Seminar in Deep Neural Networks

NLP: Embeddings

Presenter: Anuj Pareek

What are word embeddings?

Complete vocabulary:

$$\mathcal{V} = \{elephant, monkey, zebra\}$$

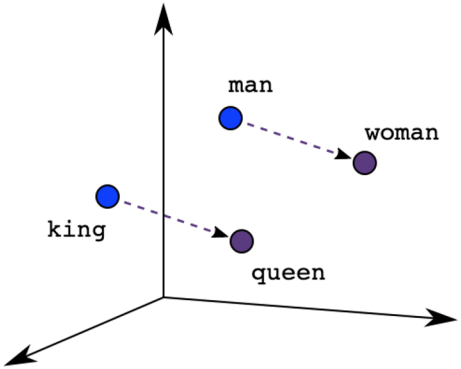
Easy! Represent words as one-hot vectors:

$$w_{elephant} = [1, 0, 0]$$
$$w_{monkey} = [0, 1, 0]$$
$$w_{zebra} = [0, 0, 1]$$

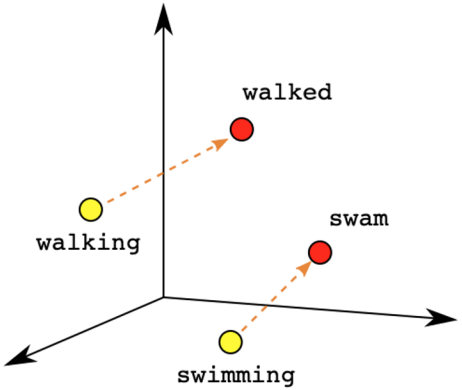
Problems! No notion of similarity/dissimilarity. They are orthogonal vectors

No contextual information in encoding!

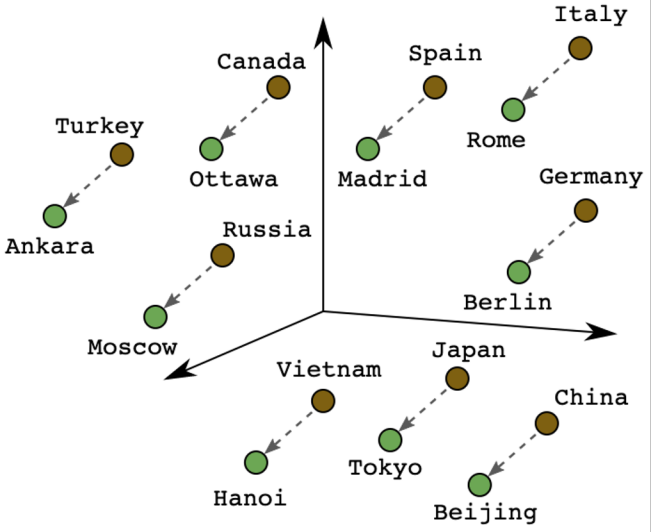
Motivation for word embeddings



Male-Female

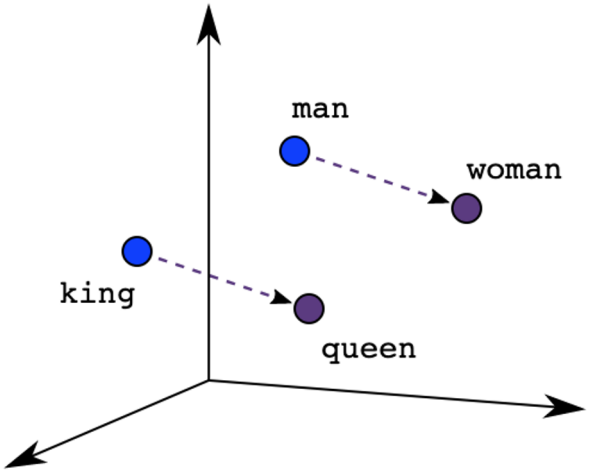


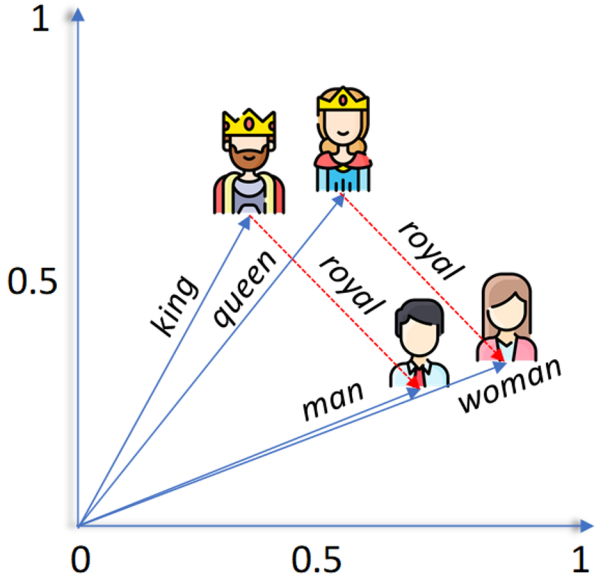
Verb Tense



Country-Capital

Motivation for word embeddings





NLP preliminaries: Bag-of-Words model

Count the word occurrences:

Document	the	cat	sat	in	hat	with
the cat sat	1	1	1	0	0	0
the cat sat in the hat	2	1	1	1	1	0
the cat with the hat	2	1	0	0	1	1

Sort of similar to one-hot encoding, but includes the counts, rather than binary.

A “count-based” model.

Mikolov et al., Efficient Estimation of Word Representations in Vector Space (Word2Vec)

Objective: *Compute continuous vector representations of words from large data sets...*

...with the expectation that not only will similar words tend to be close to each other, but that words can have multiple degrees of similarity

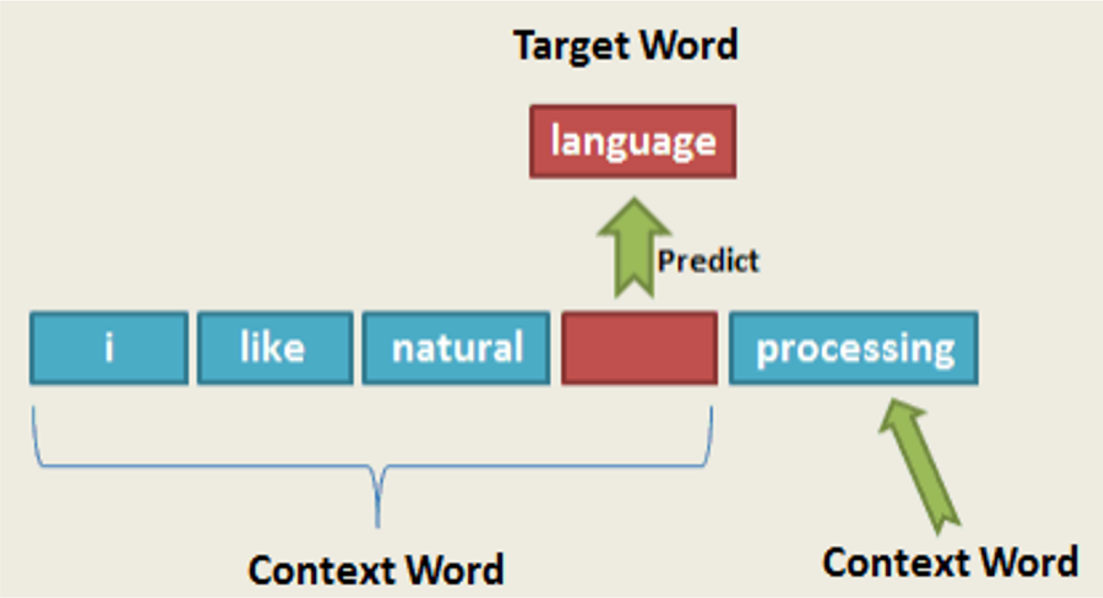
Two models in Word2Vec:

- **Continuous Bag-of-Words model**
- **Skip-gram model**

Word2Vec: Continuous Bag-of-Words model

Main idea: Create word embeddings by learning to predict target word from context words.

Example:



Word2Vec: Continuous Bag-of-Words model

Step 1:
Slide window over text.

Step 2:
Encode input and output one-hot vectors.

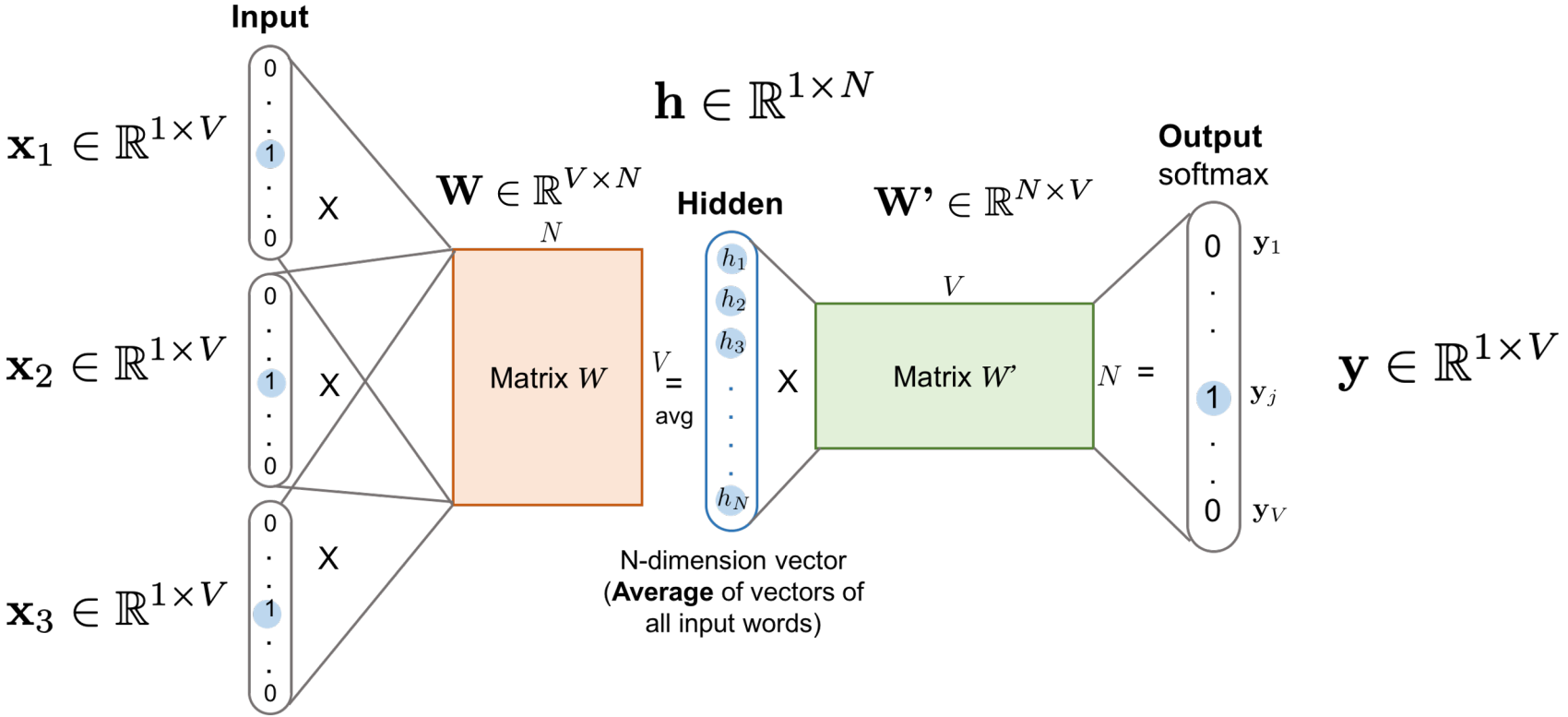
Vocabulary: $|\mathcal{V}| = V$

Input vectors: $\mathbf{x}_1 \in \mathbb{R}^{1\times V}, \mathbf{x}_2 \in \mathbb{R}^{1\times V}, \mathbf{x}_3 \ldots$

Output is single row vector: $\mathbf{y} \in \mathbb{R}^{1\times V}$

Word2Vec: Continuous Bag-of-Words model

Step 3:
Setup the model:



Note: W' is not the transpose of W. It's just "bad notation" in Word2Vec

Diagram: Lilian Weng

Word2Vec: Continuous Bag-of-Words model

Step 4:
We don't have **W, h** or **W'**.

h-vectors are embedded word vectors in N-dimensional space.

"Force" model to make **h**:

Maximize: $P(\text{target word}|\text{context words})$

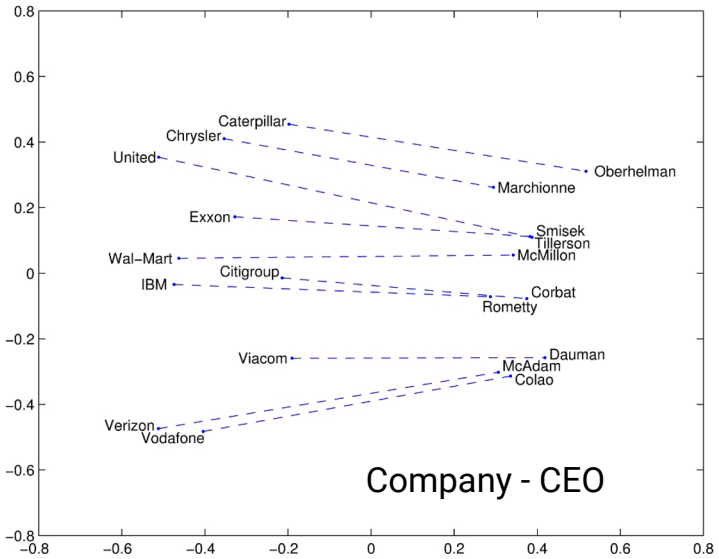
Corresponds to minimizing loss: $J = -logP(\text{target word}|\text{context words})$

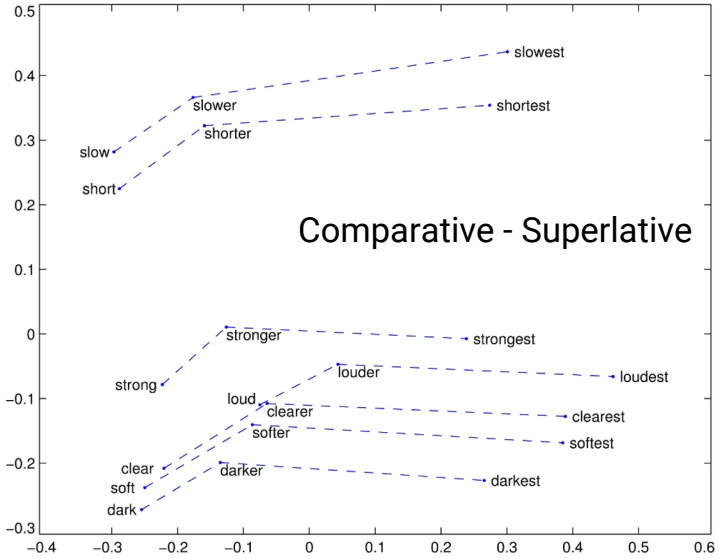
Loss *J* minimized through gradient descent, by updating parameters **W** and **W'**.

Pennington et al., GloVe: Global Vectors for Word Representation

Objective: *Propose a specific weighted least squares model that trains on **global** word-word co-occurrence counts and thus makes efficient use of statistics…*
...produce a word vector space with meaningful substructure

Sub-structure:





GloVe: Global Vectors for Word Representation

Make word-word co-occurrence matrix ***X***.

Example with window-size of 2:

1. I enjoy flying.
2. I like NLP.
3. I like deep learning.

The resulting counts matrix will then be:

$$X = \begin{array}{c} \\ I \\ like \\ enjoy \\ deep \\ learning \\ NLP \\ flying \\ . \end{array} \begin{array}{c} \begin{array}{cccccccc} I & like & enjoy & deep & learning & NLP & flying & . \end{array} \\ \begin{bmatrix} 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$$\mathbf{X} \in \mathbb{R}^{V \times V}$$

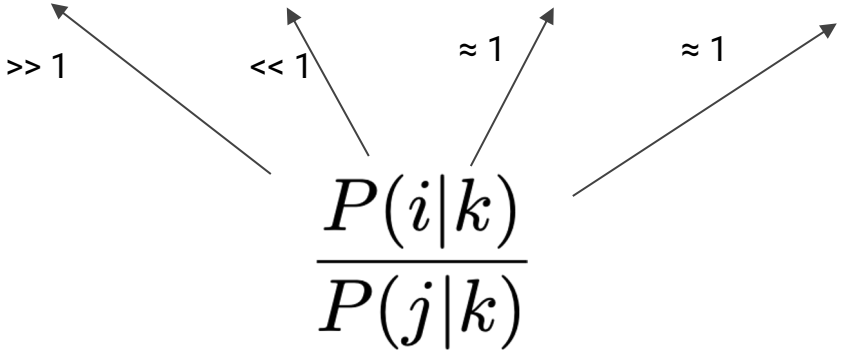
GloVe: Global Vectors for Word Representation

The probability that word ***j*** appear in context of word ***i*** is: $P(j|i)$

Relationship between ***i*** and ***j*** checked with co-occurrence ***ratios***, with probe words ***k***

Example:

i = ice, j = steam k_1 = solid, k_2 = gas, k_3 = water, k_4 = fashion



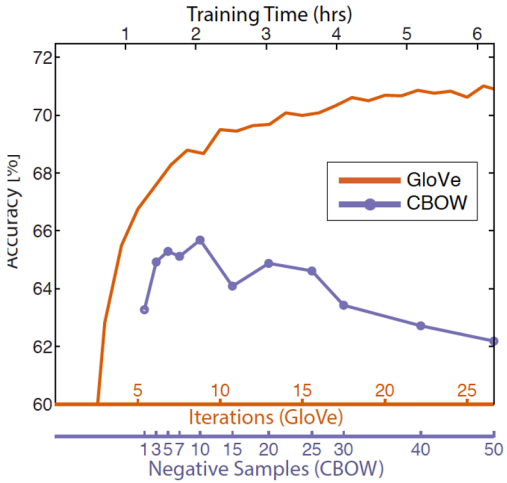
Co-occurrence ratio able to distinguish relevant words; solid (>>1) and gas (<<1) from irrelevant words; water and fashion (≈ 1) and discriminate between two relevant words.

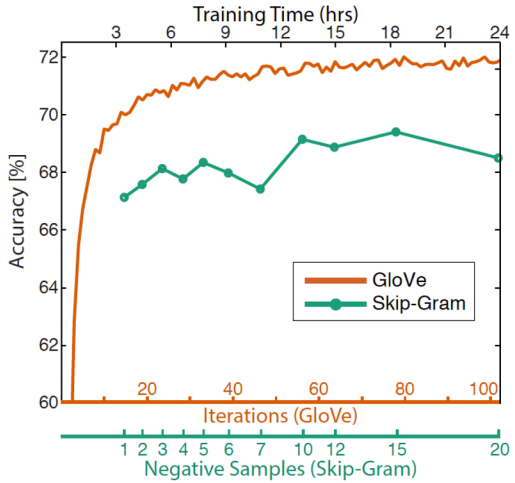
GloVe: Global Vectors for Word Representation

Train model parameters using co-occurrence ratios to create “global context” word embeddings.

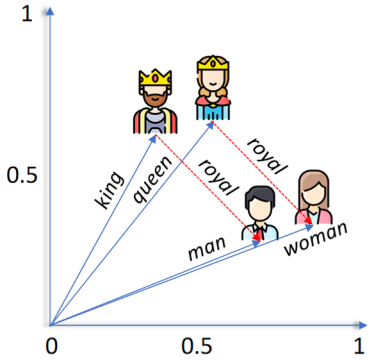
General form of model F: $$F(w_i, w_j, \tilde{w}_k) = \frac{P(i|k)}{P(j|k)}$$

Comparison to Mikolov et al.’s CBOW and Skip-gram on **word analogy task**:





A is to A as B is to ____*

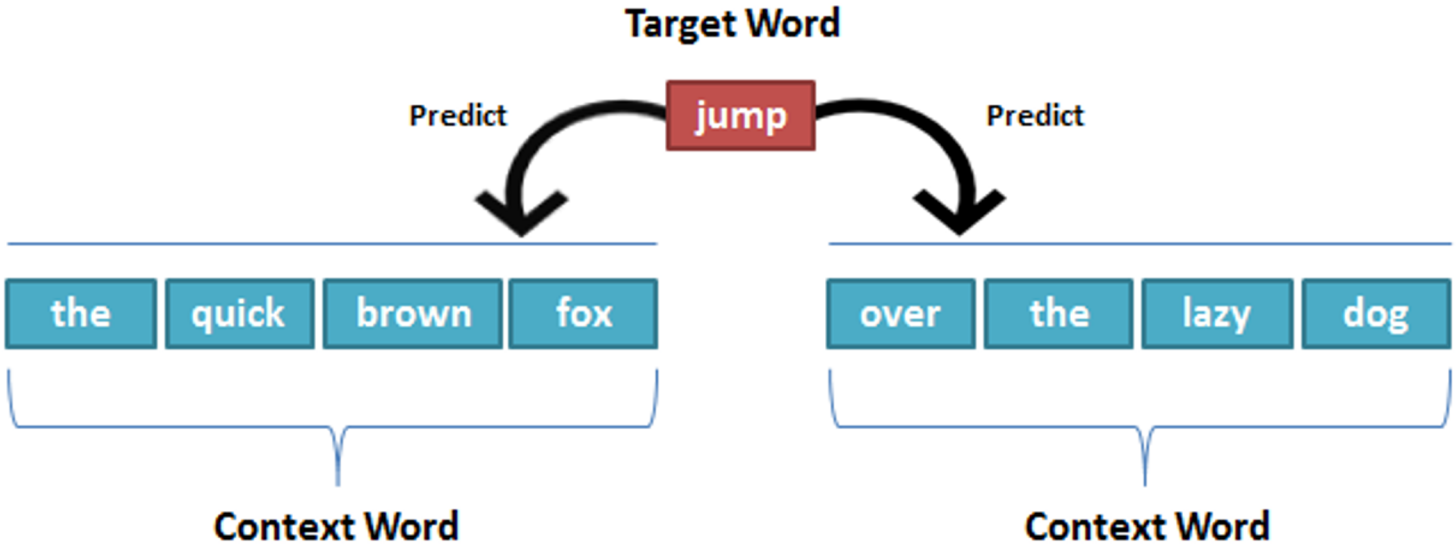


$$w_A - w_{A*} + w_B =$$

Word2Vec: **Skip-gram model**

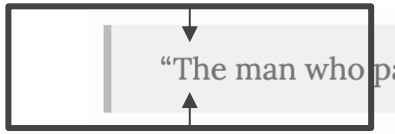
Main idea: Create word embeddings by learning probability distribution of context words from center word.

Example:



Word2Vec: Skip-gram model

Step 1:
Sliding window of size = 5 on training data:



> "The man who passes the sentence should swing the sword." – Ned Stark

Sliding window (size = 5)	Target word	Context
[The man who]	the	man, who
[The man who passes]	man	the, who, passes
[The man who passes the]	who	the, man, passes, the
[man who passes the sentence]	passes	man, who, the, sentence
...	...	...
[sentence should swing the sword]	swing	sentence, should, the, sword
[should swing the sword]	the	should, swing, sword
[swing the sword]	sword	swing, the

Word2Vec: Skip-gram model

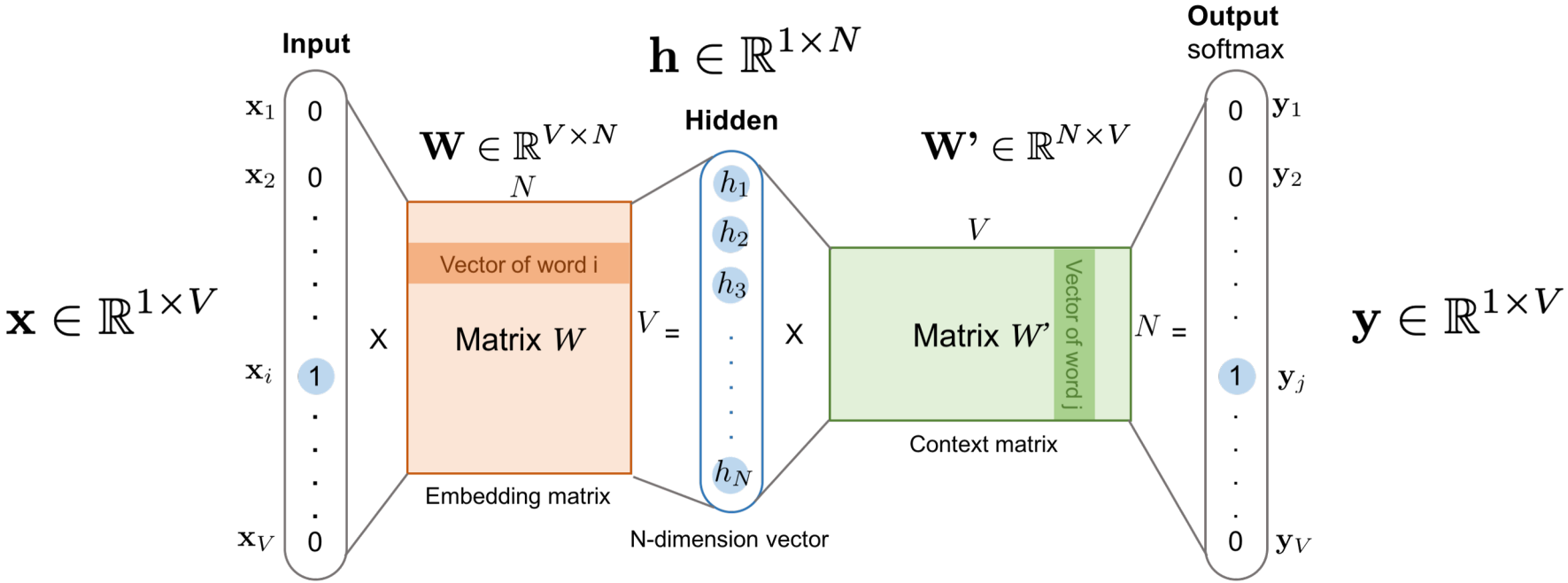
Step 2:
Encode input and output pairs as one-hot vectors.

Size of our vocabulary: $|\mathcal{V}| = V$

Input and output are row vectors: $\mathbf{x} \in \mathbb{R}^{1\times V}, \mathbf{y} \in \mathbb{R}^{1\times V}$

Word2Vec: Skip-gram model

Step 3:
Setup the model:



Note: W' is not the transpose of W. It's just "bad notation" in Word2Vec

Diagram: Lilian Weng

Word2Vec: Skip-gram model

Step 4:

h-vectors are embedded word vectors.

“Force” model to make **h**:

Maximize: $P(\text{context words}|\text{center word})$

Corresponds to minimizing loss: $J = -logP(\text{context words}|\text{center word})$

Loss *J* minimized with gradient descent, by updating parameters **W** and **W’**.

Cer et al., Universal Sentence Encoder (USE)

Objective: *Present models for encoding sentences into embedding vectors...*

...compute context aware representations of words in a sentence that take into account both the ordering and identity of all the other words.

Two models:

- **Transformer architecture model** (our focus)
- **Deep Averaging Network**

Universal Sentence Encoder (USE)

Why not separately embed **words** in a sentence?

Problem 1:
Common words increases similarity:

“*It must be true*” vs. “*He must be taking it to the car wash*”

Problem 2:
Swapping word order doesn’t change similarity:

“*Is this garbage?*” vs. “*This is garbage*”

USE: Transformer architecture model

Takes as input a sentence of arbitrary length, outputs 512-dimensional embedding vector.

Use **multi-task learning**!

Intuition; capture most informative features and discard noise

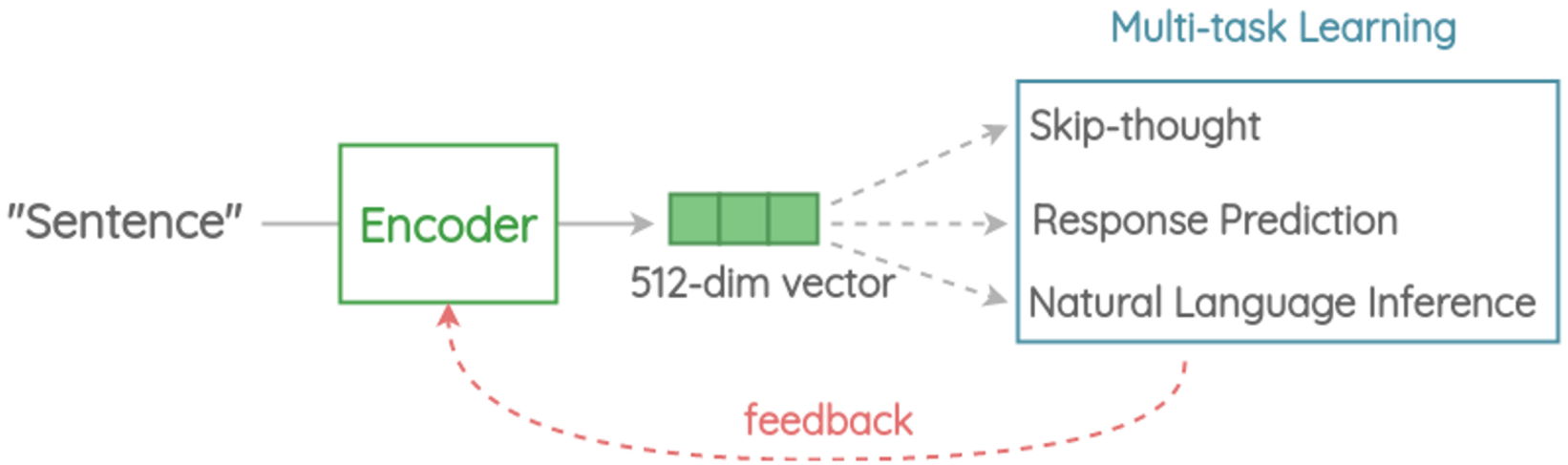
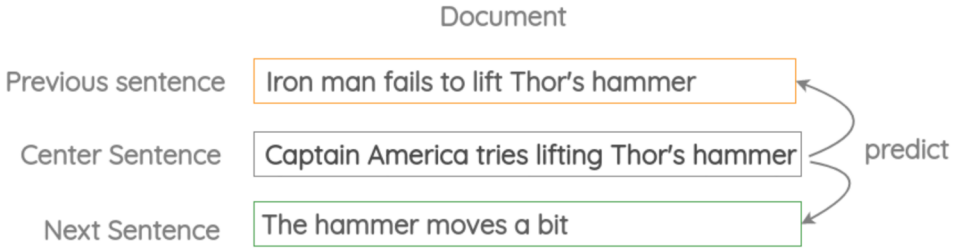


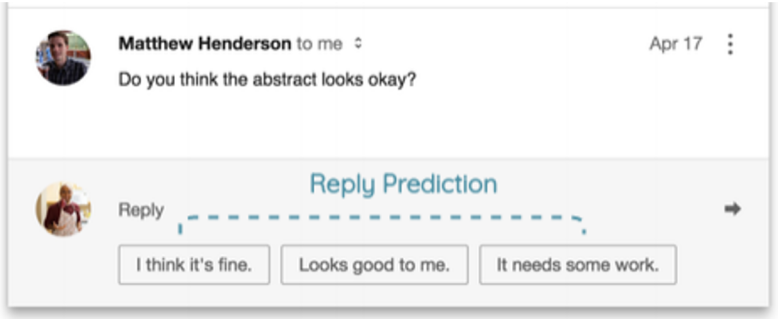
Diagram: Amit Chaudhary

USE: The multiple tasks

Skip-thought:



Conversational Response Prediction:



NLP / Text Classification:

"How old are you?"

"What is your age?"

"My phone is good."

→

Confidence is a question

(96%) "How old are you?"

(98%) "What is your age?"

(7%) "My phone is good."

USE: Transformer architecture model

Transformer architecture uses **self-attention**.

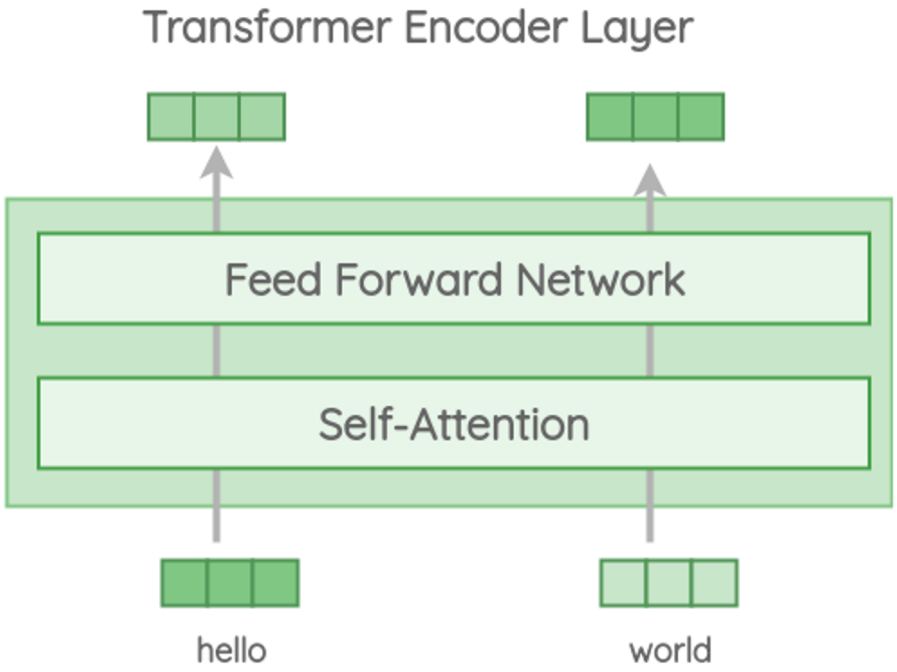
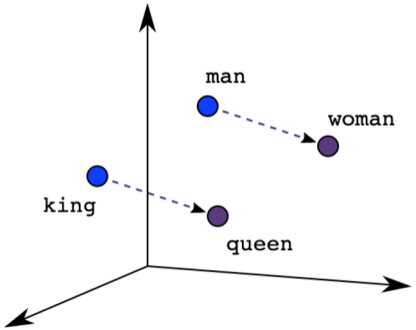
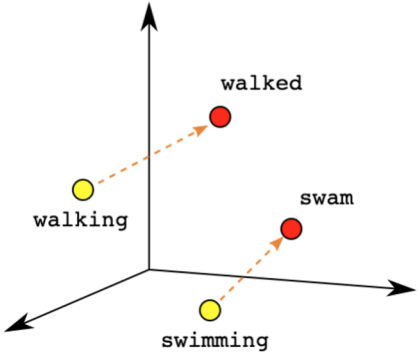


Diagram: Amit Chaudhary

Summary

- Convert free-text language into numeric values for NLP
- One-hot vectors are simple, but no contextual information
- Contextualized word embeddings capture similarity/dissimilarity in N dimensions.
- Setup model with proper task and objective function
- Continuous Bag-of-Words, GloVe and Skip-gram for word embeddings
- USE Transformer architecture for sentence embeddings





Thank you!

- E-mail: apareek@student.ethz.ch
- Thanks to mentor, Zhao Meng

References:

Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. 1st Int Conf Learn Represent ICLR 2013 - Work Track Proc 2013:1–12.

Bengio Y, Réjean D, Pascal V, Christian J. A Neural Probabilistic Language Model. J Mach Learn Res 2003;3:1137–116.

Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed Representations of Words and Phrases and their Compositionality. EMNLP 2016 - Conf Empir Methods Nat Lang Process Proc 2013:1389–99. https://doi.org/10.18653/v1/d16-1146.

Pennington J, Socher R, Manning CD. GloVe: Global vectors for word representation. EMNLP 2014 - 2014 Conf Empir Methods Nat Lang Process Proc Conf 2014:1532–43. https://doi.org/10.3115/v1/d14-1162.

Cer D, Yang Y, Kong S yi, Hua N, Limtiaco N, St. John R, et al. Universal sentence encoder for English. EMNLP 2018 - Conf Empir Methods Nat Lang Process Syst Demonstr Proc 2018:169–74. https://doi.org/10.18653/v1/d18-2029.

Eisenstein J. Introduction to Natural Language Processing. Illustrate. The MIT Press; 2019.