Fortuitous data

ESSLLI 2016, Day 3

https://fortuitousdata.github.io/

Representations and learning from related tasks

Today

1. Neural Networks: Graph view
2. Representations
3. Multi-task learning
4. Fortuitous NLP
    - Part-of-speech tagging with bi-LSTMs and auxiliary loss
    - Keystroke dynamics as source for syntactic chunking

Neural Networks - Graph view

Feed-forward Neural Network

$$y = NN(\mathbf{x})$$

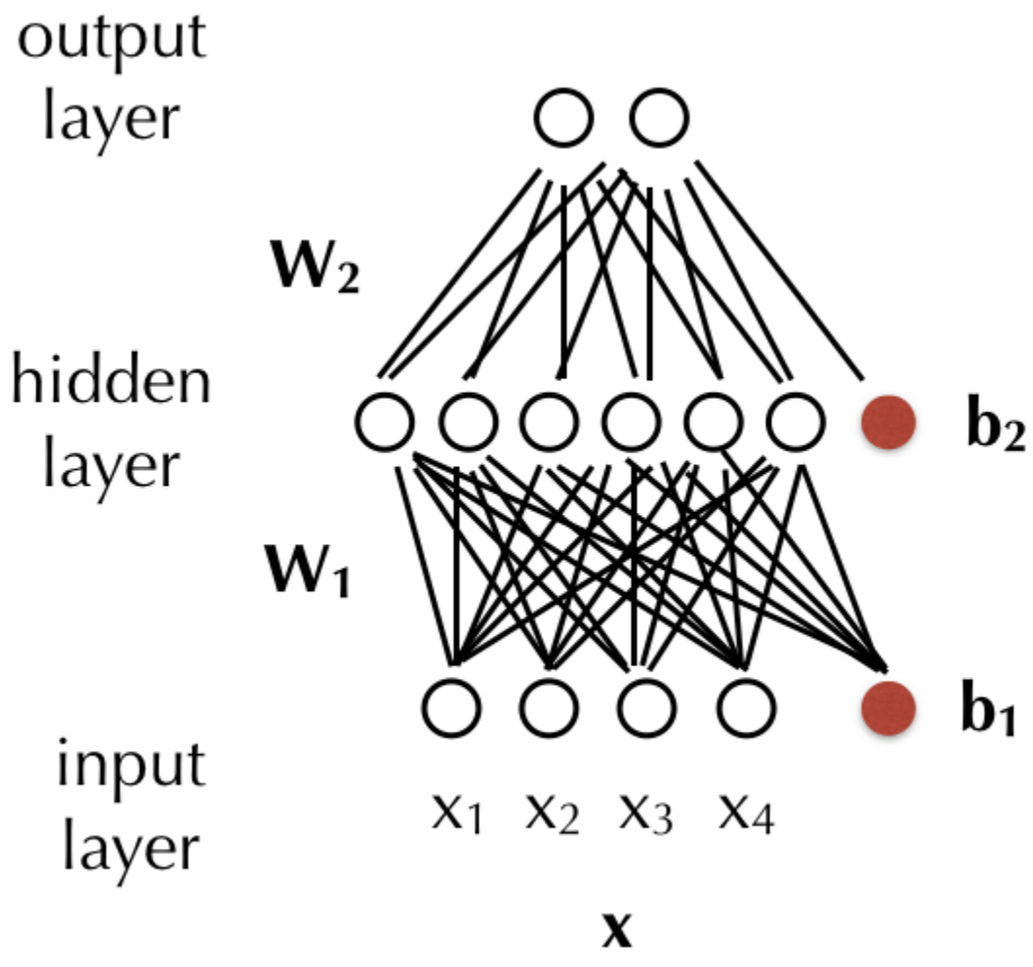
input: $\mathbf{x}$ (vector with d_{in} dimensions)

output: $\mathbf{y}$ (output with d_{out} classes)

Example

Formalization and corresponding visualization:

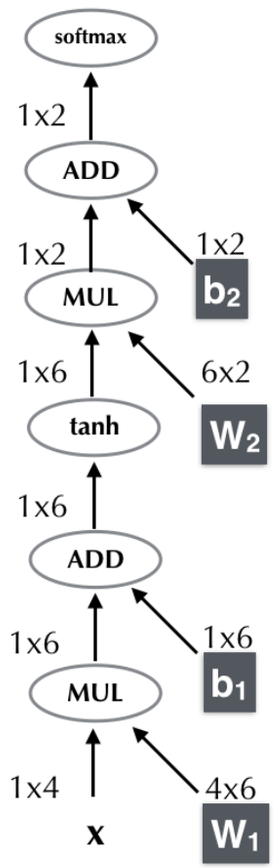
$$NN_{MLP1}(\mathbf{x}) = \sigma(g(\mathbf{x}\mathbf{W}^{\mathbf{1}} + \mathbf{b}^{\mathbf{1}})\mathbf{W}^{\mathbf{2}} + \mathbf{b}^{\mathbf{2}})$$



Computational graph:

- nodes are operations
- gray boxes are parameters

$$NN_{MLP1}(\mathbf{x}) = g(\mathbf{x}\mathbf{W}^{\mathbf{1}} + \mathbf{b}^{\mathbf{1}})\mathbf{W}^{\mathbf{2}} + \mathbf{b}^{\mathbf{2}}$$



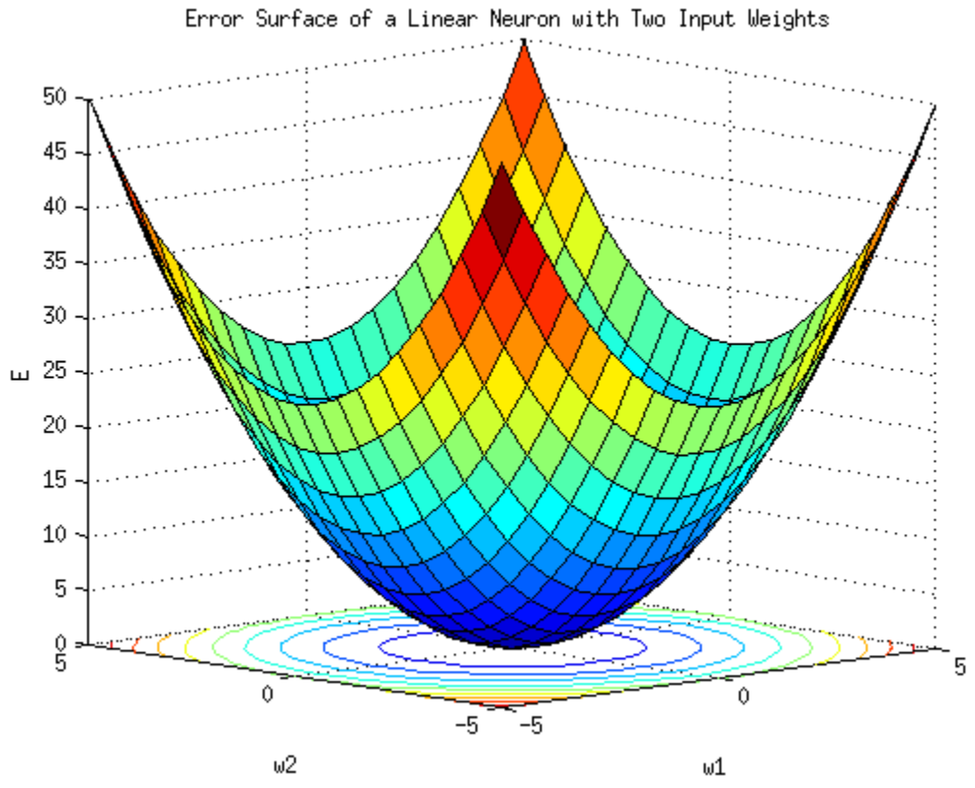
Where do the weights come frome?

It's an **optimization** problem.

We need:

1. a loss function $l(\tilde{y}, y)$
2. a way to change the model (parameters) to get closer to a good model (hint: SGD)

Minimize loss using gradient-based method



Skeleton of gradient descent:

Input: training set, loss function l

Repeat for number of iterations (**epochs**):

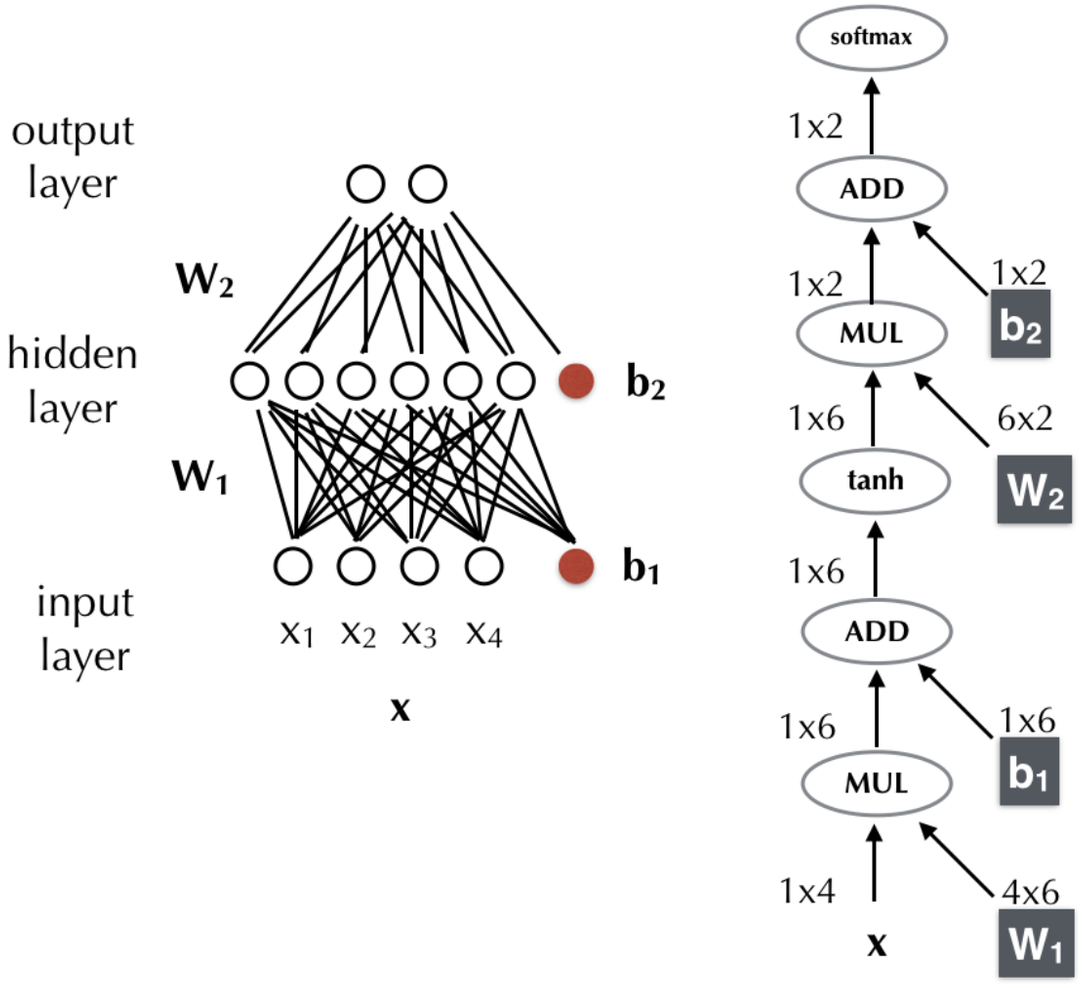
- compute loss on data: $l(X, Y)$
- compute gradients: $\mathbf{g} = \frac{\partial}{\partial \theta} l(X, Y)$ with respect to θ
- move parameters in direction of the negative gradient: $\theta \pm -\eta \mathbf{g}$

Backprop

- backward pass: gradient computations (through chain rule)

Summary: Equivalent formulizations

$$NN_{MLP1}(\mathbf{x}) = g(\mathbf{x}\mathbf{W}^{\mathbf{1}} + \mathbf{b}^{\mathbf{1}})\mathbf{W}^{\mathbf{2}} + \mathbf{b}^{\mathbf{2}}$$



Complete Neural Network

Loss

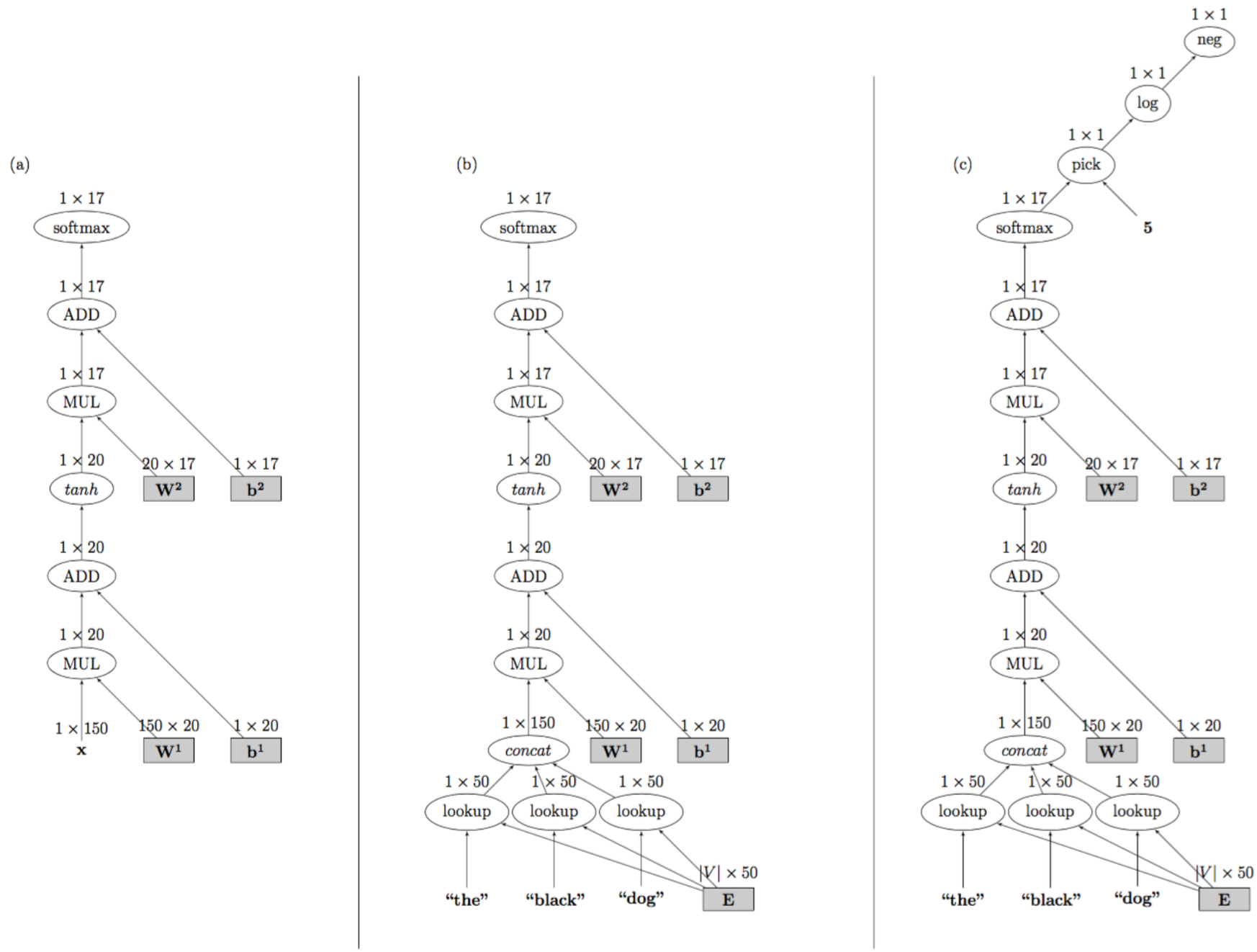


Figure 3: **Computation Graph for MLP1.** (a) Graph with unbound input. (b) Graph with concrete input. (c) Graph with concrete input, expected output, and loss node.

However, what is the input $\mathbf{x}$?

Representations

Feature representations

*Probably the biggest jump when moving from traditional linear models with sparse inputs to deep neural networks is to stop representing each feature as a unique dimension, but instead represent them as **dense vectors** (Goldberg 2015).*

discrete representation

$$\mathbf{x}_{cat} = [0, 0, 0, 0, 0, 0, 1]$$
$$\mathbf{x}_{dog} = [0, 0, 0, 0, 1, 0, 0]$$

similarity on discrete representations?

$$sim(\mathbf{x}_{cat}, \mathbf{x}_{dog}) = 0$$

Word embeddings

"You shall know a word by the company it keeps" (Firth, J. R. 1957:11)

i operetten Flagermusen. Partiet som elegantier og **flødebolle** løste Kristian Boland meget overbevisende, og nye alen lagde han

flødeboller, er væddemålet jo heller ikke vundet, når **flødebolle** nr. ni er fortæret. I denne tid hører og læser

hjemmelavet fedtfattigt kage. Chokolade udskiftes med en **flødebolle.** Kiks, grove eller med fyld udskiftes med riskiks, knækbrød eller

dem og på bollerne fra Netto, eller på den ene **flødebolle,** jeg købte i chokoladeforretningen, og som jeg lagde 2,50 kr.

nougat og jordbærsyltetøj og vaffel og chokoladedrys og **flødebolle** og alt. "Øj ", hviskede Bosse. "Forstår I drenge ", sagde min

lov til at synde lidt- til at spise en ugentlig **flødebolle,'** siger Henrik Byager. Når veluddannede og andre atypiske ugebladslæsere nu

med 4,75 kr. og købe chokoladeforretningens hjemmelavede **flødebolle,** den er der gods i og tommetyk chokolade udenom- til

i Købmagergade. Det ene kendt som " den omvendte **flødebolle "** p.g.a. det sære, lyddæmpende loft. Her huserede Kgl. Kammersanger Emil

pengene mere en hensigtserklæring, der rækker som én **flødebolle** i en børnehaveklasse. I stedet for at vente på regeringens

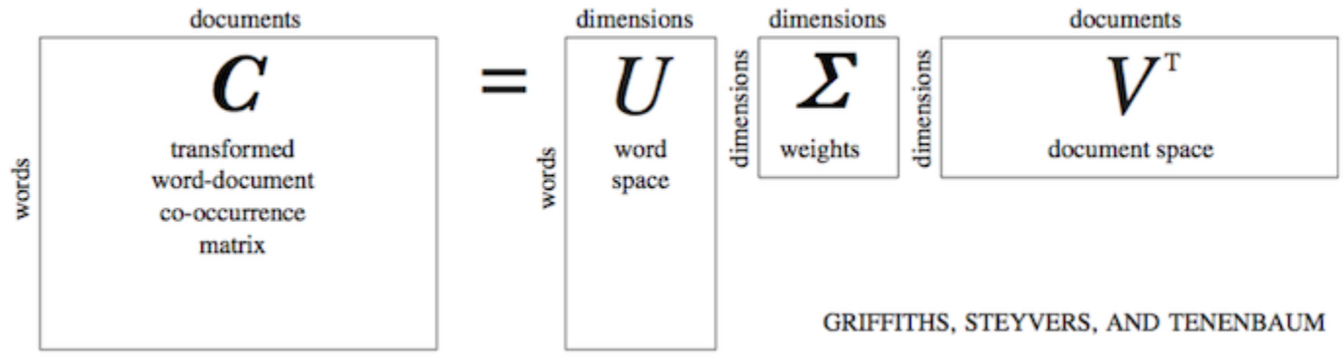
Approaches

1. Traditional approach: LSA (SVD) on word-coocurrence matrix
2. word2vec

LSA - Latent Semantic Analysis

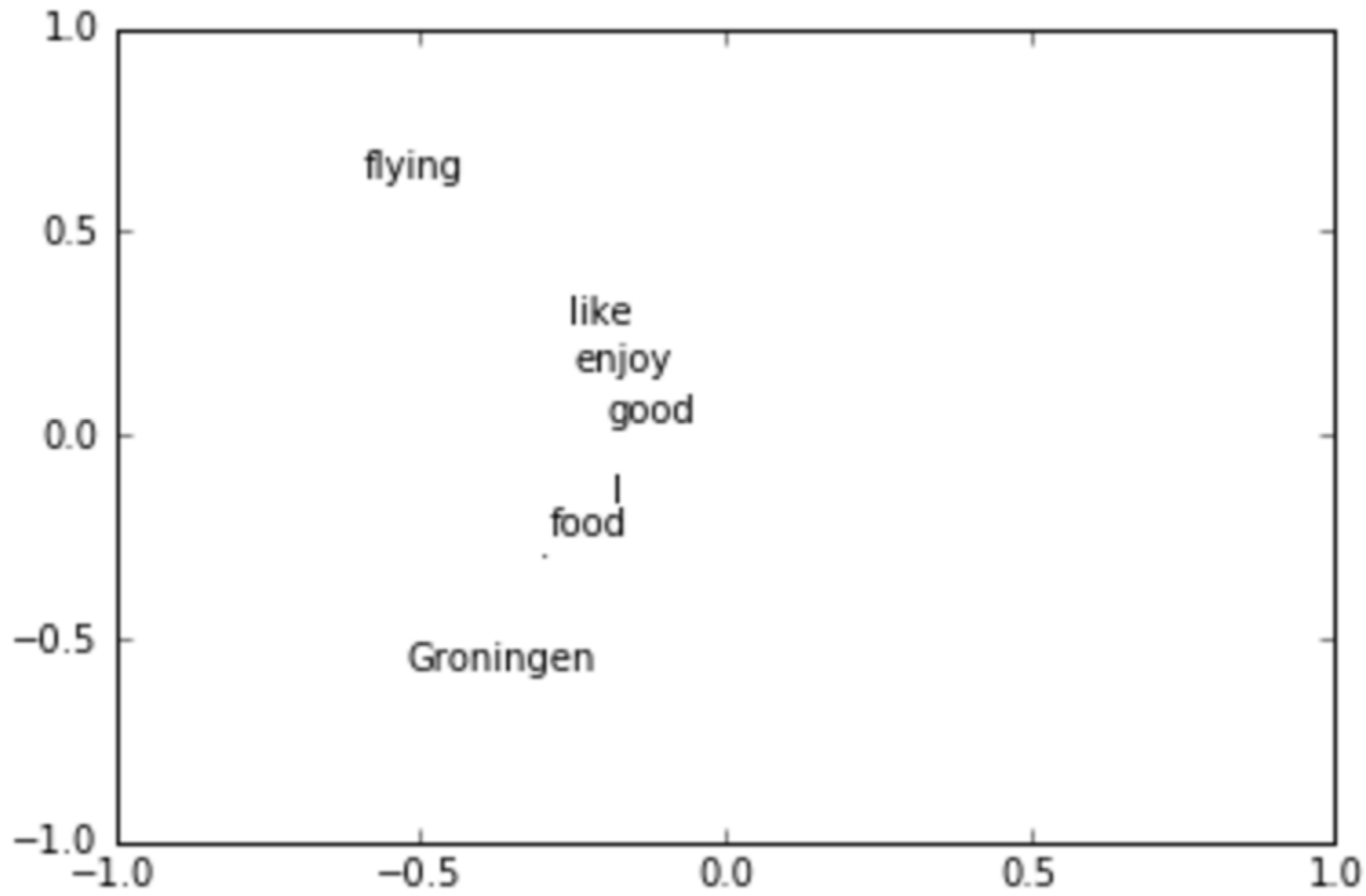
Approximate a matrix $\mathbf{C}$ through a decomposition into three submatrices (**of smaller dimensionality**) - Singular Value Decomposition (SVD):

$$\mathbf{C} \approx \mathbf{U} \sum \mathbf{V}^{\mathbf{T}}$$



by Simon Paarlberg

NB. = should be ≈

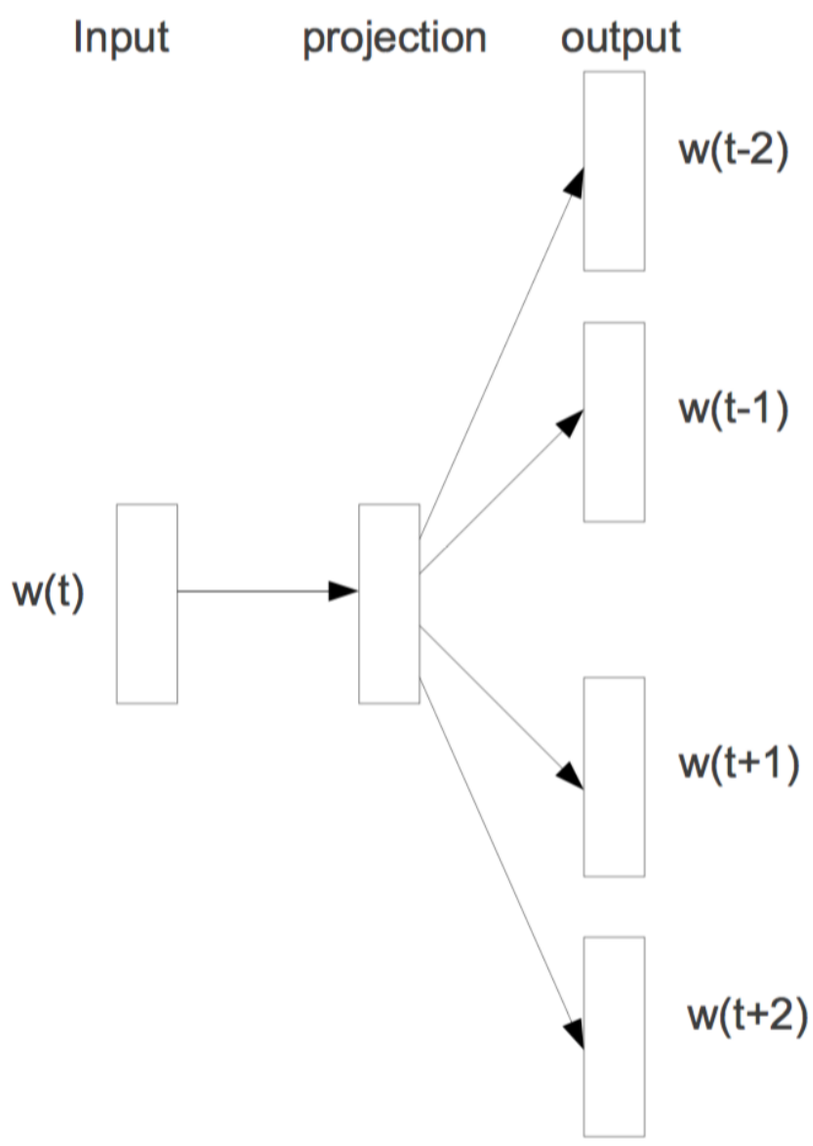

1.0
0.5
0.0
−0.5
−1.0
−1.0
−0.5
0.0
0.5
1.0
flying
like
enjoy
good
food
Groningen

word2vec

Main idea:

- instead of capturing co-occurence statistics of words
- **predict context** (surrounding words of every word); in particular, predict words in a window of length m around current word

since SVD computation cost scales quadratically with size of co-occurence matrix

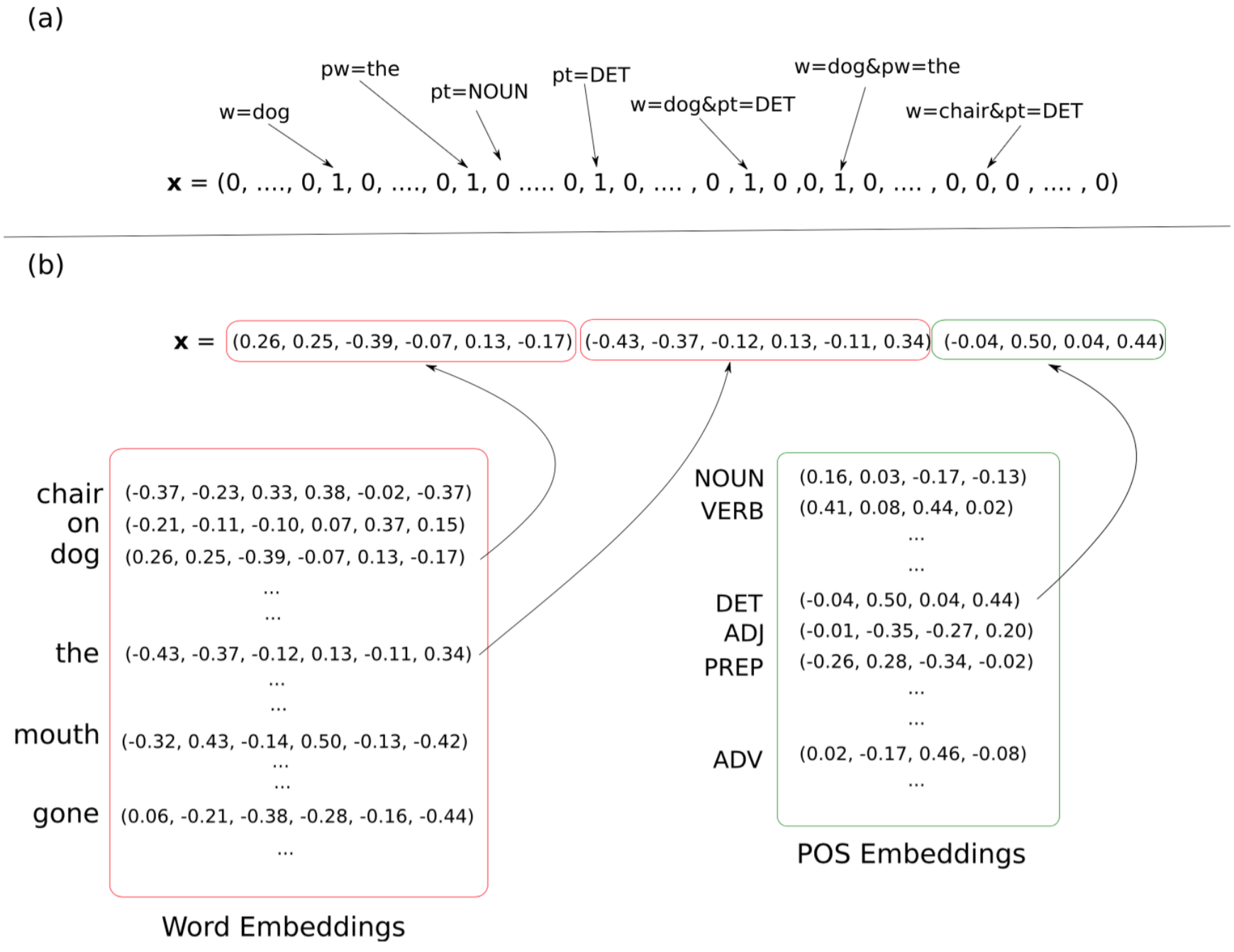


Mikolov et al. (2013)

NB. denominator $\sum$ over all words! → *negative sampling* or *hierarchical softmax*

Note: embeddings are not specific to **words**

Sparse vs dense



Goldberg (2015)

Dense feature spaces

A common choice for c is **concatenation**:

$$\mathbf{x} = c(f_1, f_2, f_3) = [v(f_1); v(f_2); v(f_3)]$$

Other representations:

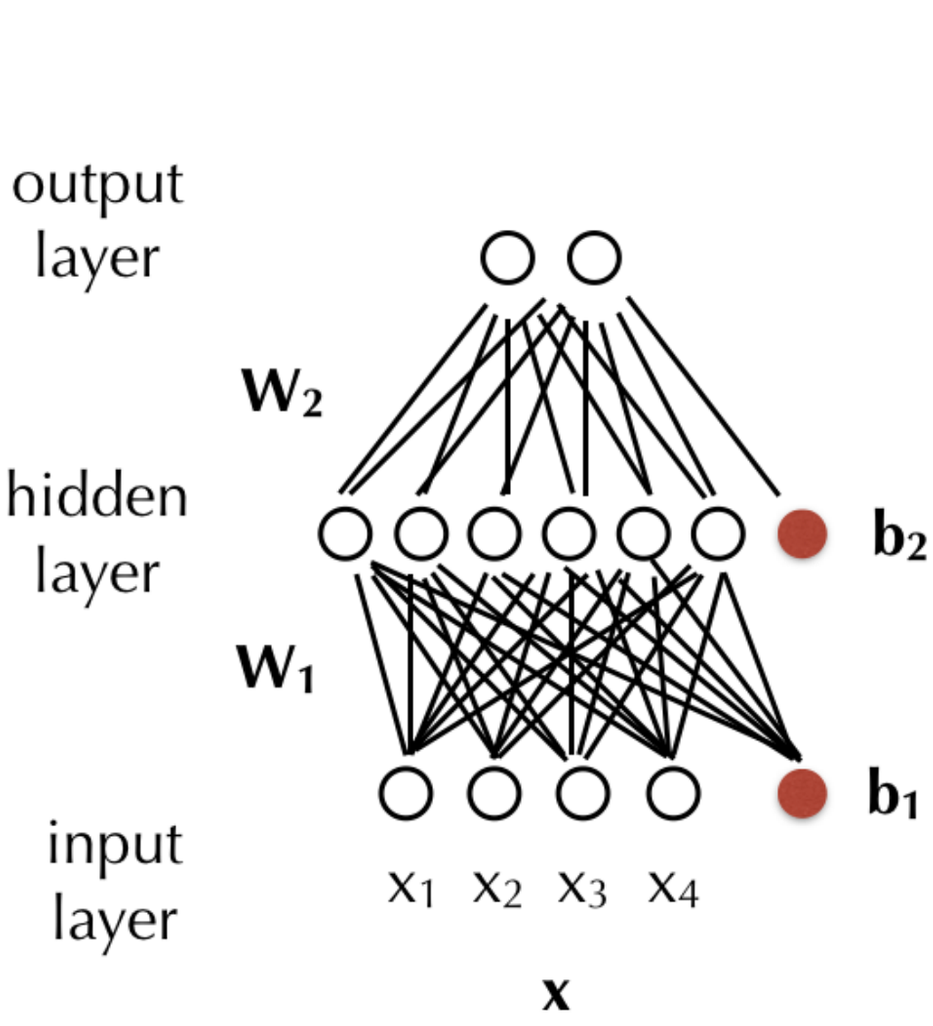
sum :

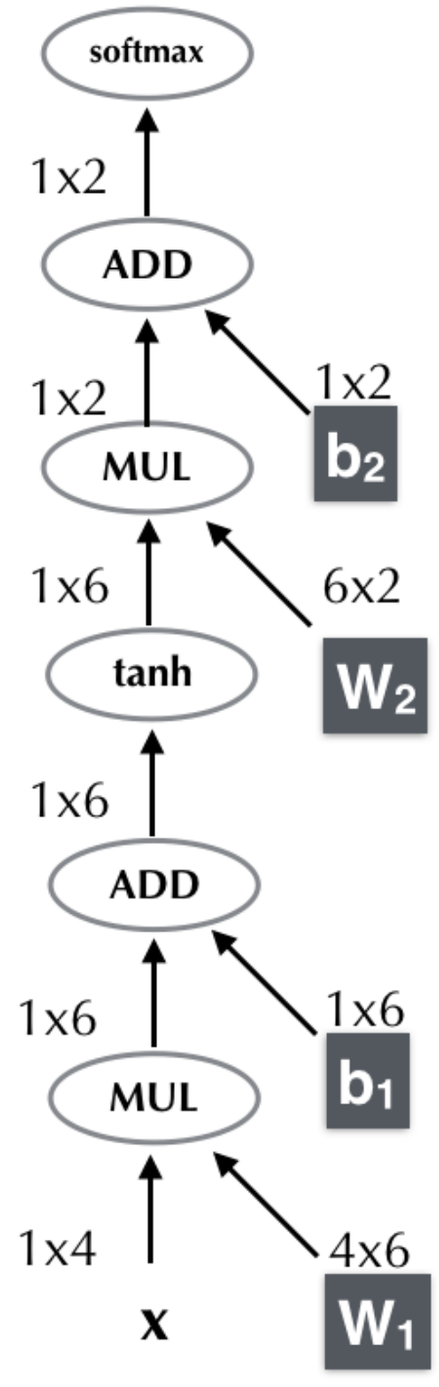
$$\mathbf{x} = c(f_1, f_2, f3) = [v(f_1) + v(f_2) + v(f_3)]$$

mean:

$$\mathbf{x} = c(f_1, f_2, f3) = [mean(v(f_1), v(f_2), v(f_3))]$$

$$NN_{MLP1}(\mathbf{x}) = g(\mathbf{x}\mathbf{W}^{\mathbf{1}} + \mathbf{b}^{\mathbf{1}})\mathbf{W}^{\mathbf{2}} + \mathbf{b}^{\mathbf{2}}$$





x

sparse, dense or both

Multi-task learning

Key idea

The idea of **multi-task learning** (Caruana 1998, Collobert et al. (2011)) to exploit the training signal of **other tasks**.

Example: Card game

- you are in beautiful Italy and want to get acquainted with local card games. You hear about 'scala 40', and are eager to learn it
- The input space are cards, and the output space are configuations (hands) of your cards.
- You know already how to play poker. Rather than starting from scratch (**tabula rasa**), you use your internal knowledge of poker (or generally how to play a card game) to learn how to play 'scala 40'.

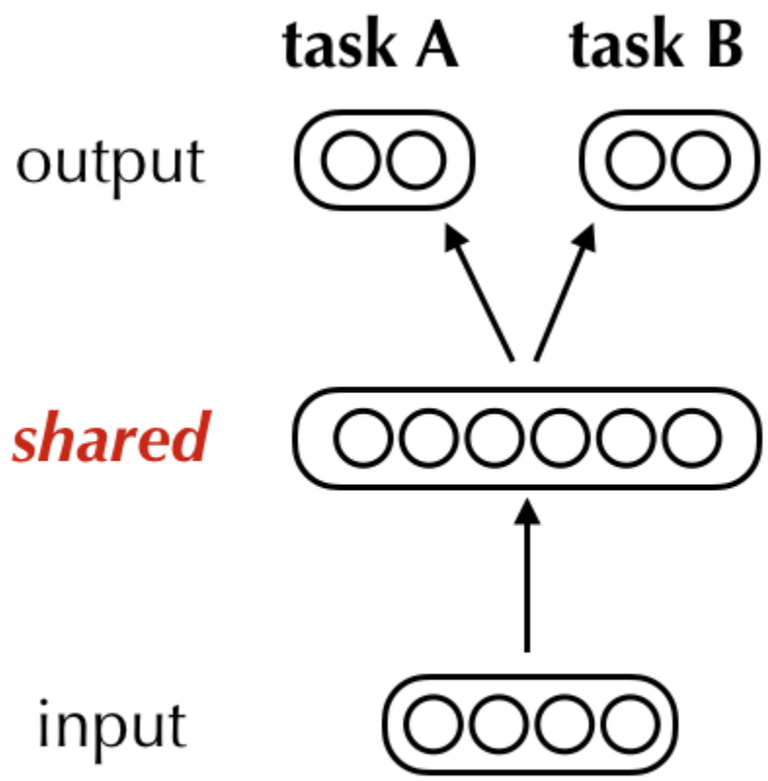
Embeddings as fortuitous data in Transfer learning

- want: model that works better on other variety of data
- pool of unlabeled data, estimate embeddings (word2vec)

Why would using embeddings work?

- embeddings provide latent space $\mathcal{Z}$, **side benefit** of optimising another objective (language model)
- add to feature space $\phi(x)$, latent space $\mathcal{Z}$
  - add to one-hot vector
  - initialize embeddings in dense

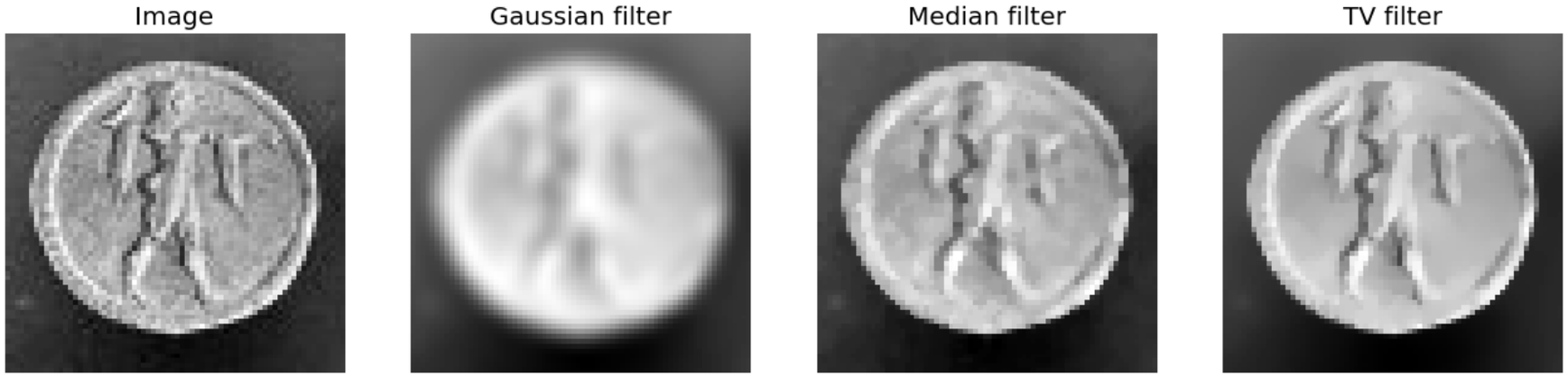
Multi-task learning



Why does MTL work?

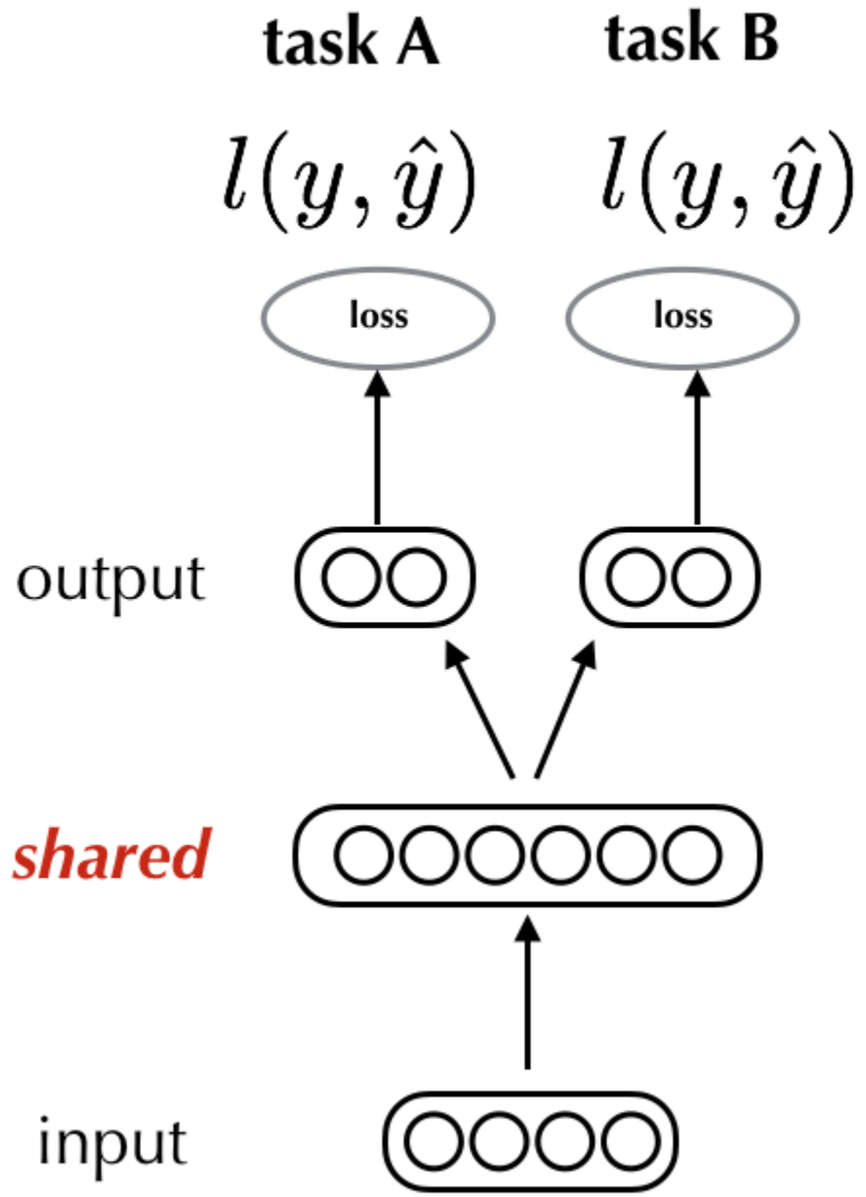
Reduced capacity (Caruana 1998)

Eavesdropping



scikit - denoising filters

Noise in extra outputs might be less harmful than in extra input (Caruana 1998) (also: **weighting** of loss)



Joint training with:

1. jointly labeled data, but also
2. distinct sources (!) (for NLP first noted in Collobert and Weston 2008)

Deep Joint Training

(Collobert and Weston 2008)

1. Select the next task.
2. Select a random training example for this task.
3. Update the NN for this task by taking a gradient step with respect to this example.
4. Go to 1.

Successful MTL

The first self-driving car

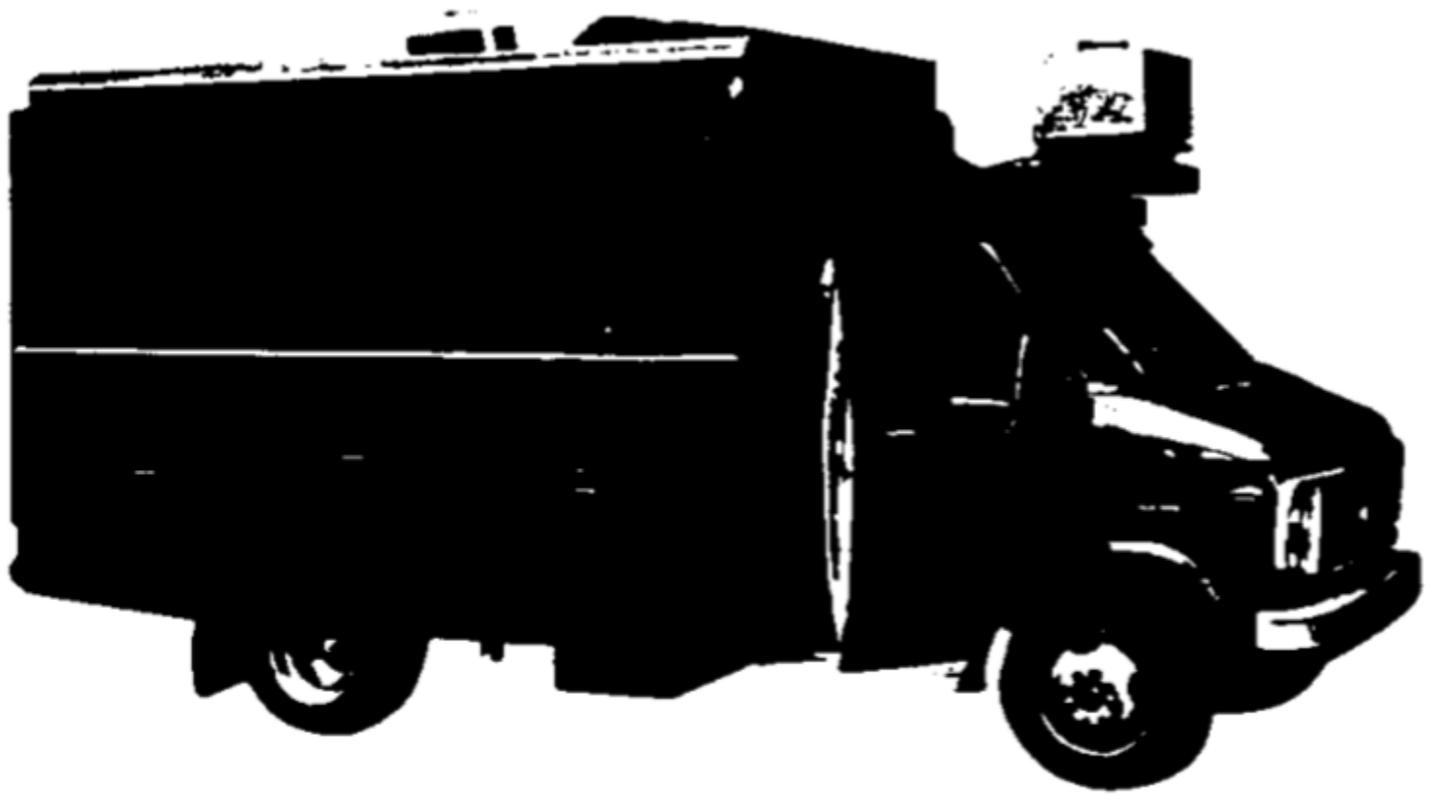


Figure 4: NAVLAB, the CMU autonomous navigation test vehicle.

CMU Alvinn MTL (Caruana 1998)

For our MTL experiments, eight additional tasks were used:

- whether the road is one or two lanes
- location of left edge of road
- location of road center
- intensity of region bordering road
- location of centerline (2-lane roads only)
- location of right edge of road
- intensity of road surface
- intensity of centerline (2-lane roads only)

CMU Alvinn MTL (Caruana 1998)

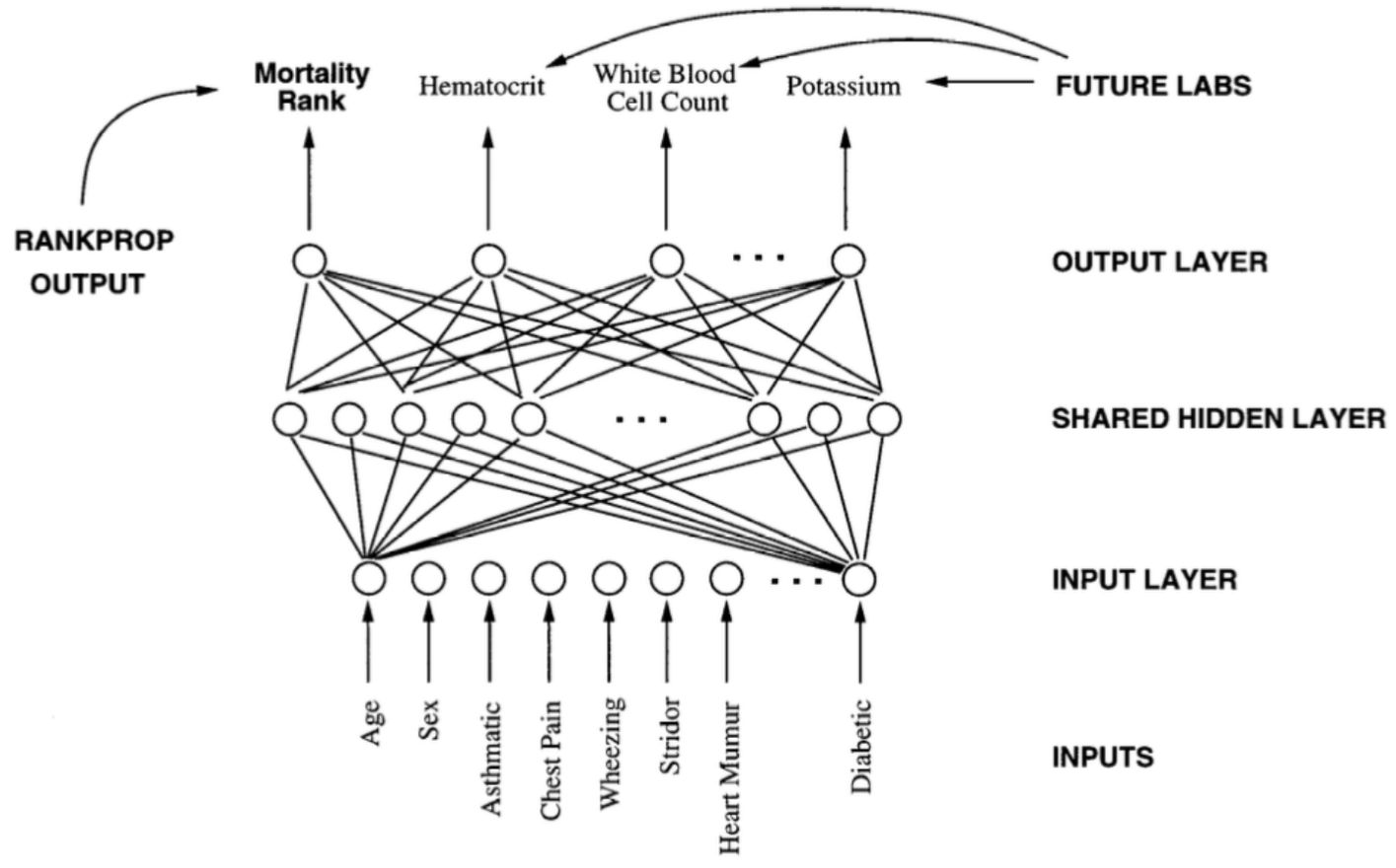
Note: here all task labels computable from data

Table 1. Performance of STL and MTL with one hidden layer on tasks in the 1D-ALVINN domain. The bold entries in the STL columns are the STL runs that performed best. Differences statistically significant at 0.05 or better are marked with an *.

	ROOT-MEAN SQUARED ERROR ON TEST SET						
TASK	**Single Task Backprop (STL)**				**MTL**	Change MTL	Change MTL
	2HU	4HU	8HU	16HU	16HU	to Best STL	to Mean STL
1 or 2 Lanes	.201	.209	.207	**.178**	**.156**	-12.4% *	-21.5% *
Left Edge	**.069**	.071	.073	.073	**.062**	-10.1% *	-13.3% *
Right Edge	.076	.062	.058	**.056**	**.051**	-8.9% *	-19.0% *
Line Center	.153	**.152**	.152	.152	**.151**	-0.7%	-0.8%
Road Center	.038	**.037**	.039	.042	**.034**	-8.1% *	-12.8% *
Road Greylevel	**.054**	.055	.055	.054	**.038**	-29.6% *	-30.3% *
Edge Greylevel	**.037**	.038	.039	.038	**.038**	2.7%	0.0%
Line Greylevel	.054	.054	**.054**	.054	**.054**	0.0%	0.0%
Steering	.093	**.069**	.087	.072	**.058**	-15.9% *	-27.7% *

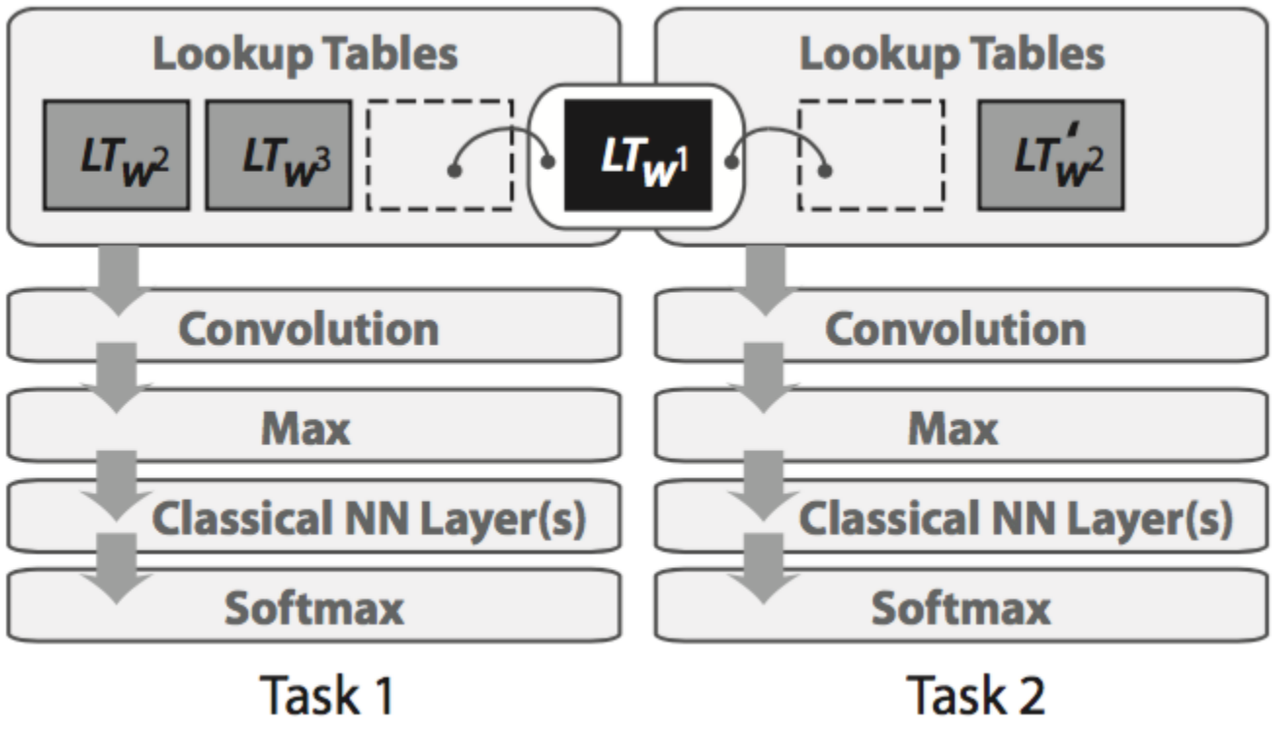
CMU Alvinn MTL (Caruana 1998)

Using the future to predict the present

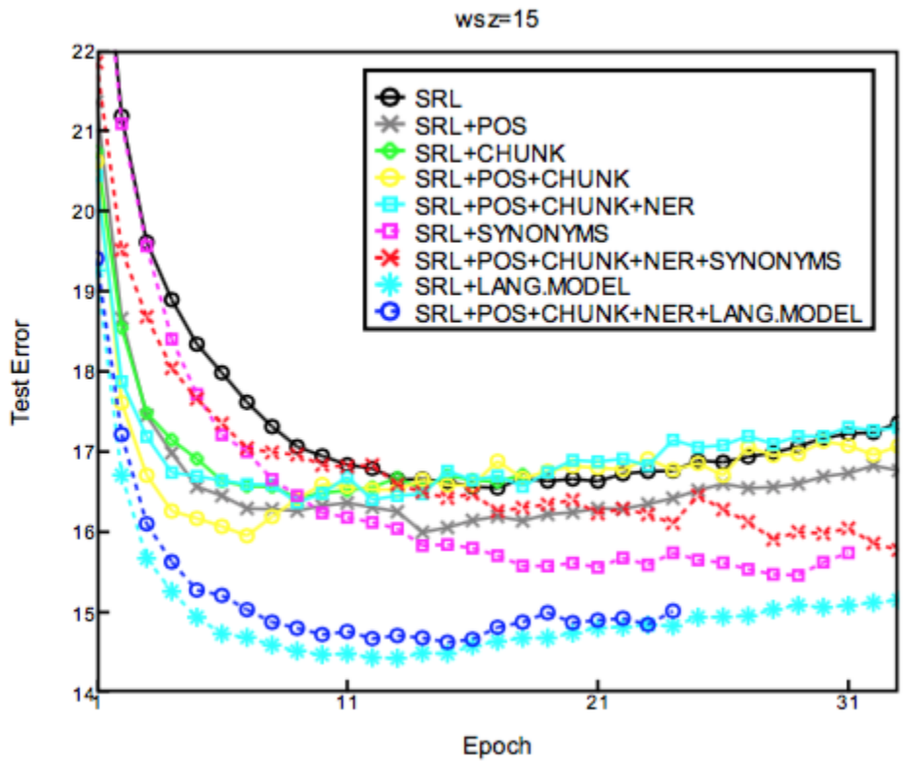


(Caruana 1998) Using future lab results as extra outputs

First approaches in NLP



Collobert and Weston (2008)



Collobert and Weston (2008)

Open domain name error detection

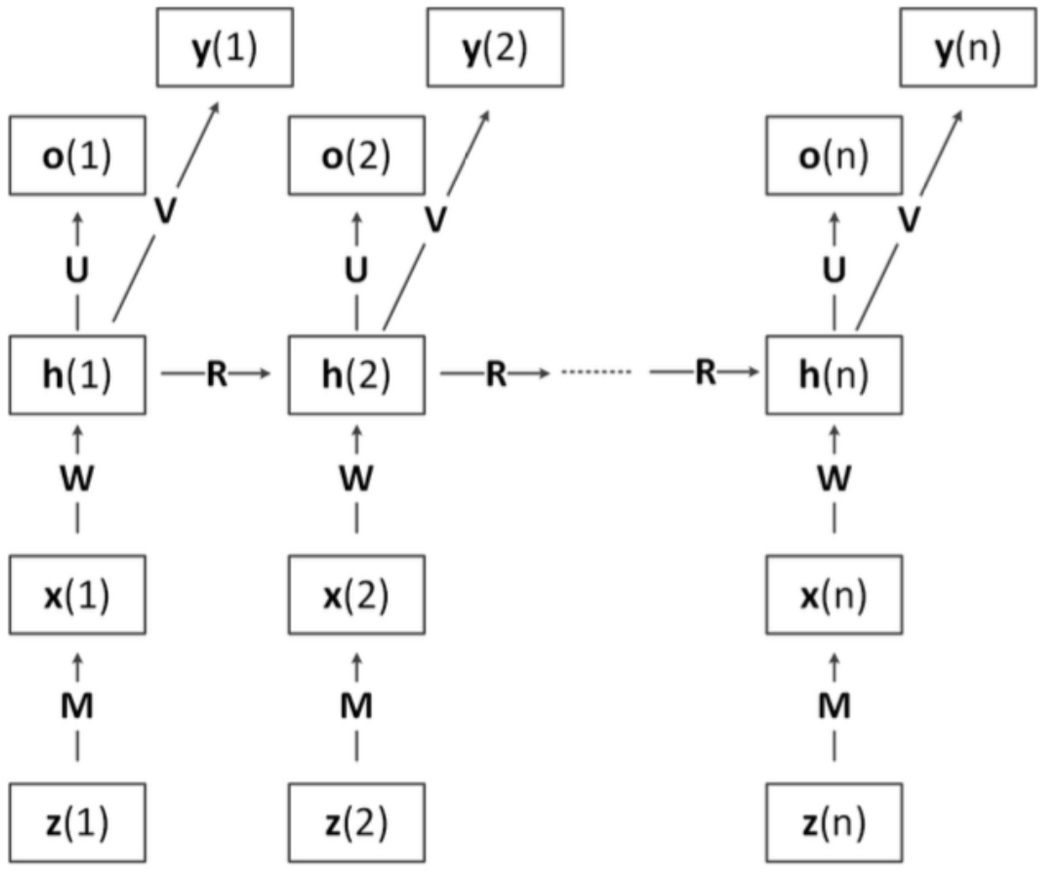


Figure 1: The structure of the proposed MT RNN model, which predicts both the next word $o(t)$ and whether the sentence contains a name $y(t)$ at each time step.

Cheng, Fang, and Ostendorf (2015)

Encoder Decoder models / Sequence to Sequence

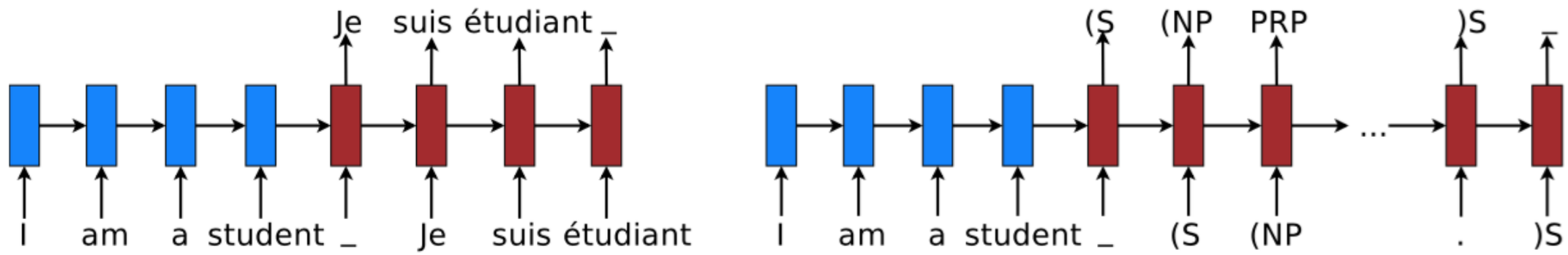
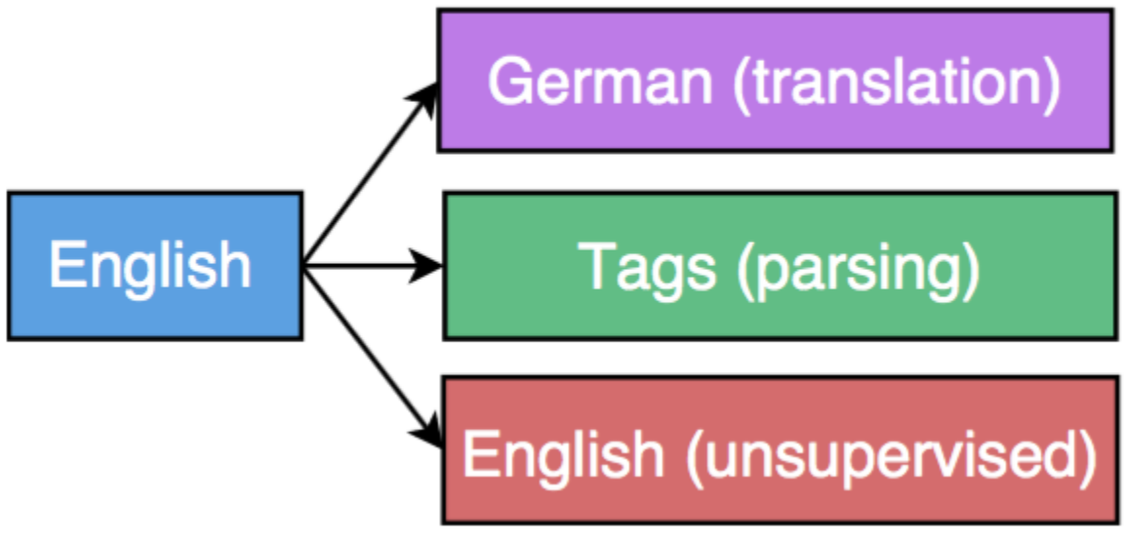


Figure 1: **Sequence to sequence learning examples** – (left) machine translation (Sutskever et al., 2014) and (*right*) constituent parsing (Vinyals et al., 2015a).

Luong et al. (2016)

Sequence to Sequence multi-task learning model



Luong et al. (2016)

All that glitters is not ...

- more computation
- difficulty of defining task relatedness, really knowing when it works
- does not always work

Fortuitous NLP

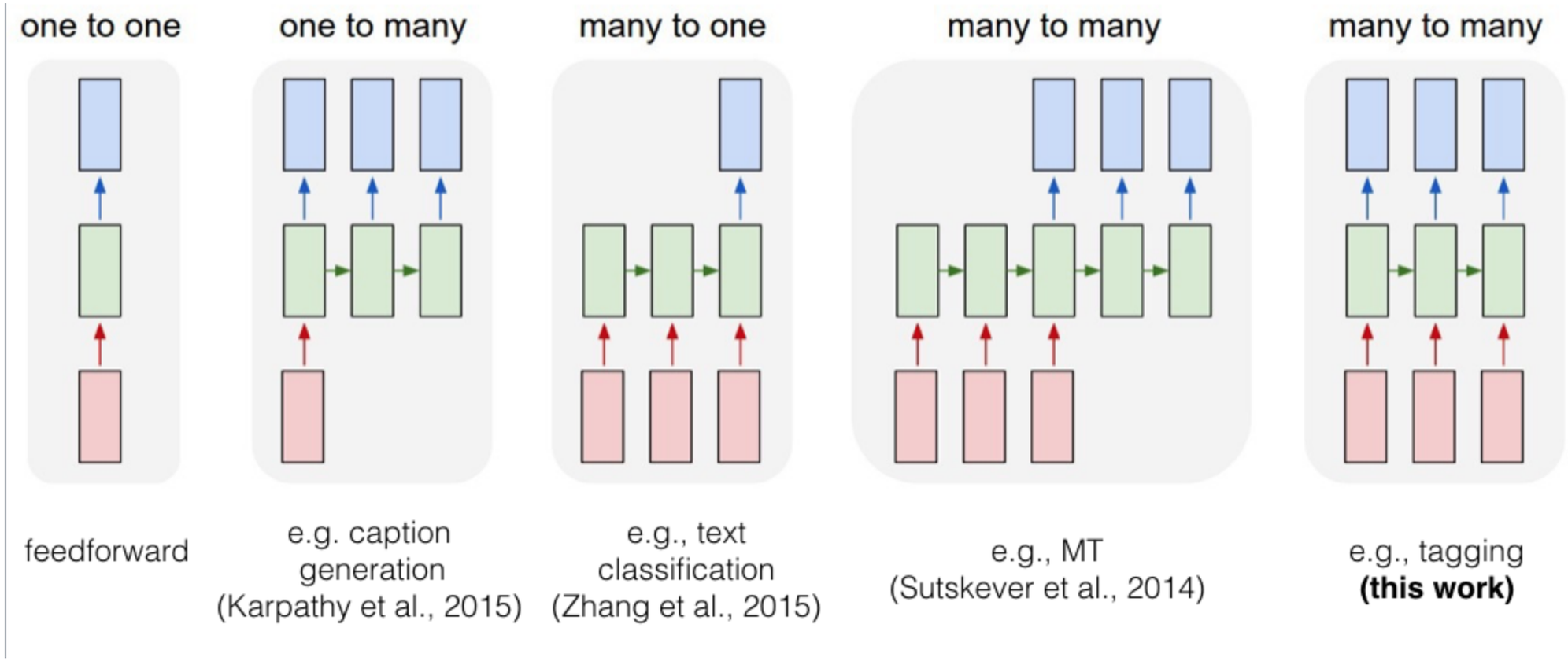
Multilingual POS tagging with auxiliary loss

How affected are neural network-based taggers by...?

- representation
- language
- data set size

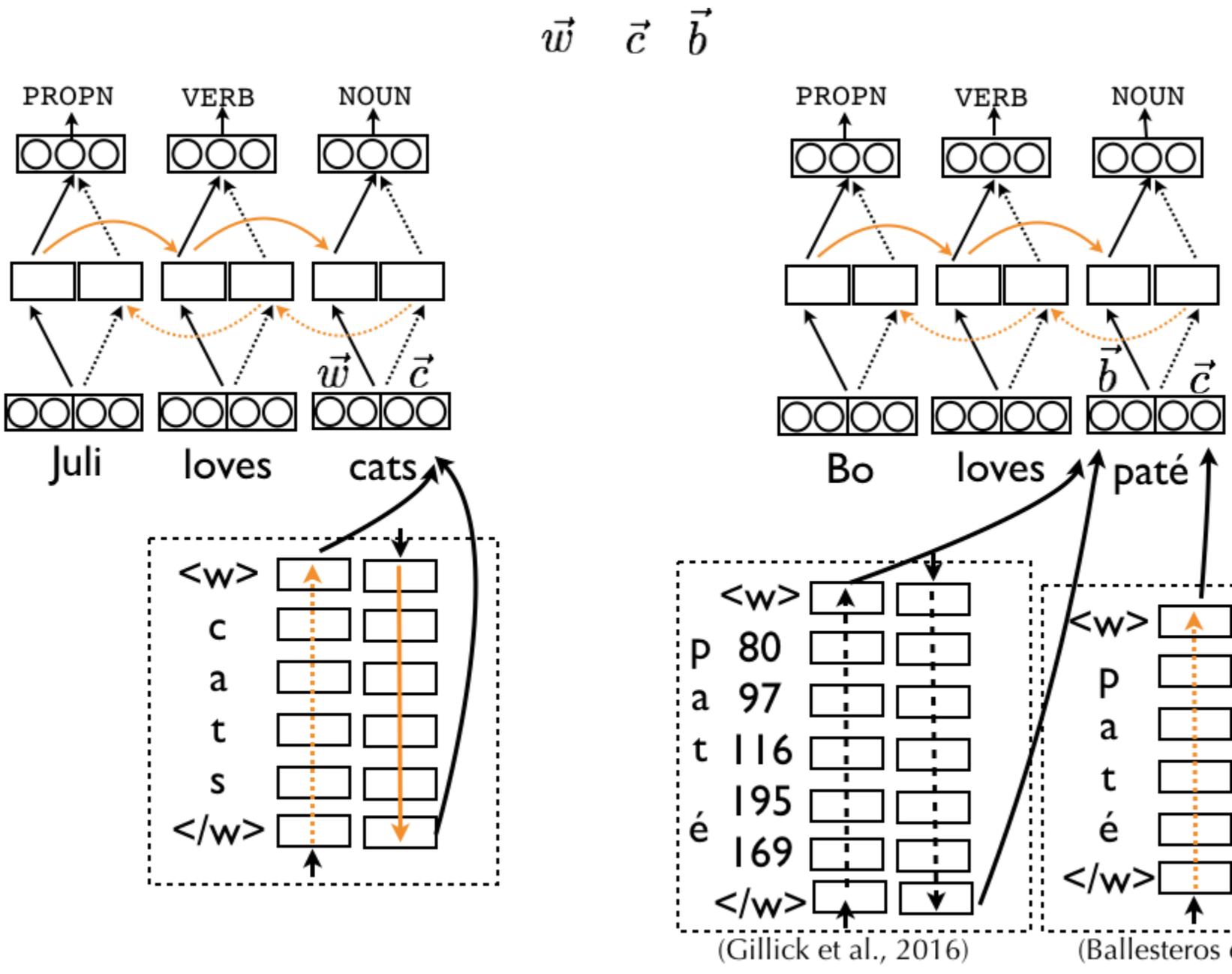
(Plank, Søgaard, and Goldberg 2016)

RNN-based tagger



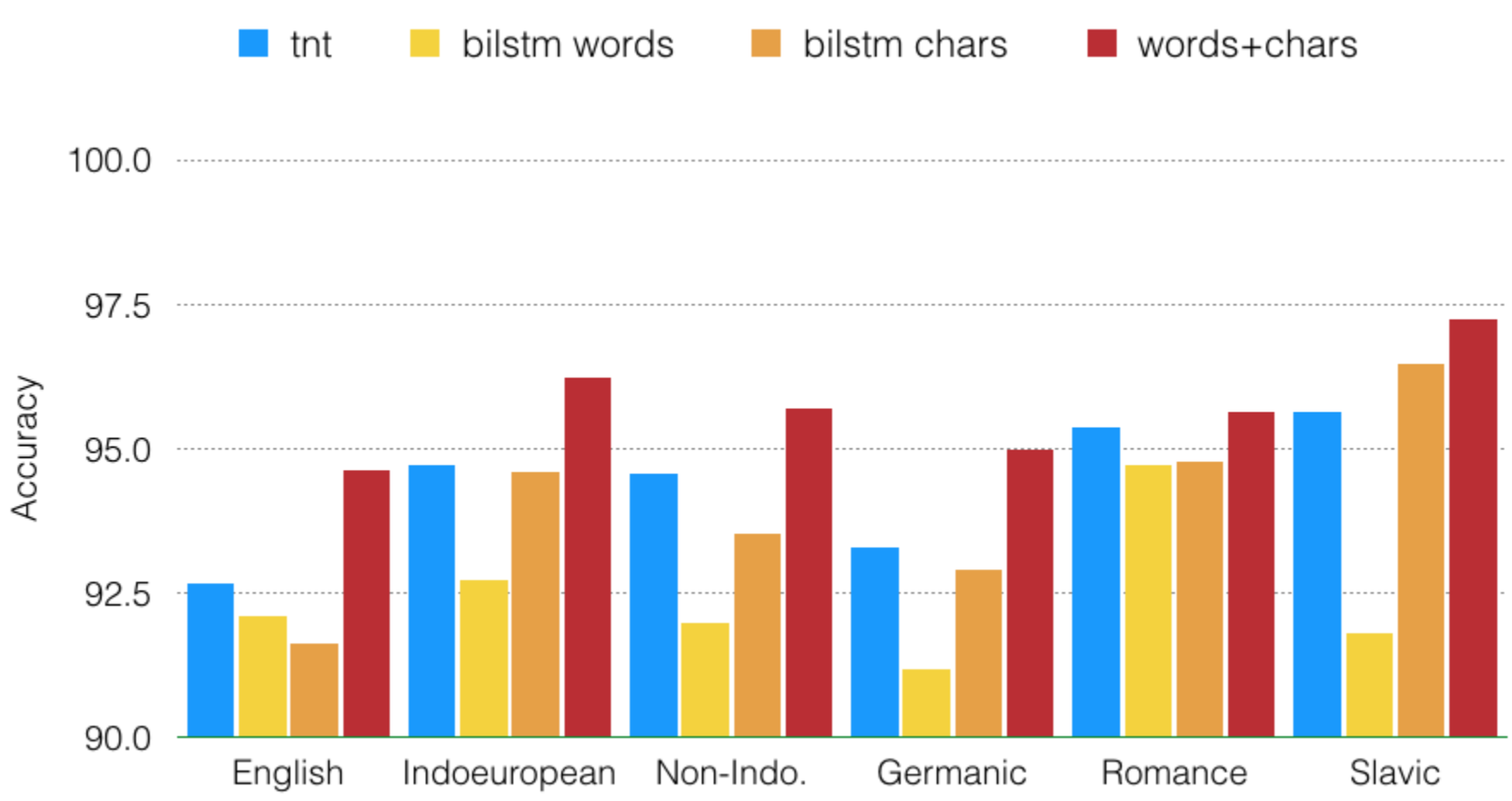
Karpathy

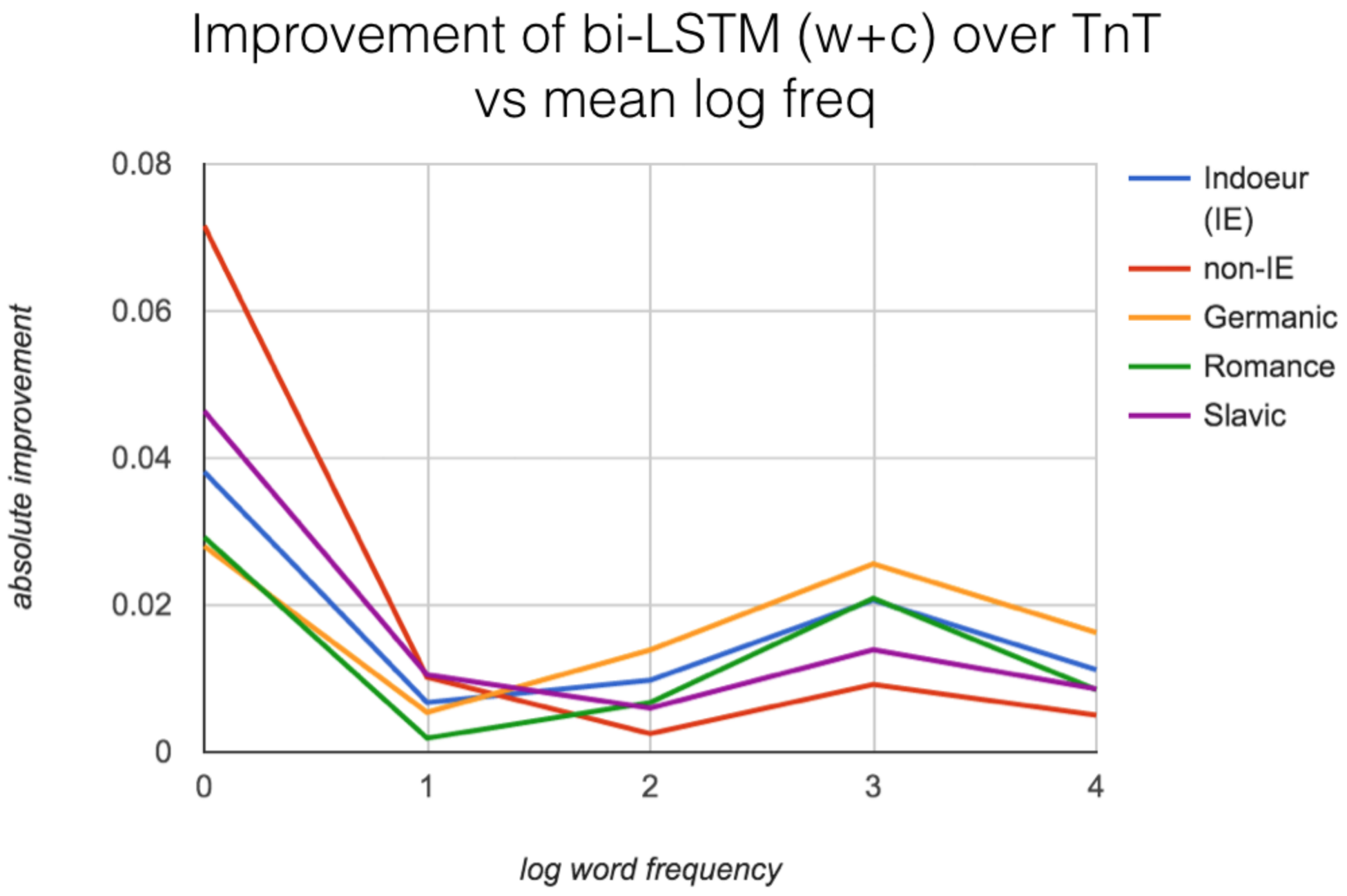
Model



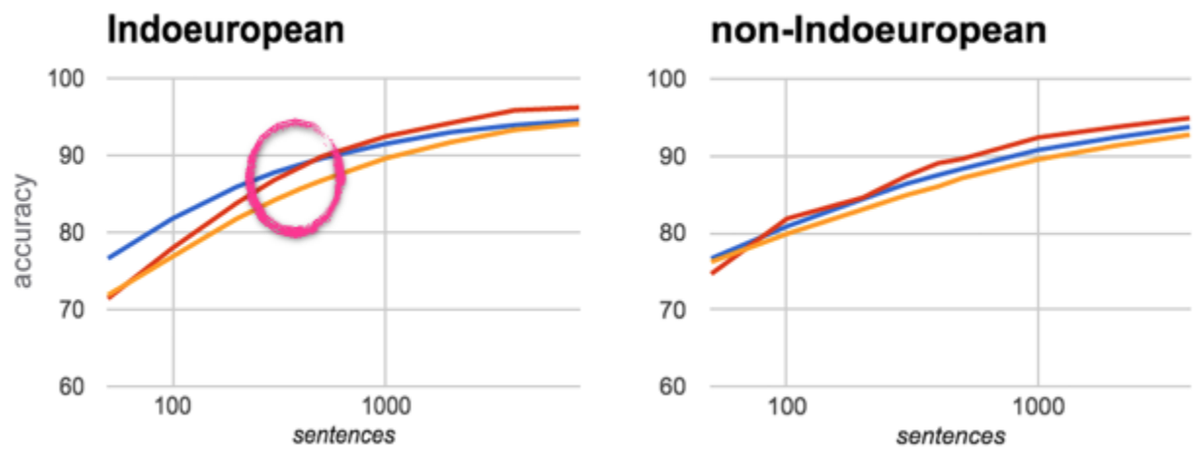
Plank, Søgaard, and Goldberg (2016)

Results


tnt
bilstm words
bilstm chars
words+chars
100.0
97.5
95.0
92.5
90.0
Accuracy
English
Indoeuropean
Non-Indo.
Germanic
Romance
Slavic

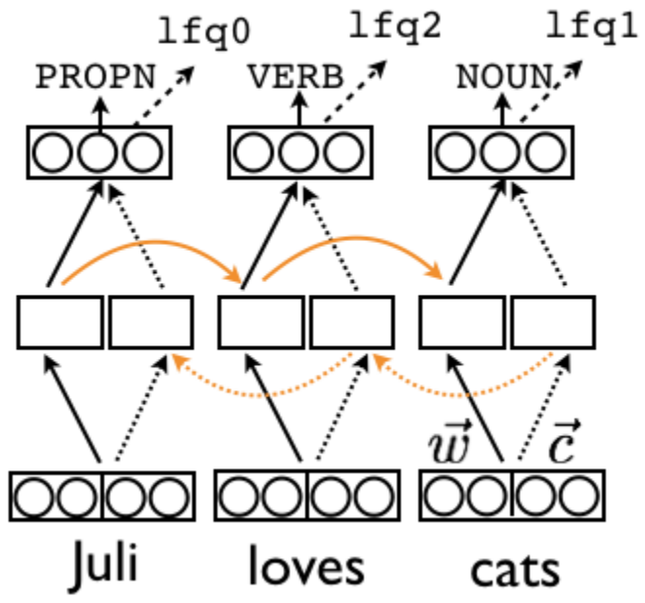

Improvement of bi-LSTM (w+c) over TnT
vs mean log freq
absolute improvement
0.08
0.06
0.04
0.02
0
0
1
2
3
4
log word frequency
Indoeur (IE)
non-IE
Germanic
Romance
Slavic

Learning curves



(more learning curves in paper)

Auxiliary loss



$$L(\hat{y}_t, y_t) + L(\hat{y}_a, y_a)$$

```
freqbin = int(log(freq_train(w)))
```

```
freqbin("the")=5
freqbin("xray")=0
```

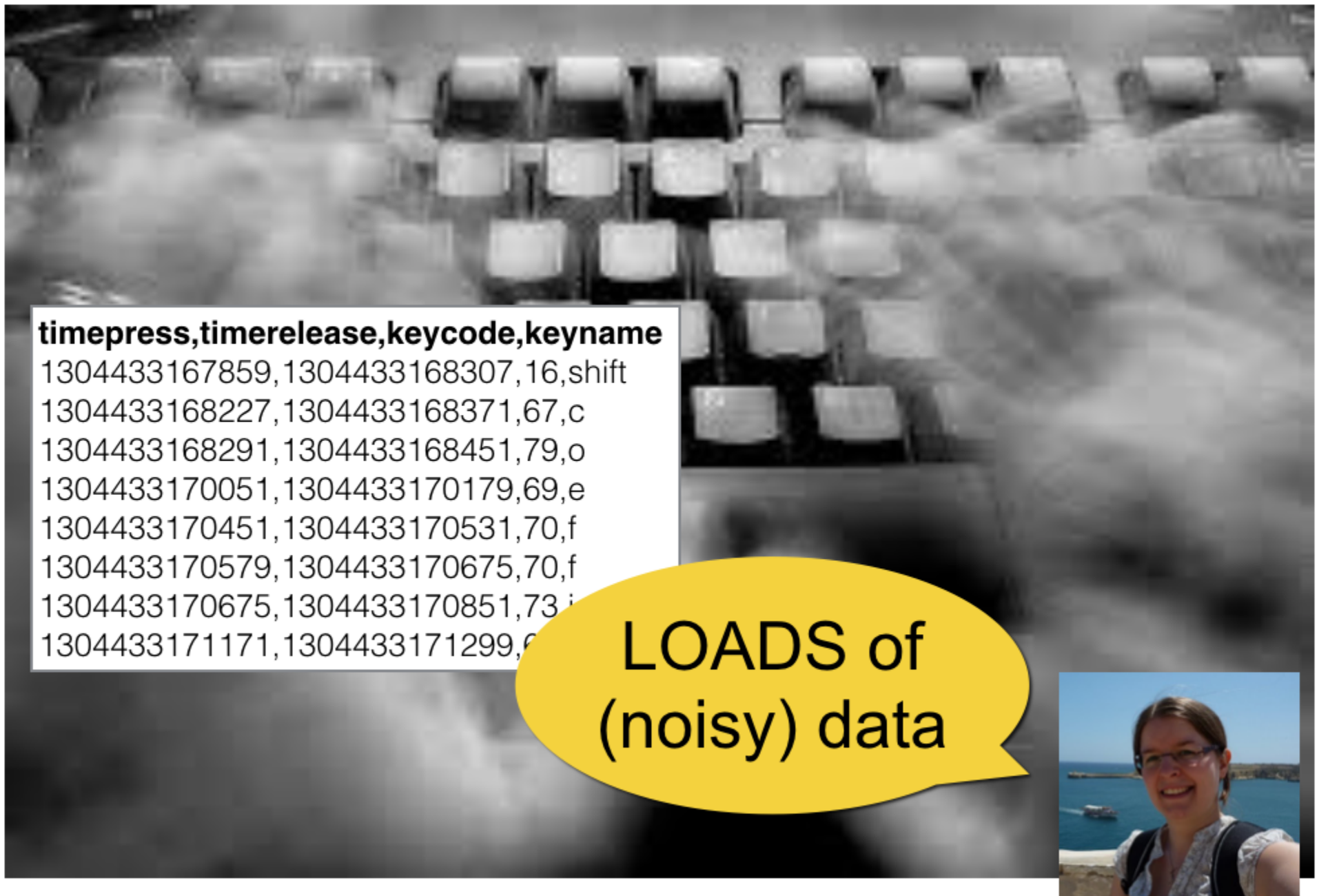
Plank, Søgaard, and Goldberg (2016)

	$\vec{w}+\vec{c}$ +POLYGLOT		OOV ACC	
	bi-LSTM	FREQBIN	bi-LSTM	FREQBIN
avg	**96.50**	**96.52**	83.48	87.98
Indoeur.	**96.63**	**96.63**	82.77	87.63
non-Indo.	96.21	**96.28**	87.44	90.39
Germanic	**95.55**	95.49	81.22	85.45
Romance	**96.93**	**96.93**	81.31	86.07
Slavic	97.42	**97.50**	86.66	91.69
ar	98.87	**98.91**	95.04	96.21
bg	**98.23**	97.97	87.40	90.56
cs	98.02	**98.24**	89.02	91.30
da	96.16	**96.35**	77.09	86.35
de	**93.51**	93.38	81.95	86.77
en	**95.17**	95.16	71.23	80.11
es	95.67	**95.74**	71.38	79.27
eu	95.38	**95.51**	79.87	84.30
fa	**97.60**	97.49	80.00	89.05
fi	95.74	**95.85**	86.34	88.85
fr	**96.20**	96.11	78.09	83.54
he	96.92	**96.96**	80.11	88.83
hi	96.97	**97.10**	81.19	85.27
hr	96.27	**96.82**	84.62	92.71
id	93.32	**93.41**	88.25	87.67
it	97.90	**97.95**	83.59	89.15
nl	**93.82**	93.30	76.62	75.95
no	**98.06**	98.03	92.05	93.72
pl	**97.63**	97.62	91.77	94.94
pt	**97.94**	97.90	92.16	92.33
sl	**96.97**	96.84	80.48	88.94
sv	96.60	**96.69**	88.37	89.80

Take-home message

- LSTM-based tagger less suspectible to large data requirement than assumed
- Char embeddings especially helpful for Slavic and non-IE languages
- Alternative view of data (fortuitous data!) via multi-task learning helpful!

Example 2: Are keystroke logs informative for NLP?



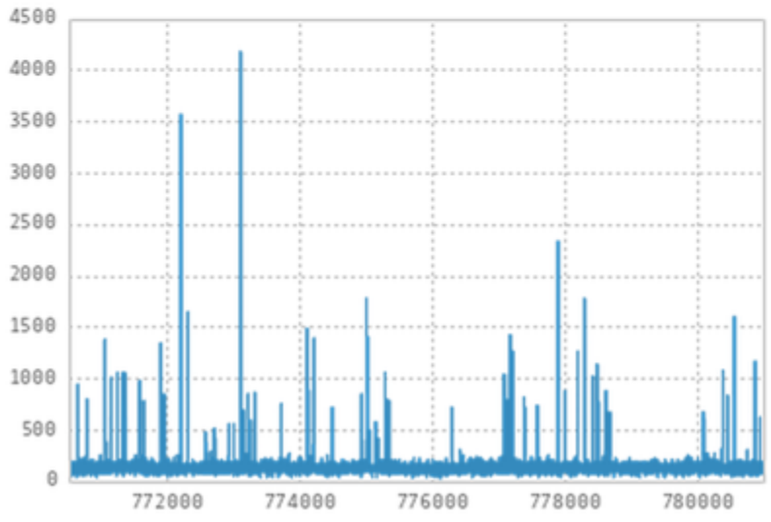
Typology of fortuitous data

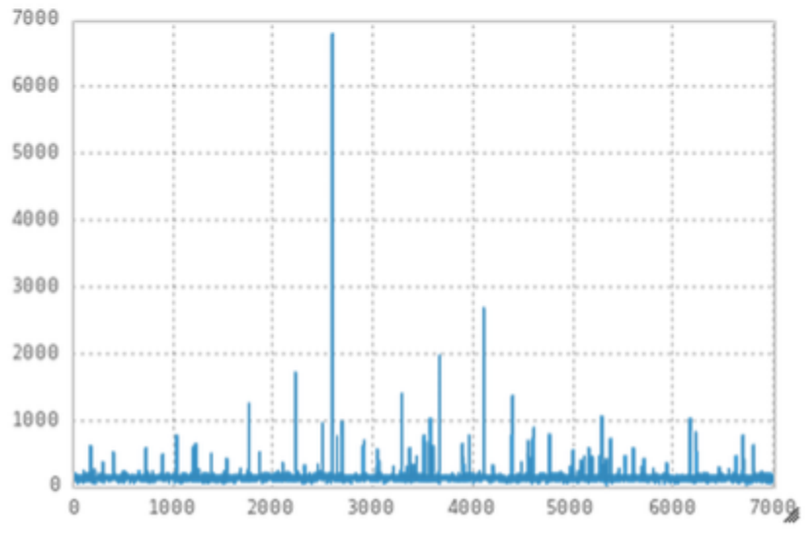
Side benefit of:	availability	readiness
User-generated content	+	+
Annotation	-	+
Behavior	+	-

Table 1: Typology of fortuitous data

Motivation

Do pre-word pauses carry syntactic information?





>500ms pauses

==========

'is': 176

'a': 80

==========

'measure': 1424

'used': 656

'in': 192

==========

'statisitcal': 3200

'model': 1568

==========

'analysis': 1824

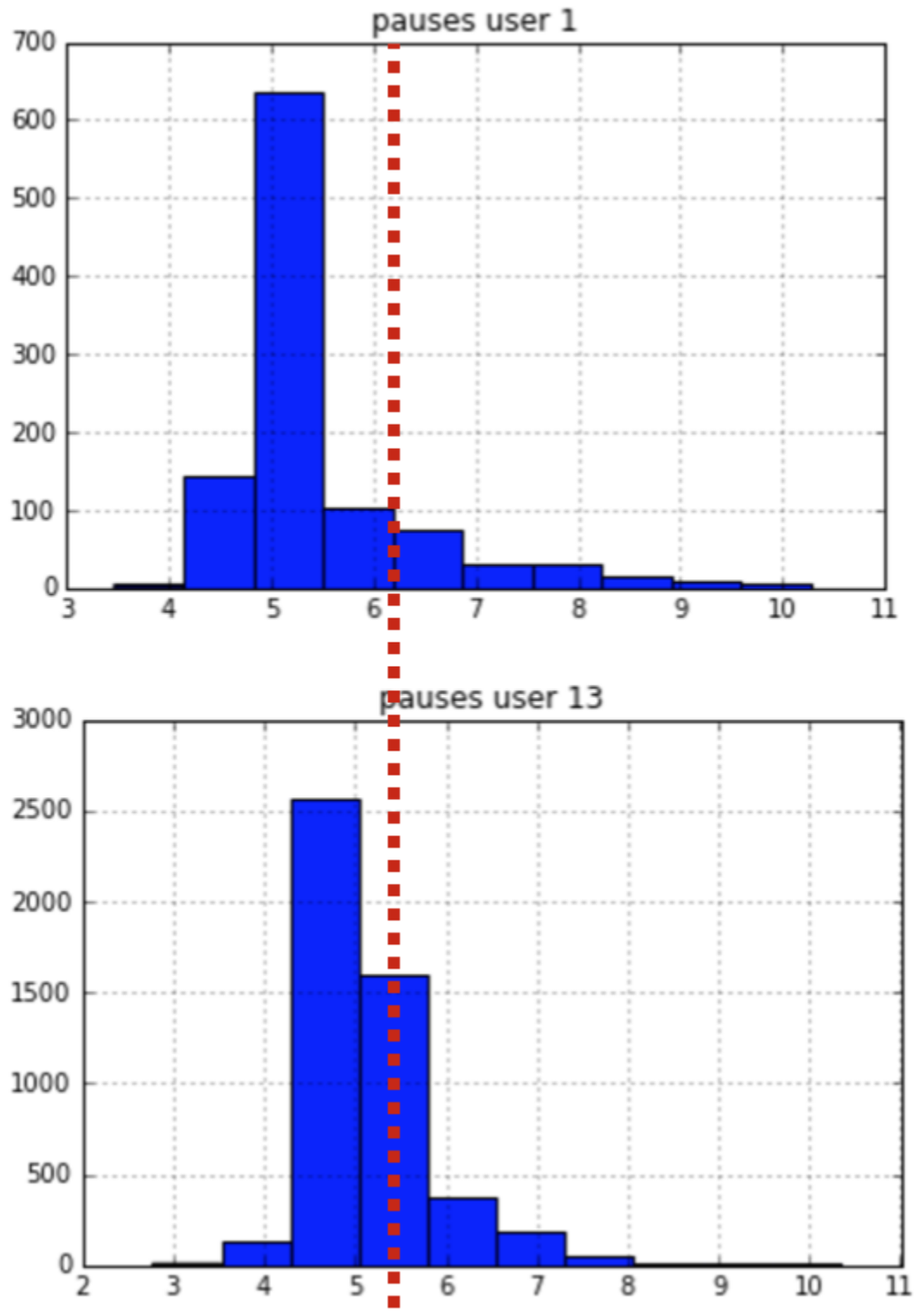
[is a]

[measure used in]

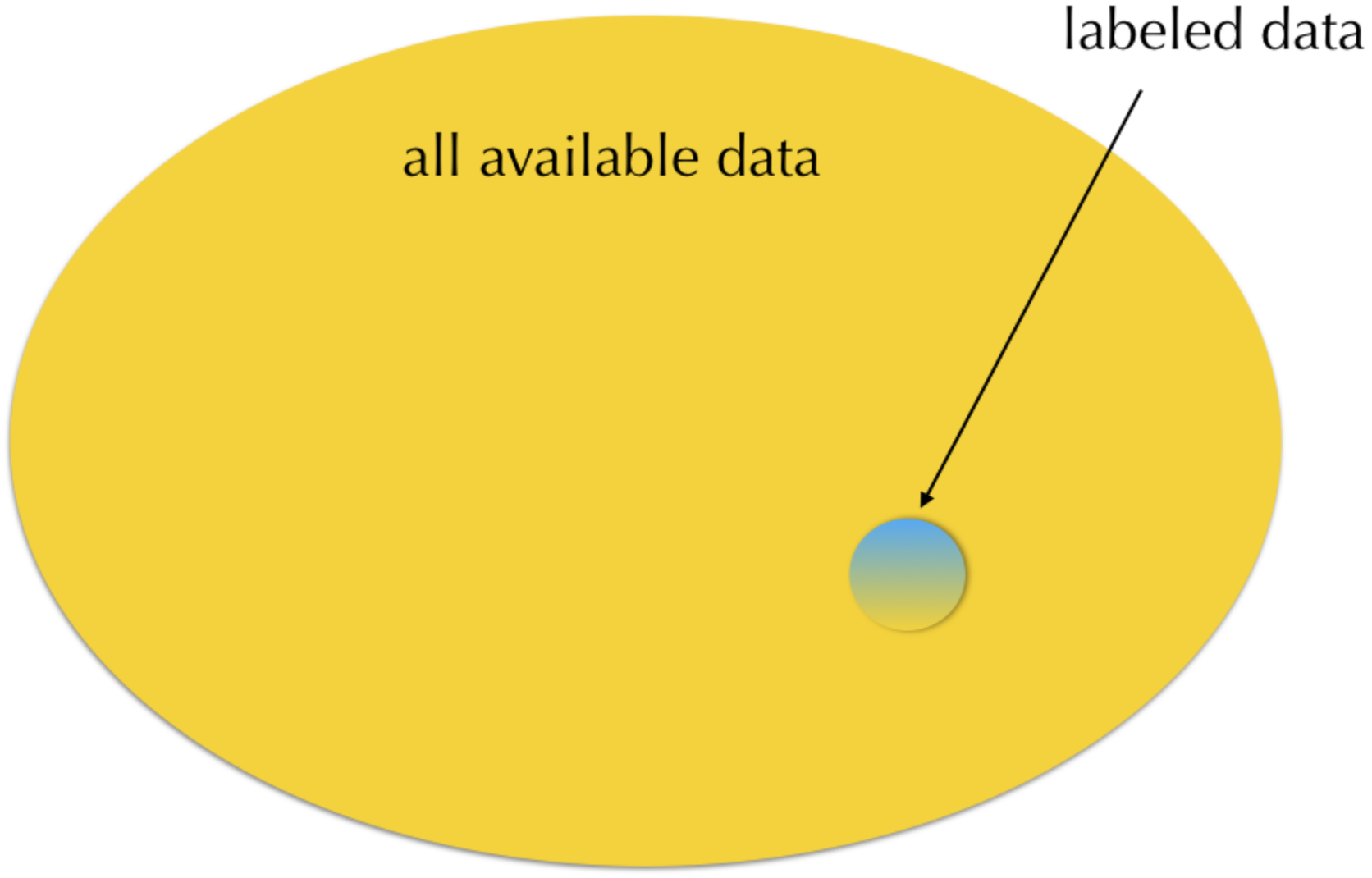
[statistical model]

[analysis]

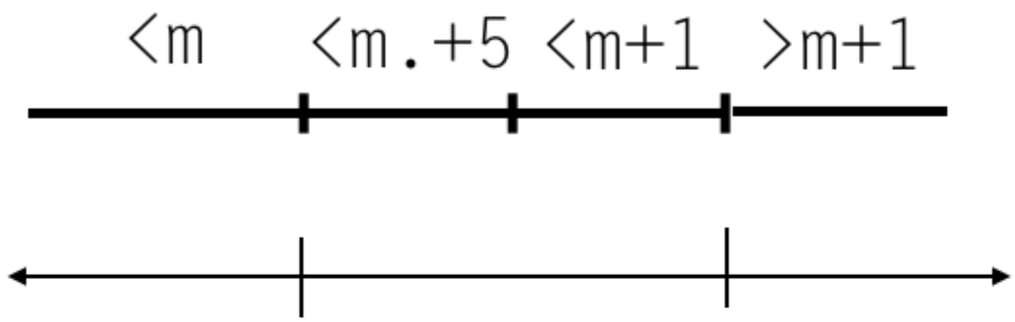
500ms (6.2) seems arbitrary




Step 1: *refine* the data


labeled data
all available data

From keystrokes to labels



m *+mad*

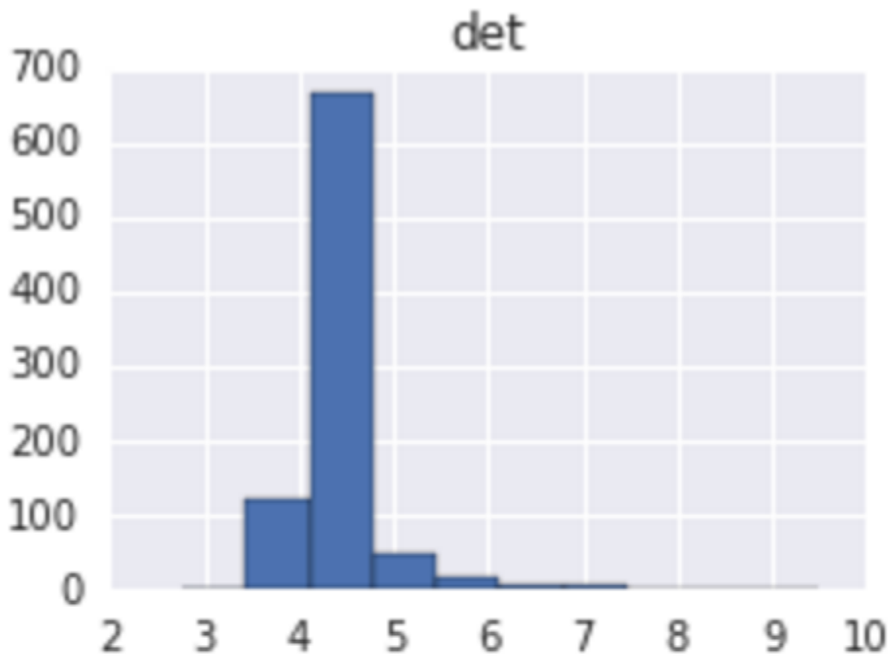
BIO encoding

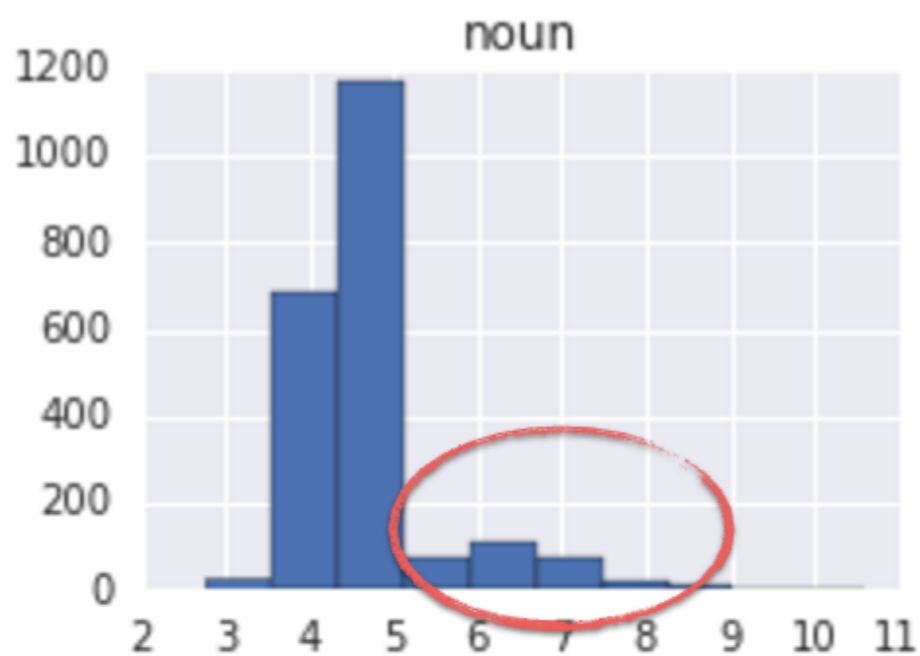
O: punctuation

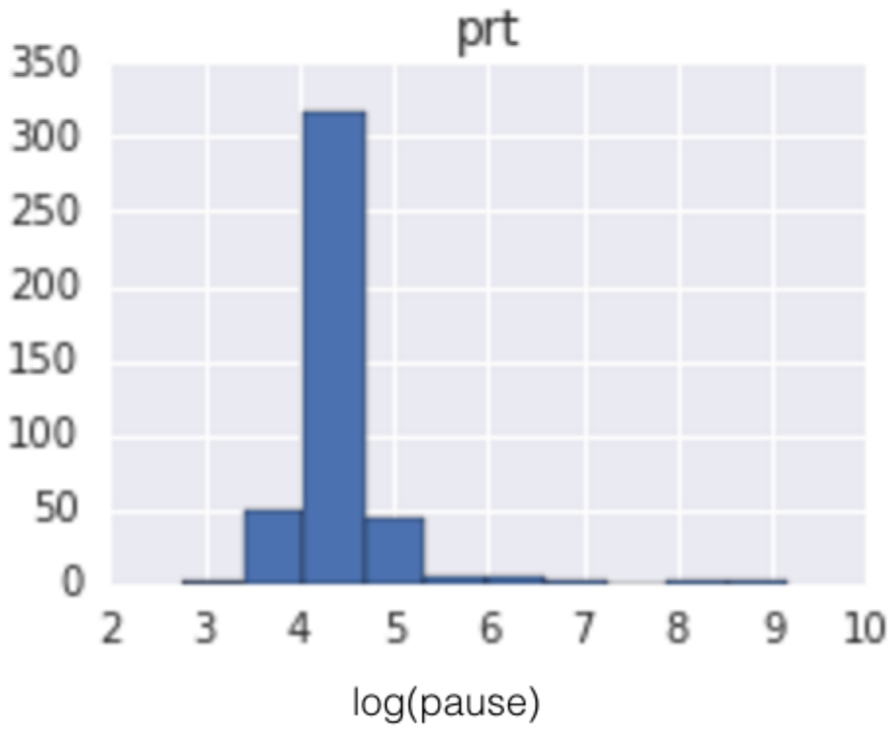
B-<m	I-<m	I-<m	I-<m	B-<m+.5	B-<m+1	B->m+1
the	closer	the	number	is	to	1

Table 1: Example auto-derived keystroke annotation.

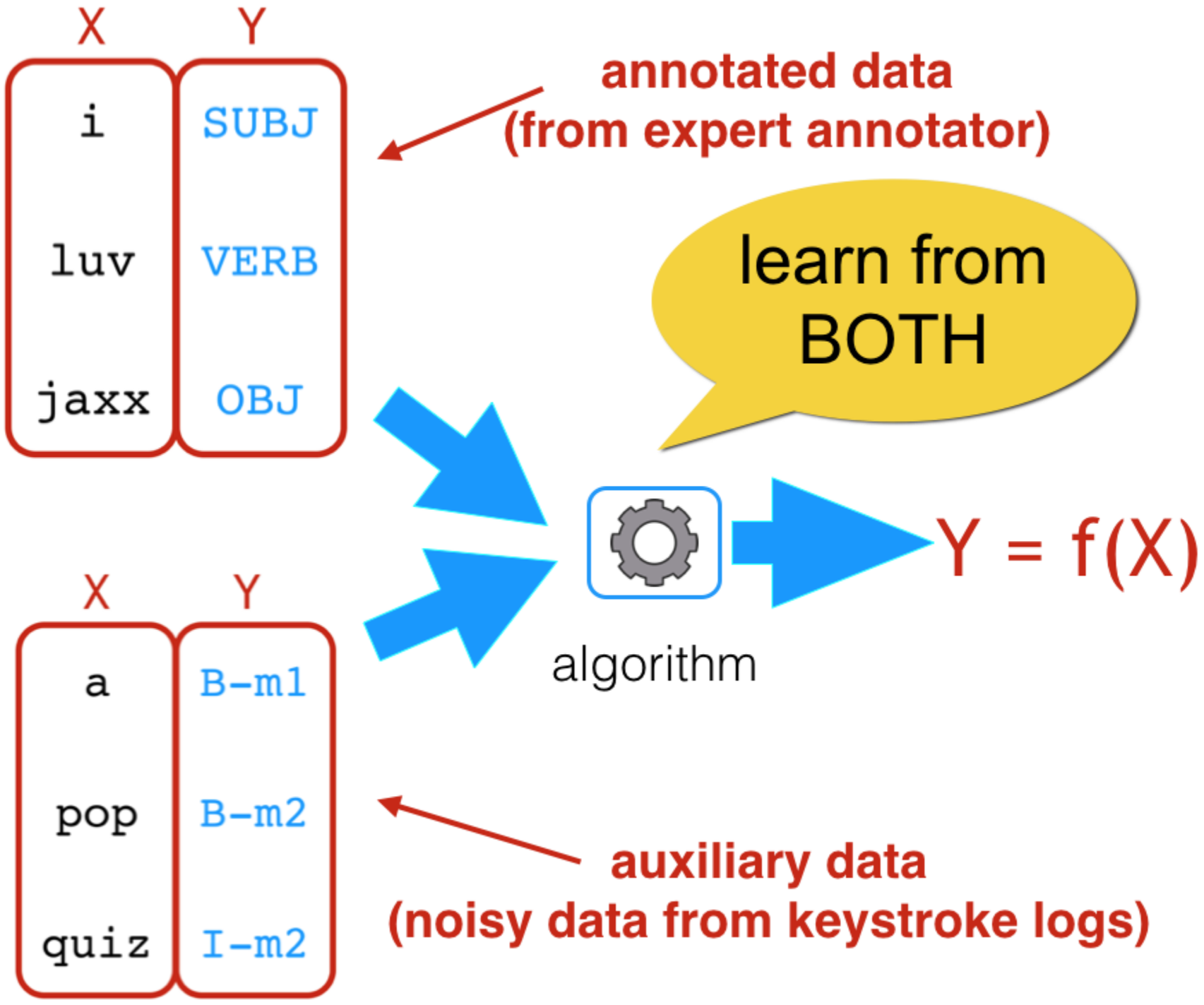
Word pauses and POS



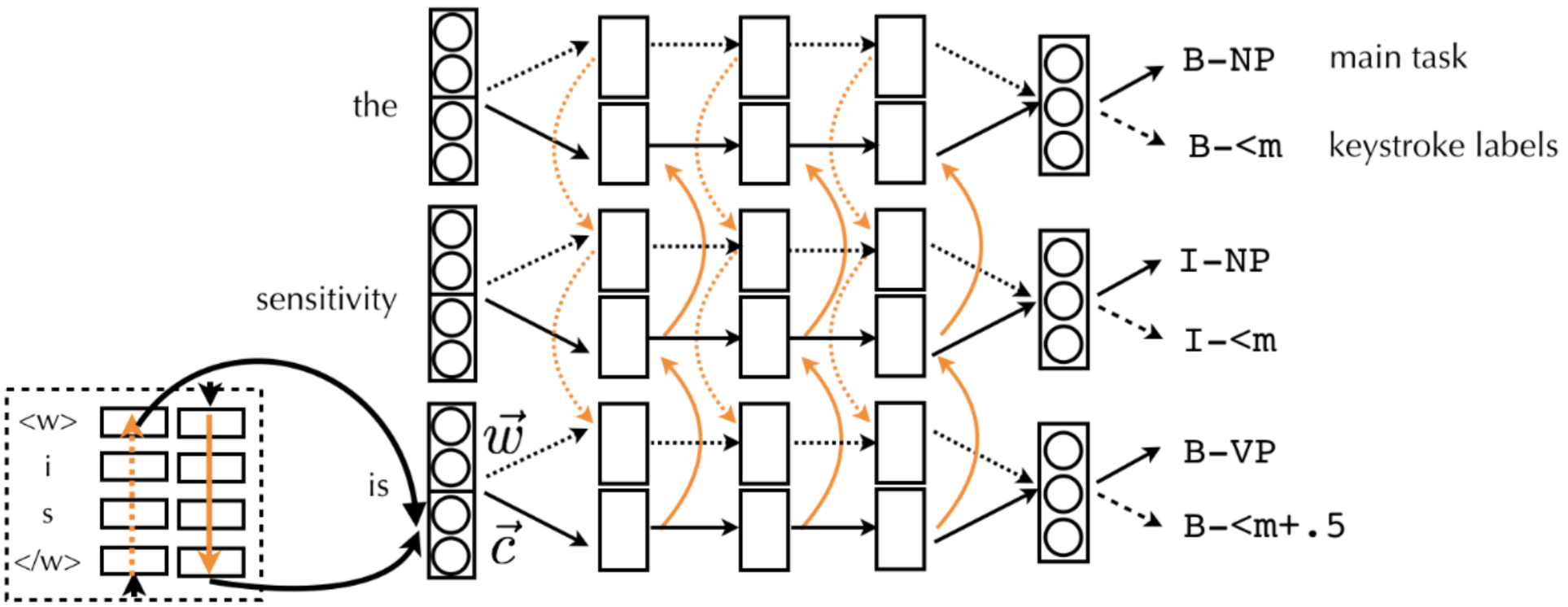




Multi-task learning



Model


the
sensitivity
is
<w>
i
s
</w>
$\vec{w}$
$\vec{c}$
B-NP
main task
B-<m
keystroke labels
I-NP
I-<m
B-VP
B-<m+.5

Results

		FOSTER.DEV	FOSTER.TEST	RITTER
Baseline	NP	72.18	71.41	61.76
	VP	70.25	73.44	75.13
	PP	93.25	91.85	89.05
+PAUSE	NP	**74.03**	**72.83**	**62.41**
	VP	**70.38**	**75.09**	75.05
	PP	93.13	90.74	88.96

Table 6: Chunking results per label.

Token	Gold	Baseline	Model
Auburn	B-NP	I-NP	B-NP
party	I-NP	I-NP	I-NP
at	B-PP	B-PP	B-PP
man	B-NP	B-NP	B-NP
utd	I-NP	B-VP	I-NP
sounds	B-VP	B-VP	B-VP
bithcy	B-ADJP	B-VP	B-ADJP
)	O	I-NP	O

(also promising results for CCG tagging)

References

References

Caruana, Rich. 1998. “Multitask Learning.” In *Learning to Learn*, 95–133. Springer.

Cheng, Hao, Hao Fang, and Mari Ostendorf. 2015. “Open-Domain Name Error Detection Using a Multi-Task Rnn.” In. EMNLP.

Collobert, Ronan, and Jason Weston. 2008. “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning.” In *Proceedings of the 25th International Conference on Machine Learning*, 160–67. ACM.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. “Natural Language Processing (Almost) from Scratch.” *The Journal of Machine Learning Research* 12. JMLR. org: 2493–2537.

Goldberg, Yoav. 2015. “A Primer on Neural Network Models for Natural Language Processing.” *ArXiv* abs/1510.00726.

Luong, Minh-Thang, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. “Multi-Task Sequence to Sequence Learning.”

ArXiv Preprint ArXiv:1511.06114.

Mikolov, T, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." *Advances in Neural Information Processing Systems*.

Plank, Barbara, Anders Søgaard, and Yoav Goldberg. 2016. "Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss." In *ACL*.