Recurrent Neural Networks II

Suleyman Demirel University

CSS634: Deep Learning

PhD Abay Nussipbekov

Word Representation

V = [a, aaron, …, zulu, <UNK>]

1-hot representation

$$
\begin{array}{cccccc}
\text{Man} & \text{Woman} & \text{King} & \text{Queen} & \text{Apple} & \text{Orange} \\
(5391) & (9853) & (4914) & (7157) & (456) & (6257) \\
\begin{bmatrix}0\\0\\0\\0\\\vdots\\1\\\vdots\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\\vdots\\1\\\vdots\\0\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} &
\begin{bmatrix}0\\\vdots\\1\\\vdots\\0\\0\\0\\0\\0\end{bmatrix} &
\begin{bmatrix}0\\0\\0\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix} \\
O_{5391} & O_{9853} & & & &
\end{array}
$$

I want a glass of orange ______.

I want a glass of apple______.

Word Representation

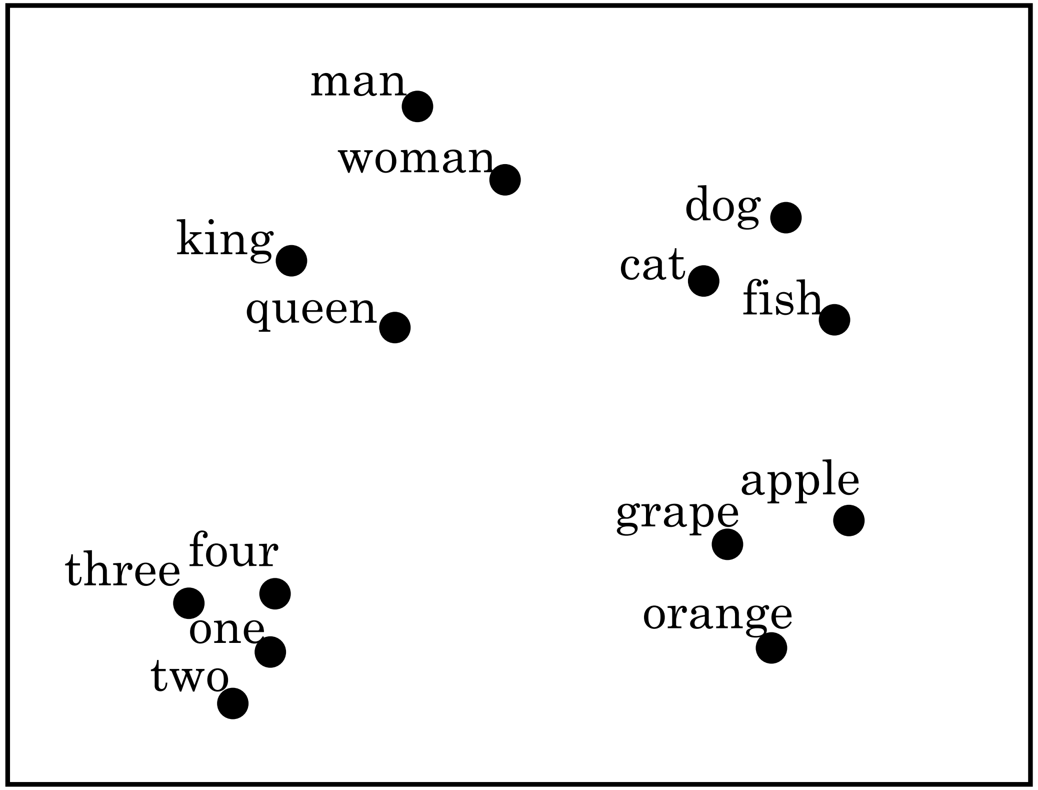
	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.7	0.69	0.03	-0.02
Food	0.04	0.01	0.02	0.01	0.95	0.97
Size						
Cost						
Verb						

similar vectors

I want a glass of orange ______.

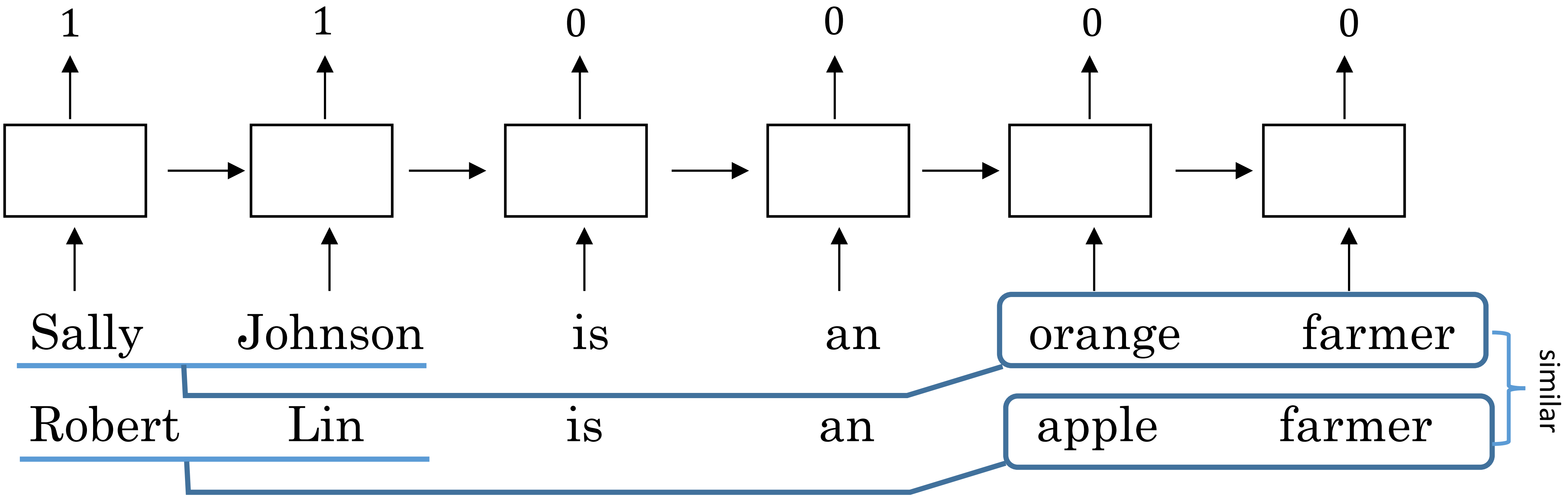
I want a glass of apple______.

Word Representation

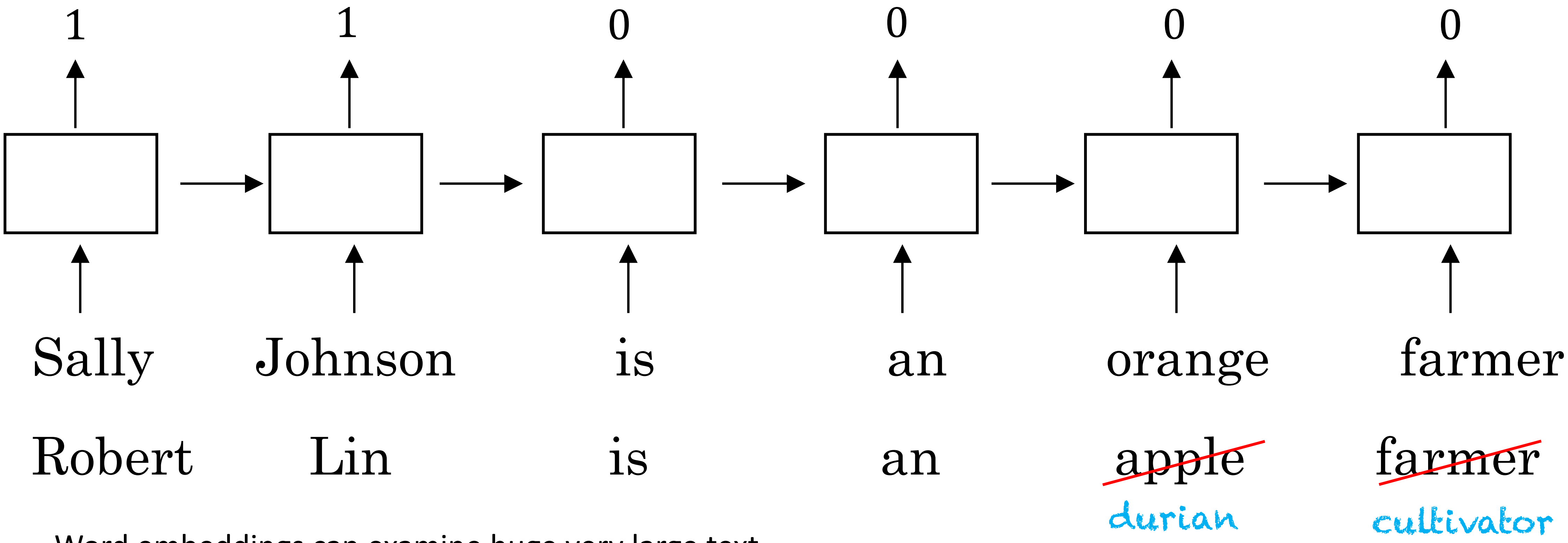


t-SNE

Named Entity Recognition Example



Named Entity Recognition Example



Word embeddings can examine huge very large text corpuses (eg. 100 billion words) – self supervised learning

TRANSFER LEARNING!

haven't seen in training set but learned in word embedding -> will be able to generalize and understand that it is also a person

Transfer Learning and Word Embeddings

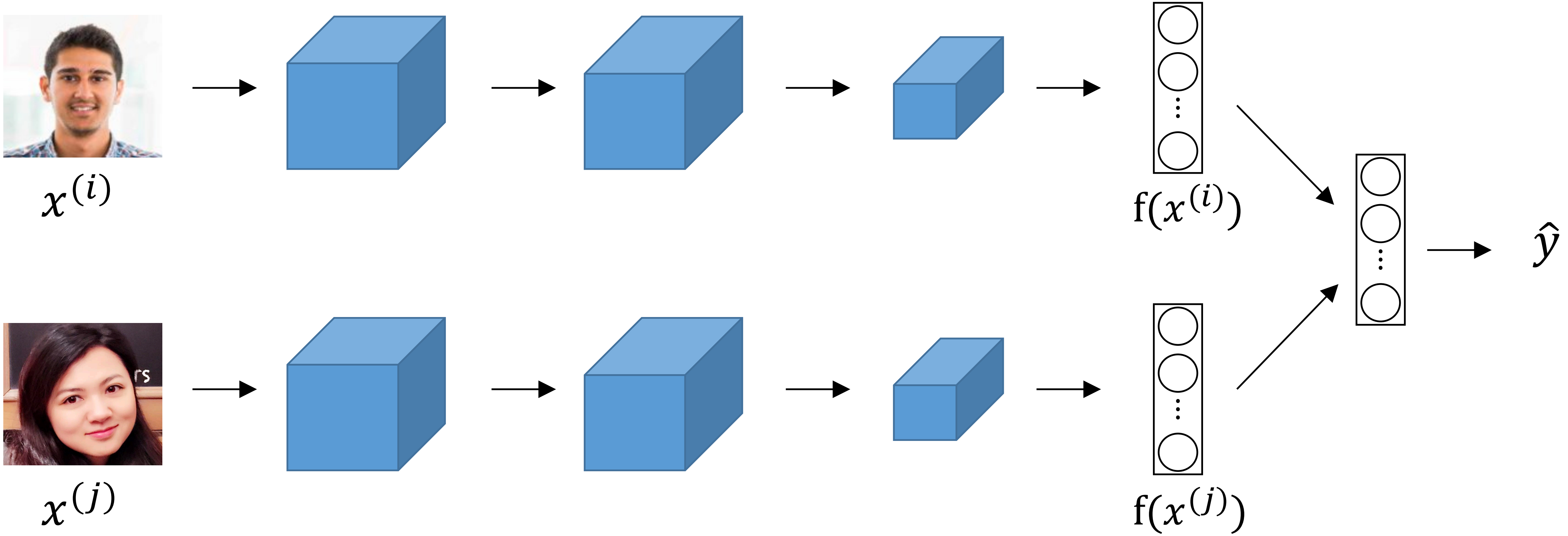
1. Learn word embeddings from large text corpus. (1-100B words)

   (Or download pre-trained embedding online.)

2. Transfer embedding to new task with smaller training set. (say, 100k words)

3. Optional: Continue to finetune the word embeddings with new data.

Relation to Face Encoding (embedding)



The only difference is that in face recognition we can take a face which we haven't seen before while in word embeddings we have a fixed vocabulary

Analogies

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0.00	0.01
Royal	0.01	0.02	0.93	0.95	-0.01	0.00
Age	0.03	0.02	0.70	0.69	0.03	-0.02
Food	0.09	0.01	0.02	0.01	0.95	0.97

e_{391}
e_{man} e_{woman}

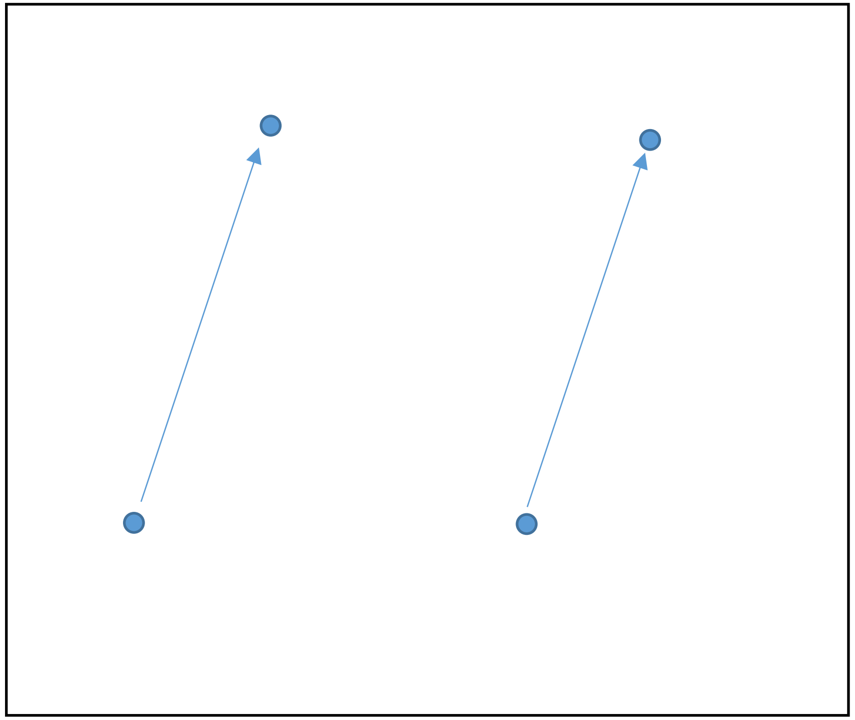
$$e_{man} - e_{woman} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Man → Woman as King → ?

$$e_{king} - e_{queen} \approx \begin{bmatrix} -2 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

[Mikolov et. al., 2013, Linguistic regularities in continuous space word representations]

Analogies Using Word Vectors



300 D

$$e_{man} - e_{woman} \approx e_{king} - e_?$$

Find word w: $\arg\max_{w} sim(e_w, e_{king} - e_{man} + e_{woman})$

Cosine similarity

$$sim(e_w, e_{king} - e_{man} + e_{woman})$$

$$sim\,(u, v) = \frac{u^T v}{\|u\|_2 \|v\|_2}$$

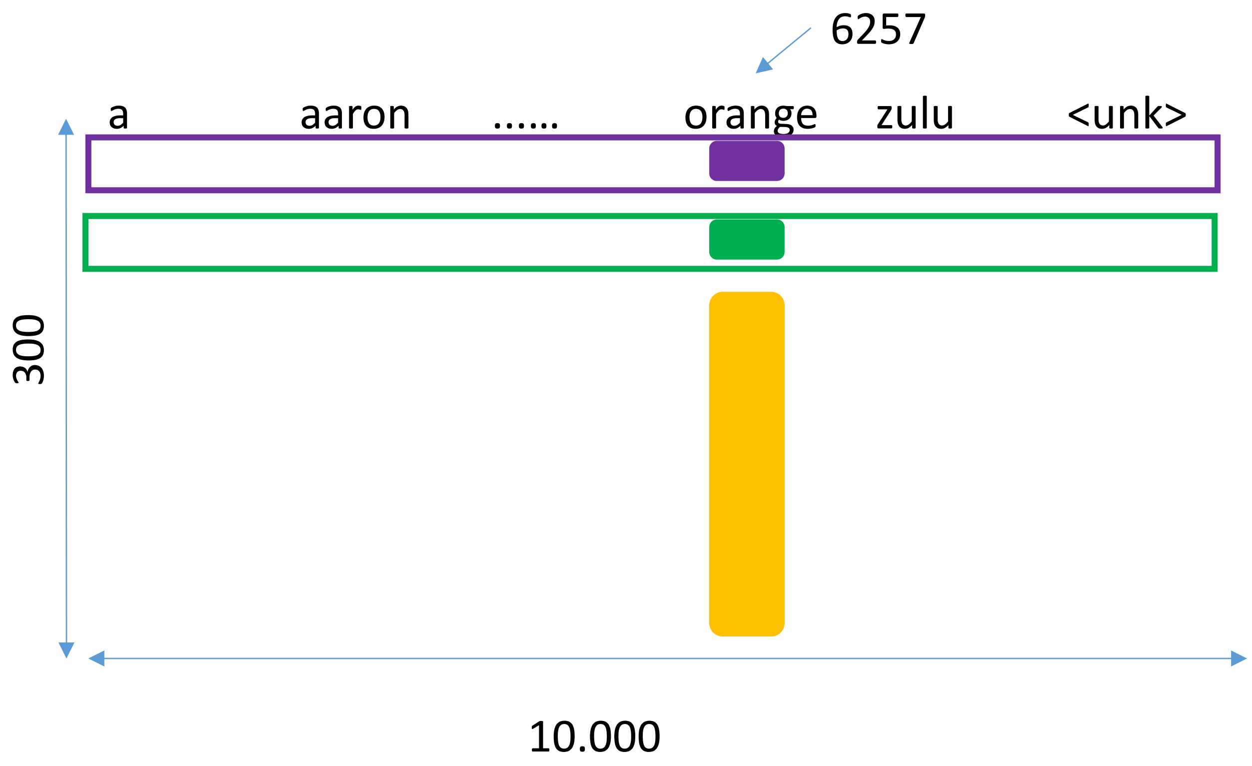
Man:Woman as Boy:Girl

Ottawa:Canada as Nairobi:Kenya

Big:Bigger as Tall:Taller

Yen:Japan as Ruble:Russia

Embedding Matrix



O_{6257}

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ 1 \\ \vdots \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

6257

10.000

$E \cdot O_{6257} =$  $= e_{6257}$ → $E \cdot o_j = e_j$ (embedding for word j)

In practice, use specialized function to look up an embedding.

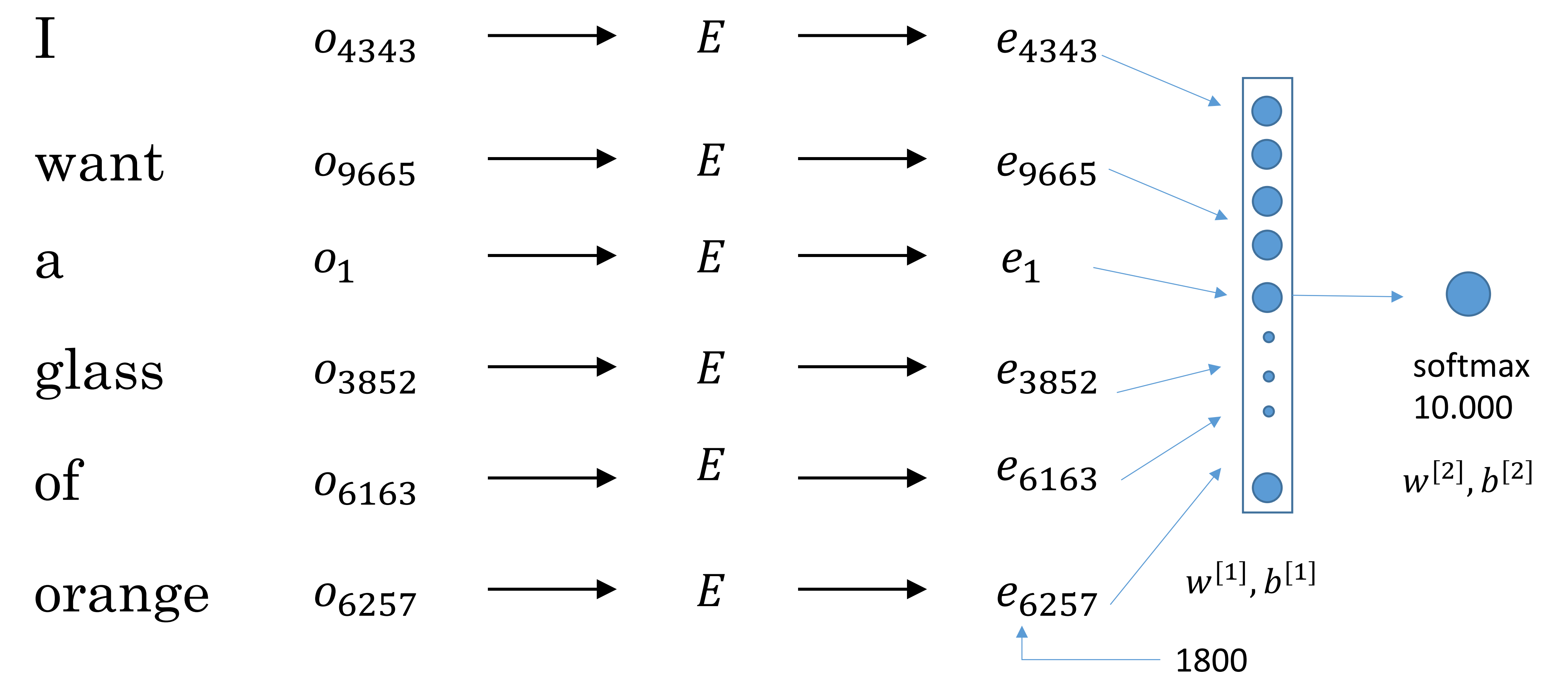
Neural Language Model

I	want	a	glass	of	orange	______.
4343	9665	1	3852	6163	6257	

It's turns out that by learning a language model (predicting next word given a sequence) will help us to learn word embeddings

Neural Language Model

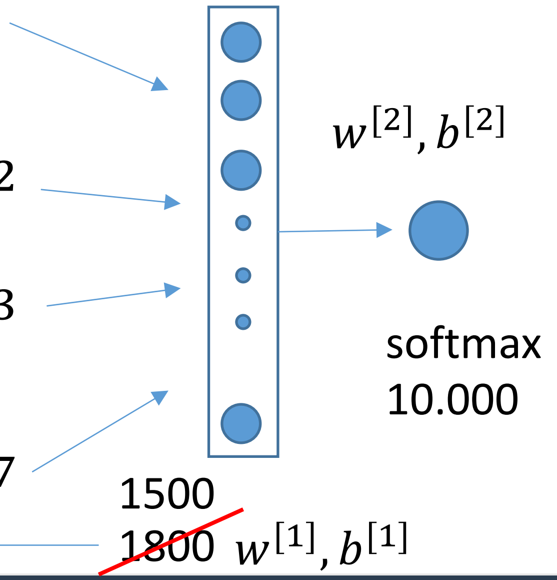
I	want	a	glass	of	orange	______.
4343	9665	1	3852	6163	6257	



Neural Language Model

I	want	a	glass	of	orange	______.
4343	9665	1	3852	6163	6257	

I	o_{4343}	$\longrightarrow$	E	$\longrightarrow$	e_{4343}	
want	o_{9665}	$\longrightarrow$	E	$\longrightarrow$	e_{9665}	
a	o_{1}	$\longrightarrow$	E	$\longrightarrow$	e_{1}	
glass	o_{3852}	$\longrightarrow$	E	$\longrightarrow$	e_{3852}	
of	o_{6163}	$\longrightarrow$	E	$\longrightarrow$	e_{6163}	
orange	o_{6257}	$\longrightarrow$	E	$\longrightarrow$	e_{6257}	



Other Context/Target Pairs

I want a glass of orange juice to go along with my cereal.

context target

Context: Last 4 words.

4 words on left & right	a glass of orange ____?___ to go along with
Last 1 word	orange ____?_____
Nearby 1 word	glass ____?_____ orange

Skip-grams

I want a glass of orange juice to go along with my cereal.

Context	Target
orange	juice
orange	glass
orange	my

Choosing +/- 10 random words as target
Goal: learn good word embeddings, not do good at this particular supervised learning problem

[Mikolov et. al., 2013. Efficient estimation of word representations in vector space.]

Model

Vocab size = 10,000k

x → y

Context c ("orange") → Target t ("juice")

$O_c \to E \to e_c \to$ ● $\to \hat{y}$

softmax

Softmax: $p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10.000} e^{\theta_t^T e_c}}$ θ_t = parameter associated with output t

$\mathcal{L}(\hat{y}, y)$ =-$\sum_{i=1}^{10.000} y_i log \hat{y}_i$

$$y = \begin{bmatrix} 0 \\ \dots \\ 1 \\ \dots \\ \dots \\ 0 \end{bmatrix}$$ 4834

Problems With Softmax Classification

$$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$$

Computational cost. Solutions:
- Hierarchical softmax classifier
  - Not using perfectly balanced tree (frequent words on top)

How to sample the context c?

Frequently occurring words: the, of, a, and, to, …
Non frequently occurring words: orange, apple, durian, …

$p(c)$: in practice is not entirely uniformly random but distributed according to some heuristic

Defining a New Learning Problem

I want a glass of orange juice to go along with my cereal.

	Context	Word	Target
	orange	juice	1
k	orange	king	0
	orange	took	0
	orange	the	0
	orange	of	0

k = 5-20 for smaller datasets
k = 2-5 for larger datasets

[Mikolov et. al., 2013. Distributed representation of words and phrases and their compositionality]

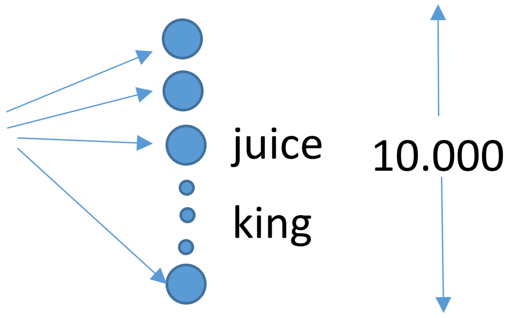
Model (Negative Sampling)

Softmax: $$p(t|c) = \frac{e^{\theta_t^T e_c}}{\sum_{j=1}^{10,000} e^{\theta_j^T e_c}}$$

$$P(y = 1 \mid c, t) = \sigma(\theta_t^T e_c)$$

Orange (6257):

$$O_{6257} \to E \to e_{6257}$$



Negative sampling k

X		Y
context	word	target?
orange	juice	1
orange	king	0
orange	book	0
orange	the	0
orange	of	0
c	t	y

Instead of training 10.000 sofmax we have 10.000 binary classifications and on every iteration we train only k+1 of them

Sentiment Classification Problem

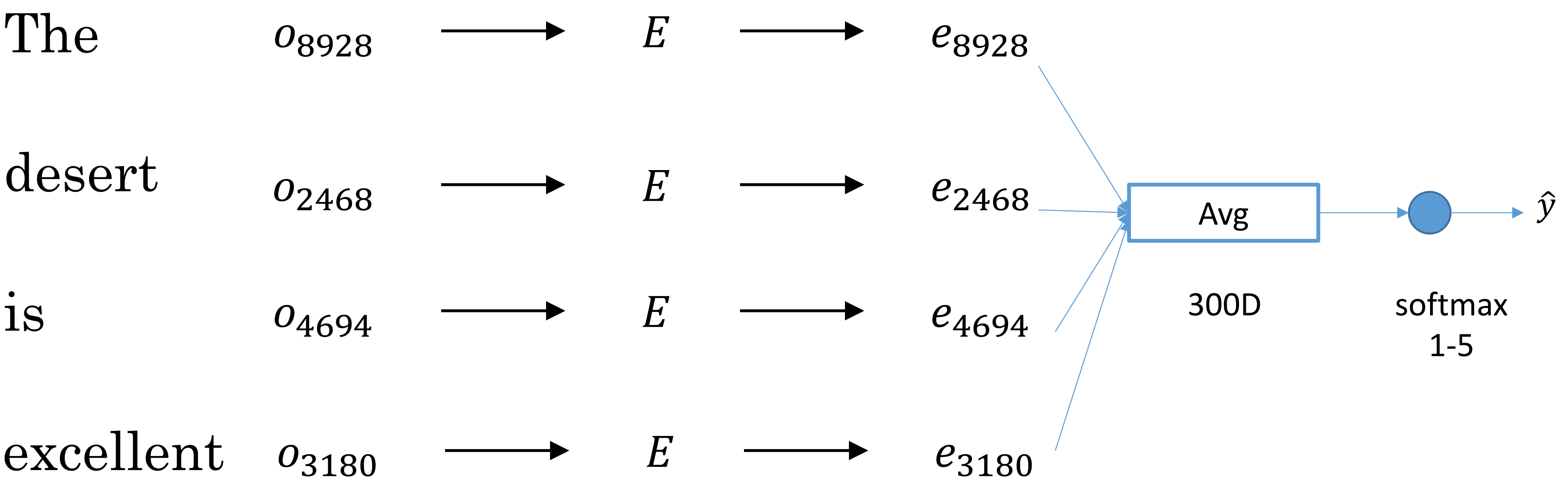
x	y
The dessert is excellent.	★★★★☆
Service was quite slow.	★★☆☆☆
Good for a quick meal, but nothing special.	★★★☆☆
Completely lacking in good taste, good service, and good ambience.	★☆☆☆☆

Even with small datasets we can give good performance because of word embeddings

Simple Sentiment Classification Model

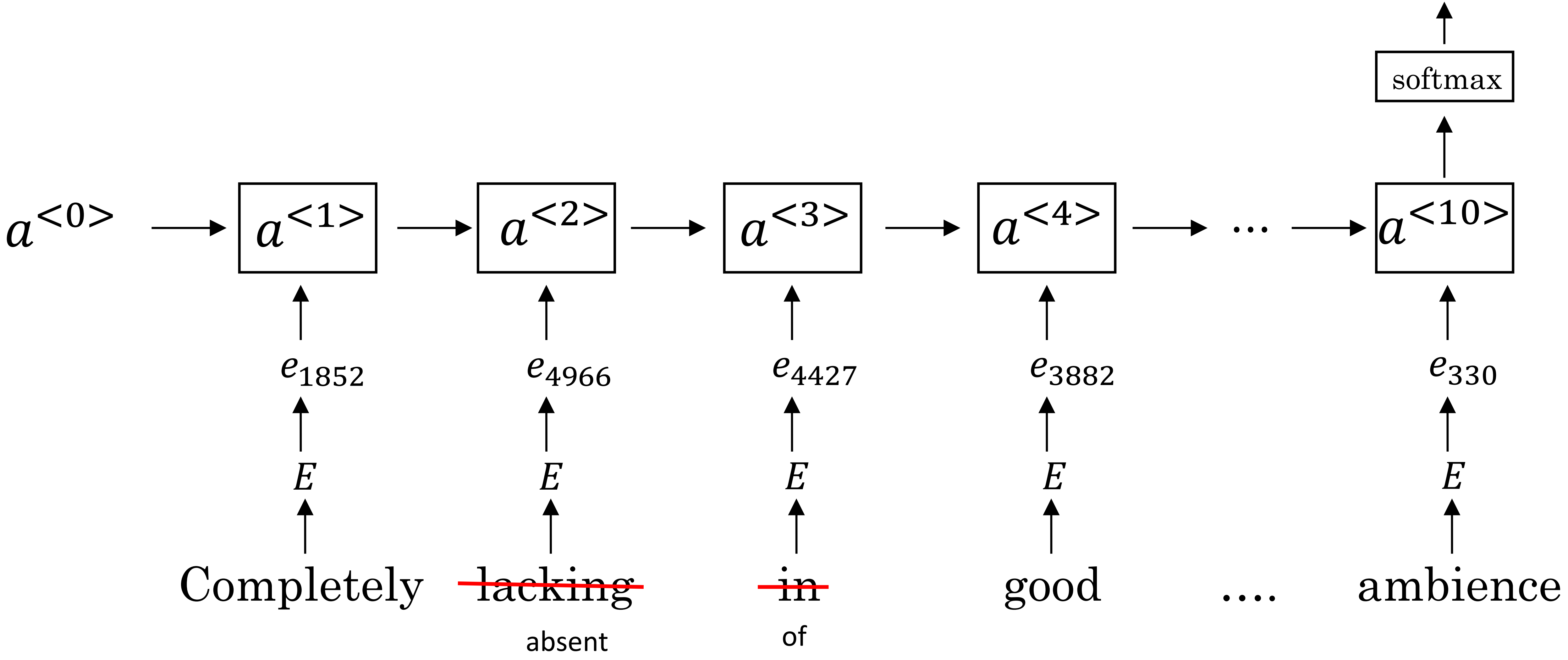
The dessert is excellent ★★★★☆

8928 2468 4694 3180



"Completely lacking in good taste, good service, and good ambience."

RNN for Sentiment Classification



The Problem of Bias in Word Embeddings

Man:Woman as King:Queen

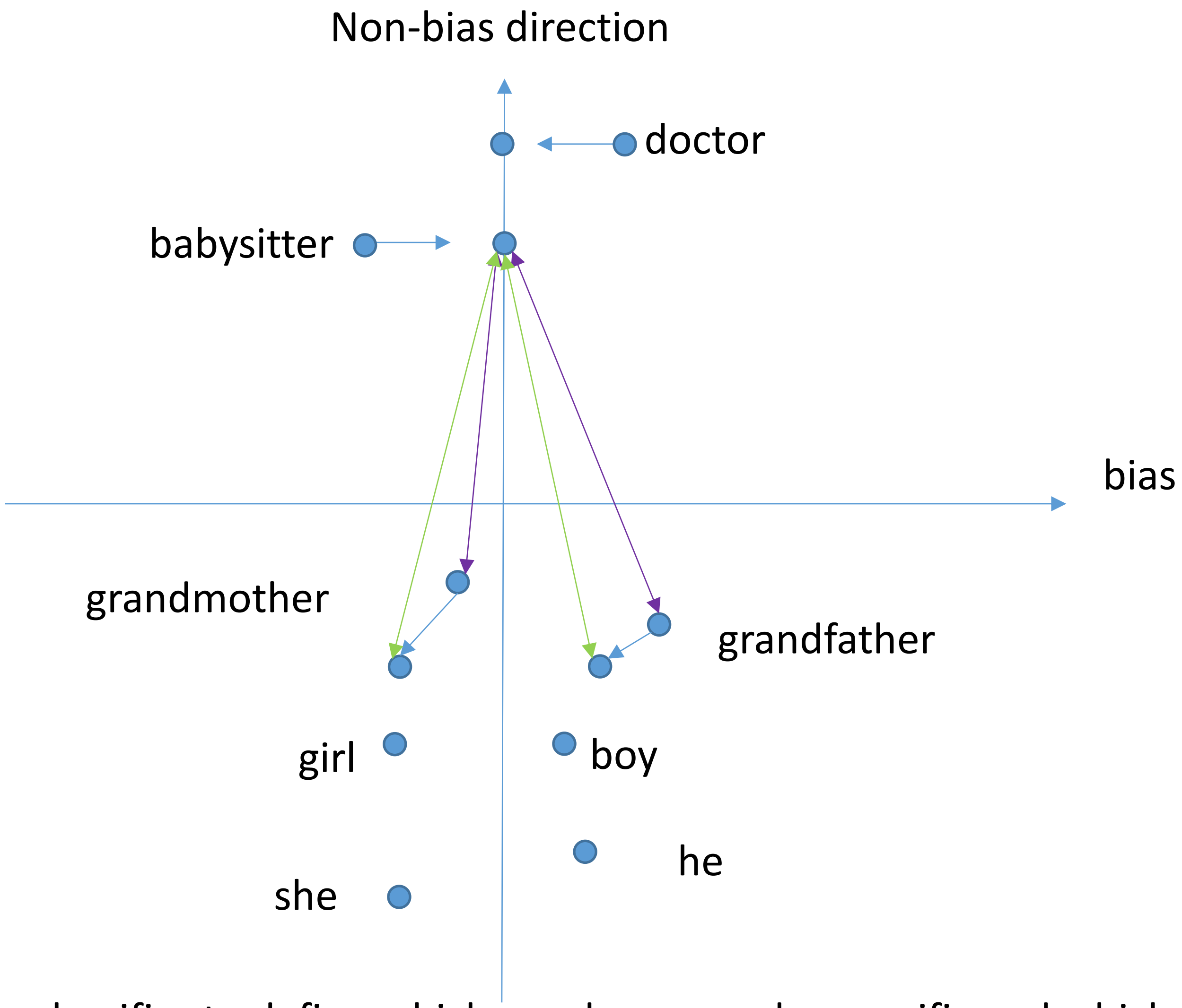
Man:Computer_Programmer as Woman: Homemaker

Father:Doctor as Mother: Nurse

Word embeddings can reflect gender, ethnicity, age, sexual orientation, and other biases of the text used to train the model.

[Bolukbasi et. al., 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings]

Addressing Bias in Word Embeddings



1. Identify bias direction.

$$\left.\begin{array}{c} e_{he} - e_{she} \\ e_{male} - e_{female} \\ \ldots\ldots \end{array}\right\}$$

average

2. Neutralize: For every word that is not definitional, project to get rid of bias.

3. Equalize pairs.

Grandmother – grandfather
Girl - boy

Train a classifier to define which words are gender specific and which are not

Resources Used

- Deeplearning.ai by Andrew Ng