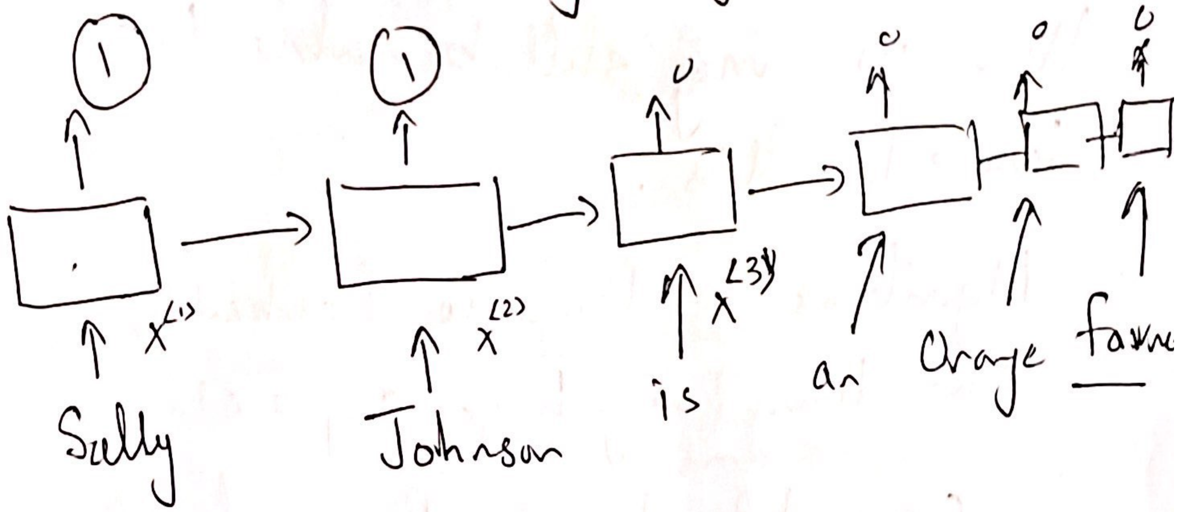
NAMED Entity Recognition



Recognizing that Sally Johnson is a
a name and not a corporation

Robert Lin is an apple farmer

Robert Lin is also a name
as orang and apple are related (similar)
Encodings will help.

What if we see some thing different

Robert Lin is a durian cultivator

durian is a unique fruit

What if dataset is small

If we have learned embedding
then we may still be able to
~~lean~~ solve this.

Algorithms to learn word embedding
can learn very large corpuses.

e.g. 1B to 100 B words.

orange → durian
farmer → cultivator } Similar

very large corpus of unlabelled text

By examining unlabelled text we
can find similarity

We can apply word embedding
training set to the ~~word em~~ named
entity training set which is much
smaller

→ 100 k words.

So word embeddings can be used for transfer Learning as well.

1- Learn word embeddings from Large text- corpus (1-100B words)
(Or download pre-trained embeddings online)

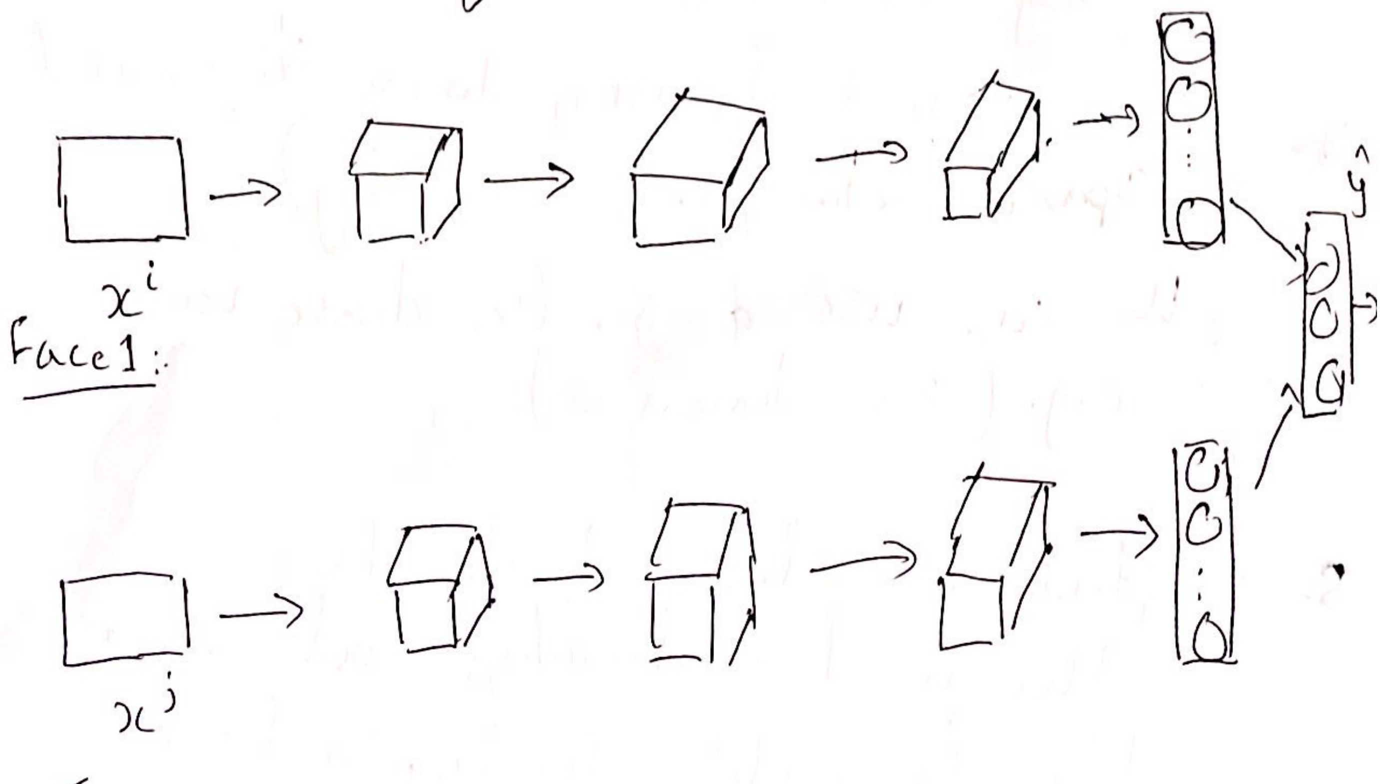
2- Transfer embeddings to new task with smaller training set.
(Say, 100K words)

~~3-~~ instead of using 10,000 dimensional Sparse vector (1-hot encoding)
We can use a smaller dense vector e.g. (300-dimension)

3- Optional: Continue to fine tune the word embeddings with new data. (Possible for big datasets)

- Named Entity Recognition
- Text Summarization
- Co-reference Resolution
- Not much useful for Machine translation or Language modeling.

Word Embedding are also useful for face Encoding:



Face 2:

For face recognition, take any input

for face recognition, take any input (VI)
any face picture and have a neural
network do encoding

for word embedding, we have a ...
fixed vocabulary say 10,000 words
and we'll learn a vector e_1 through
$e_{10,000}$ that learns a fixed coding/embedding
for each of the words

embeddings and encodings may be used
interchangeably.

fixed vocabulary vs unlimited pictures/faces.

Conclusion:

Word Embedding $\longrightarrow$ Transfer Learning.

Embeddings As Anologies | Properties of Word Embedding.

	Man (5391)	Woman (9853)	King (4914)	Queen (7157)	Apple (456)	Orange (6257)
Gender	-1	1	-0.95	0.97	0	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

Man → Woman as King → ?

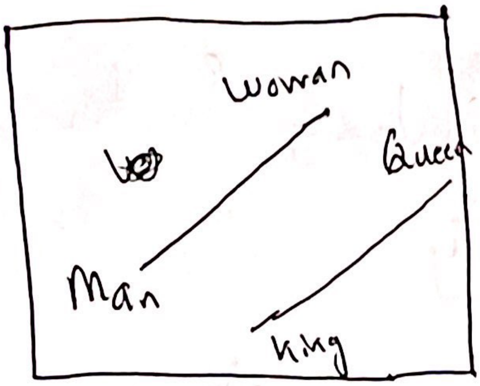
Let's suppose a 4-D vector is used

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

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

We can draw anologies

Compute $e_{man} - e_{woman}$ and Compute e_{king} − Other words to determine



300D

Vector difference b/w man and Woman is similar to Vector difference b/w King and Queen

Find word w:- $\underset{w}{\arg\max}\ sim(e_w, e_{King} - e_{man} + e_{woman}$

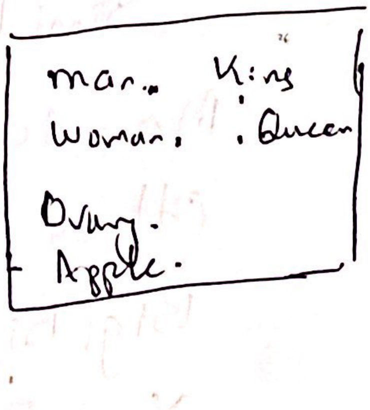
Find word w:- $\underset{w}{\arg\max}\ Sim(e_w, e_{King} - e_{man} + e_{woman})$

The above figure is good for representation in 300 dimensions.

If we transform this to 2-D (using t-SAE)

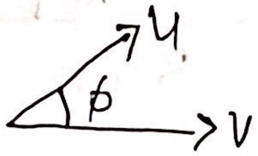
the ~~transformation~~ difference will not be obvious

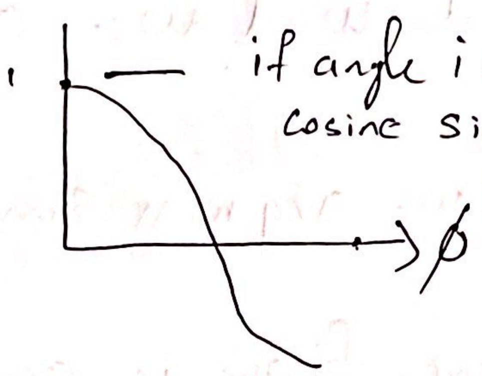
i.e, this is non-linear in 2-d.



The most common similarity function is cosine similarity

$$\text{Sim}(u,v) = \frac{u^T v}{\|u\| \|v\|}$$





if angle is 0
cosine similarity is 1

Euclidean distance can also be used
Main difference between the normalization approaches.

Similarities can be learned

Man : Woman as Boy : Girl
Ottawa : Canada as Nairobi : Kenya
Big : Bigger as Tall : Taller
Yen : Japan as Ruble : Russia