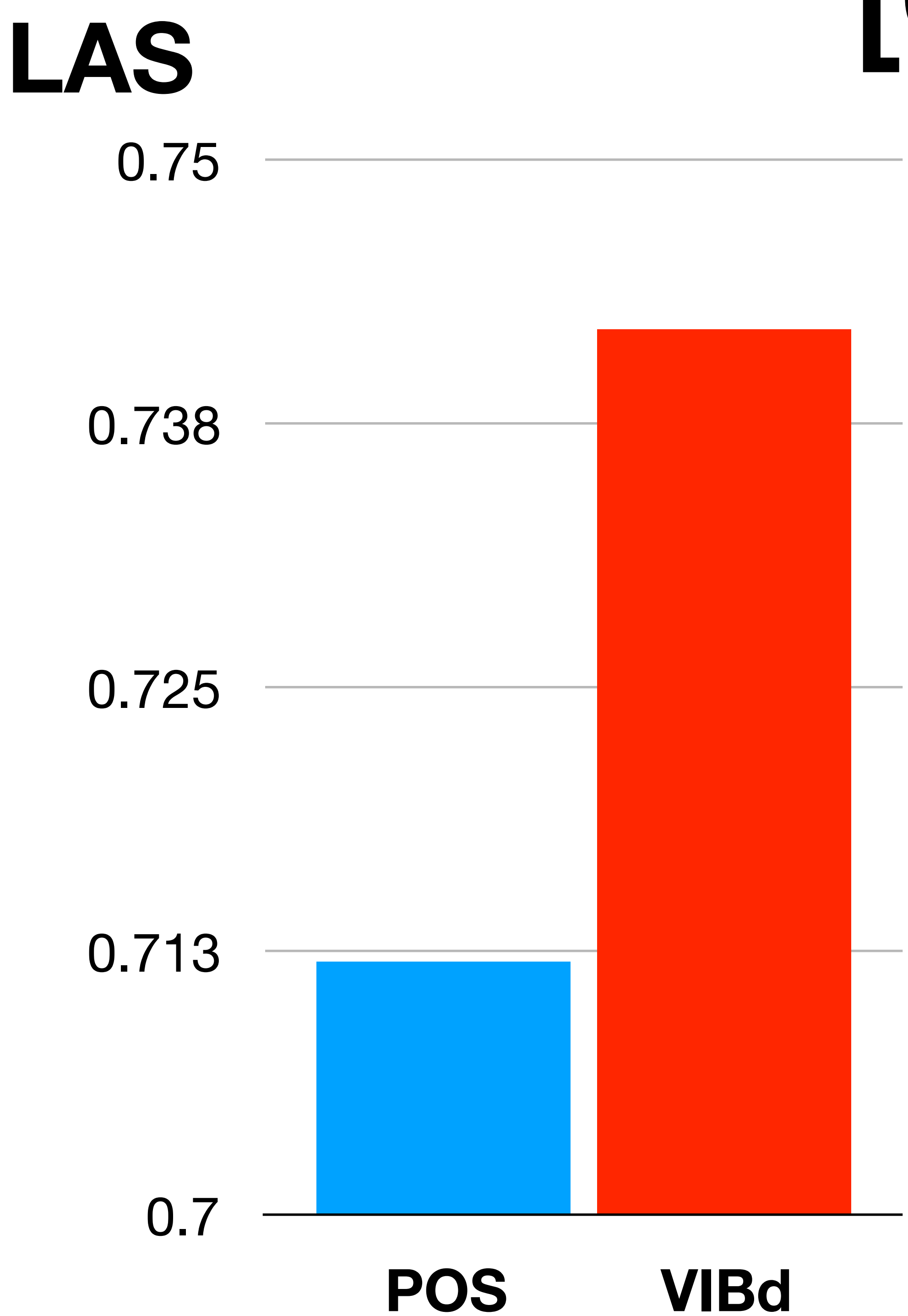
~~English~~

Wins or ties across 9 languages.

$\approx$ 1.0 point improvement on average

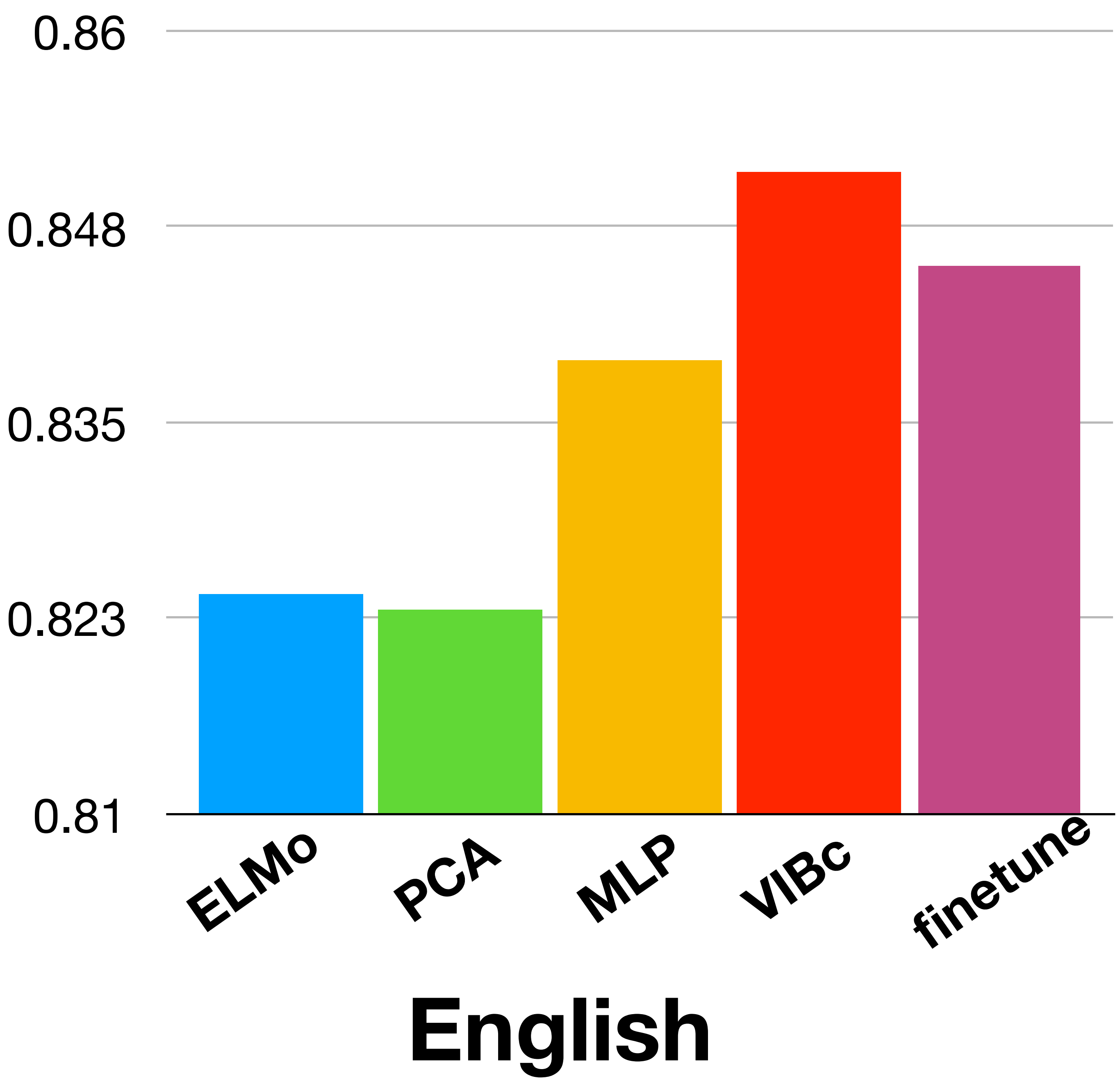
- **ELMo :** train by 1024 dimensional ELMo-layer2
- **PCA :** train by 256 dimensional embeddings after applying PCA to ELMo-layer2
- **MLP:** train a non-linear cleanup layer after the ELMo embeddings jointly with the parser.
- **VIBc:** our method ;)

Parsing Performance [discrete]



- POS: train the parser on gold POS tags
- VIBd: Our discrete version.

Parsing Performance [continuous]



- **ELMo :** train by 1024 dimensional ELMo-layer2
- **PCA :** train by 256 dimensional embeddings after applying PCA to ELMo-layer2
- **MLP:** train a non-linear cleanup layer after the ELMo embeddings jointly with the parser.
- **VIBc:** our method ;)
- **finetune + mild hyperparameter tuning**

Thanks