Assignment 1

Learning distributed word representations

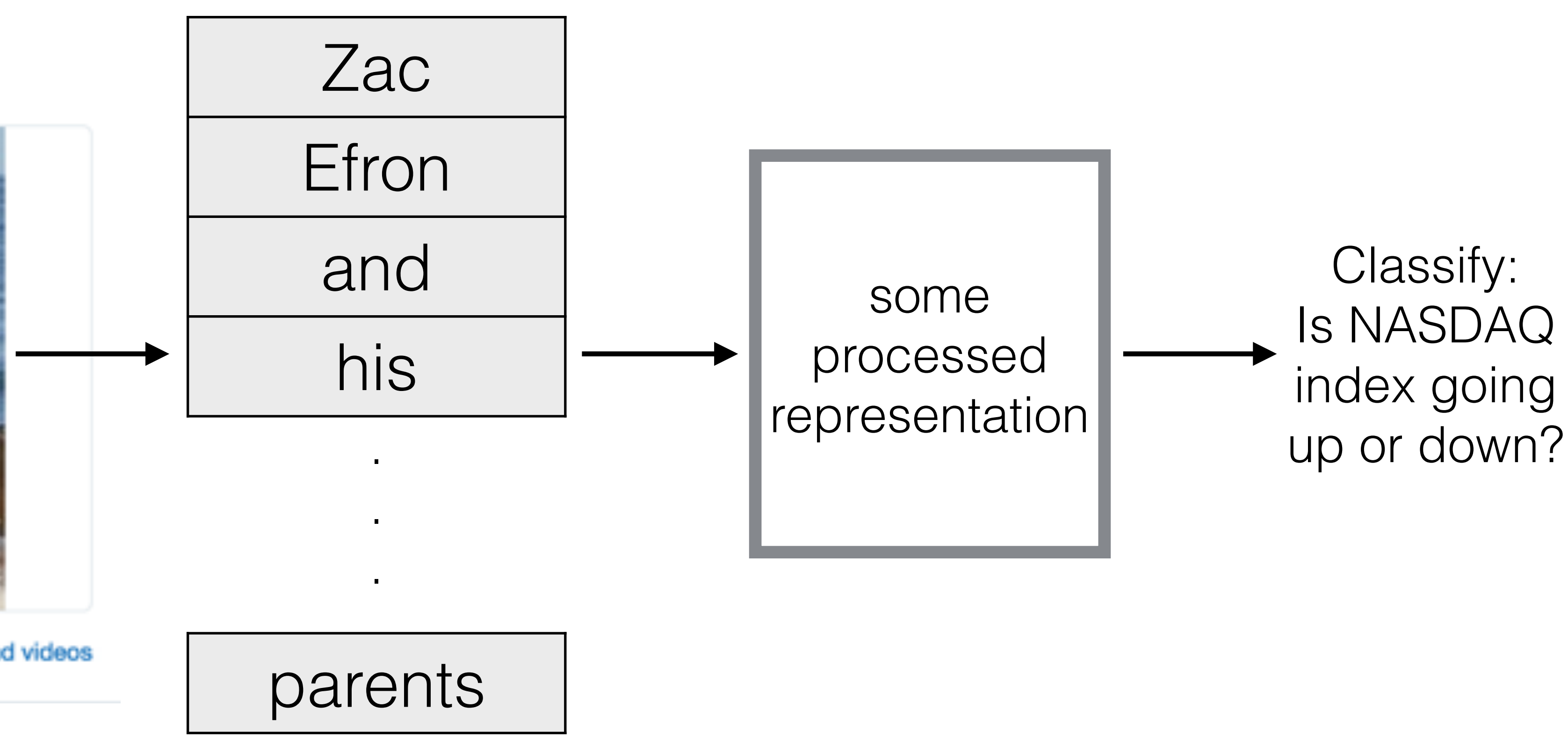
Jimmy Ba
csc321ta@cs.toronto.edu

Background

- Text and language play central role in a wide range of computer science and engineering problems
- Applications that depend on language understanding/processing includes: speech processing, search/query internet, social media, recommendation system, artificial intelligence and many others

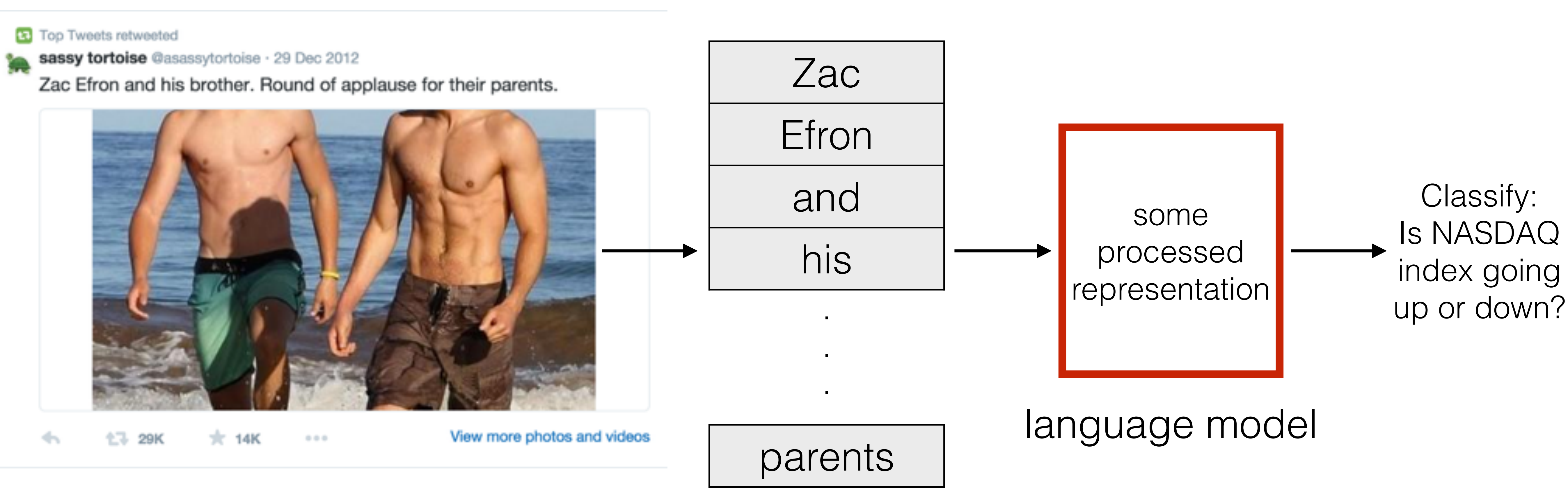
Motivation

- Getting meaningful representations from text data are often the key component in Google search engine or your next big start-up ideas



Motivation

- Getting meaningful representations from text data are often the key component in Google search engine or your next big start-up ideas



Language Model

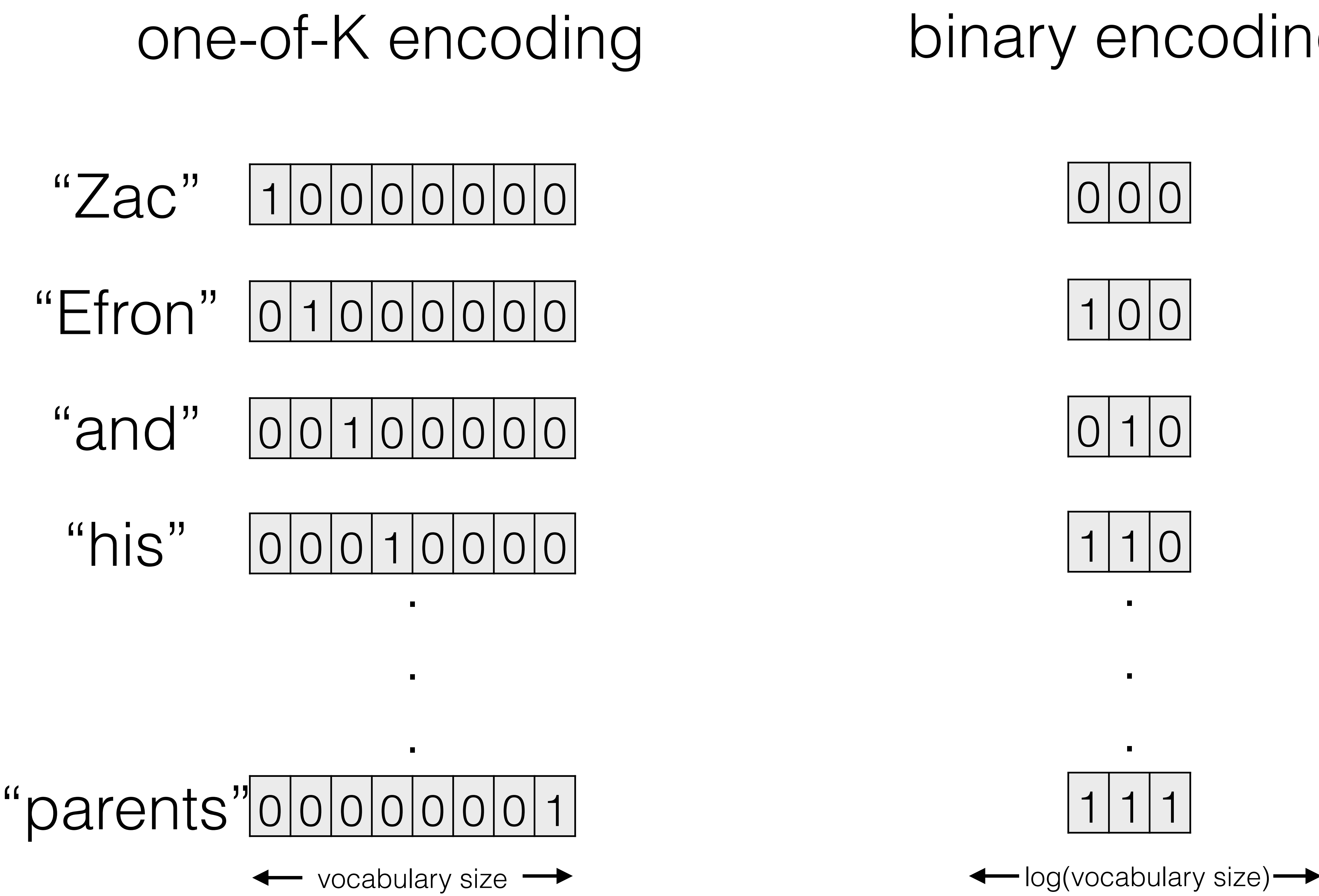
- We need to represent text data in a way that is "easy" for the later stage classification problem or learning algorithms

- "Easy": Be able to handle large scale vocabulary and words have similar syntactic/semantic meaning should be close in the representation space

Language Model

	one-of-K encoding	binary encoding
“Zac”	1 0 0 0 0 0 0 0	0 0 0
“Efron”	0 1 0 0 0 0 0 0	1 0 0
“and”	0 0 1 0 0 0 0 0	0 1 0
“his”	0 0 0 1 0 0 0 0	1 1 0
	. . .	. . .
“parents”	0 0 0 0 0 0 0 1	1 1 1

Language Model



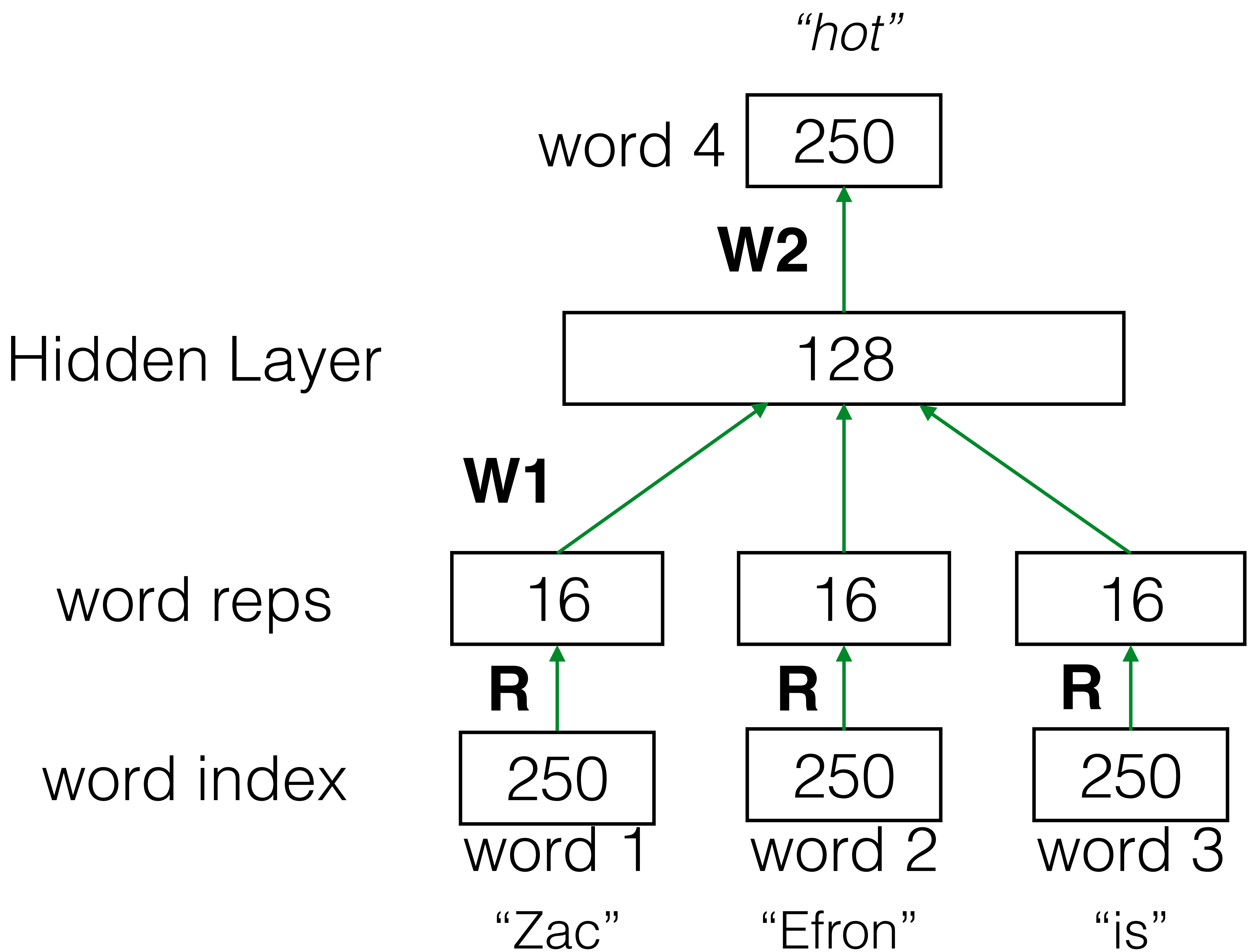
Language Model

	one-of-K encoding								distributed encoding			
"Zac"	1	0	0	0	0	0	0	0	1.5	0.1	-0.1	2.1
"Efron"	0	1	0	0	0	0	0	0	0.7	-0.1	0.3	0.4
"and"	0	0	1	0	0	0	0	0	0.1	1.6	-1.9	1.1
"his"	0	0	0	1	0	0	0	0	3.5	0.2	1.1	-2.5
.				.						.		
.				.						.		
.				.						.		
"parents"	0	0	0	0	0	0	0	1	-2.1	-3.3	-2.7	1.9

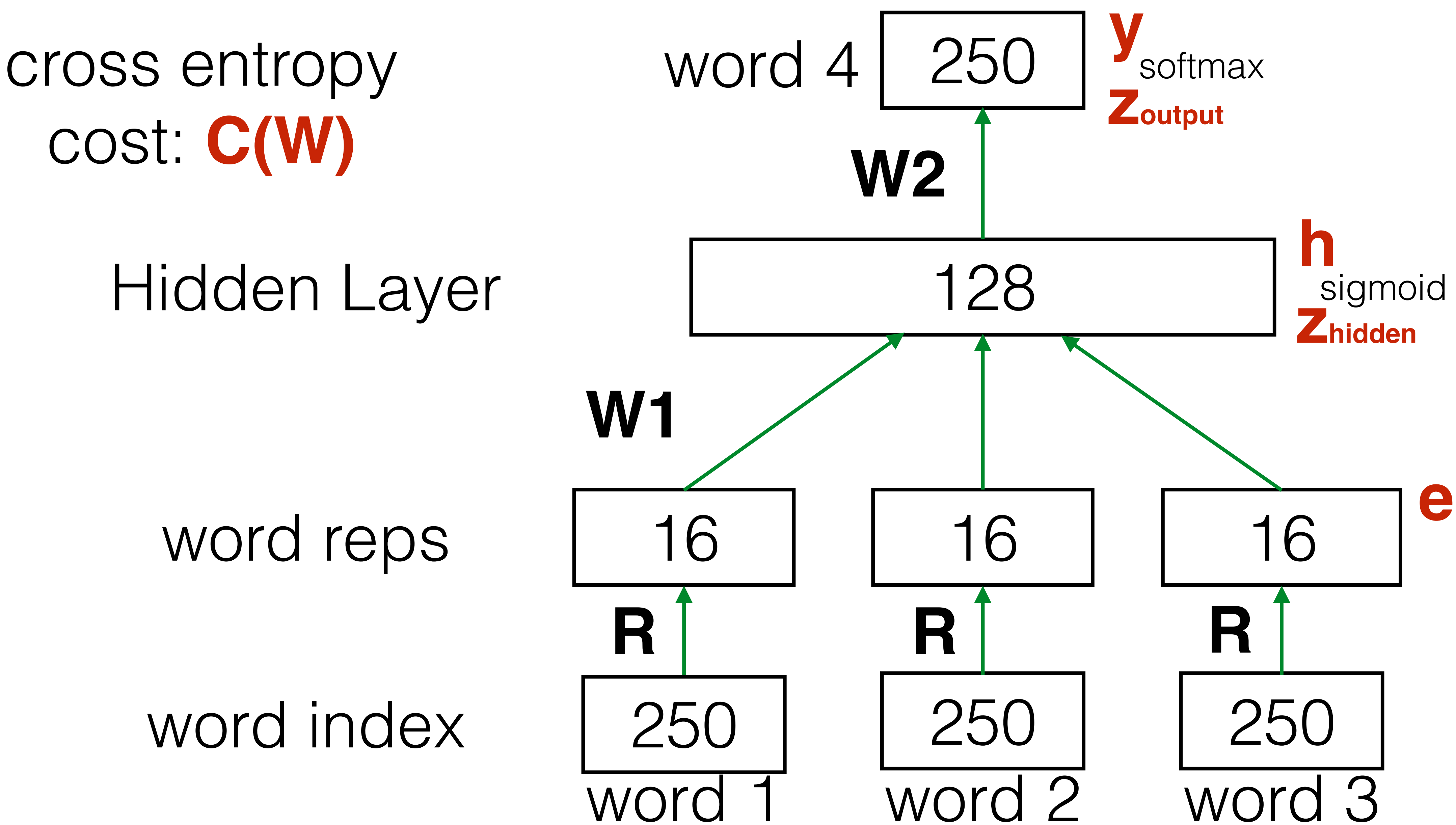
← vocabulary size →

← embedding size (constant) →

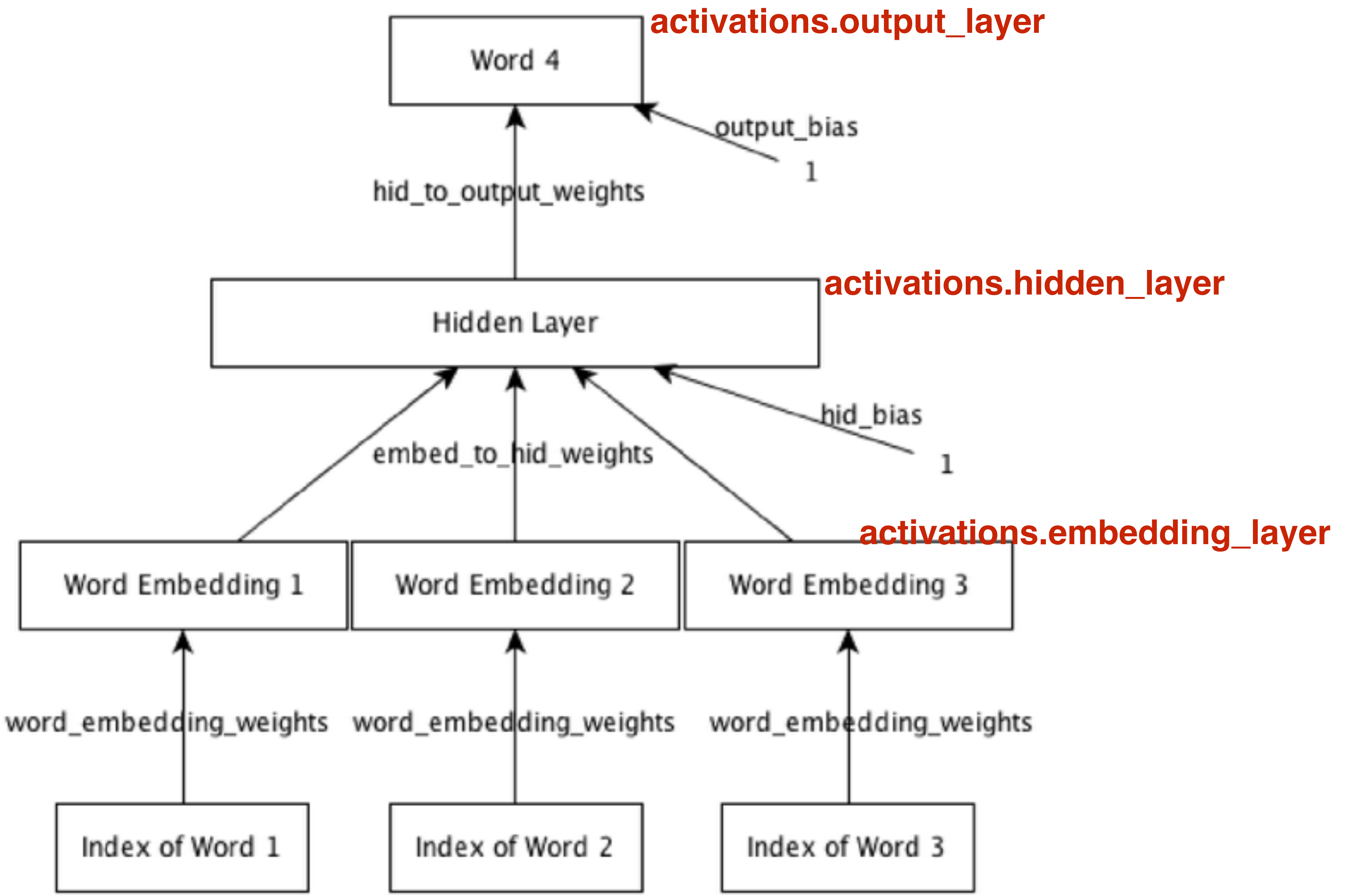
Neural Language Model



Neural Language Model

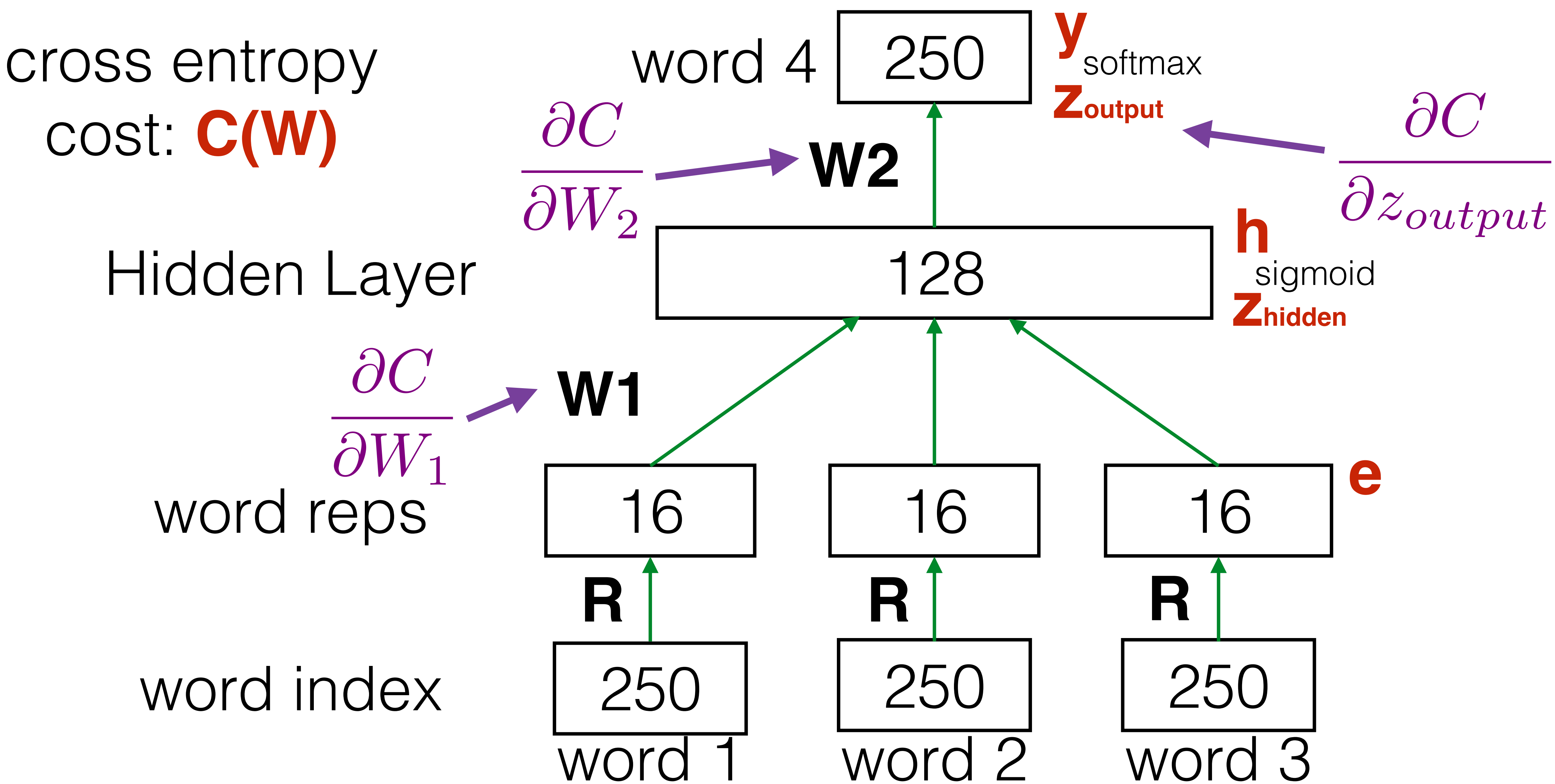


Neural Language Model



Things you need to do in assignment 1

- Part 2

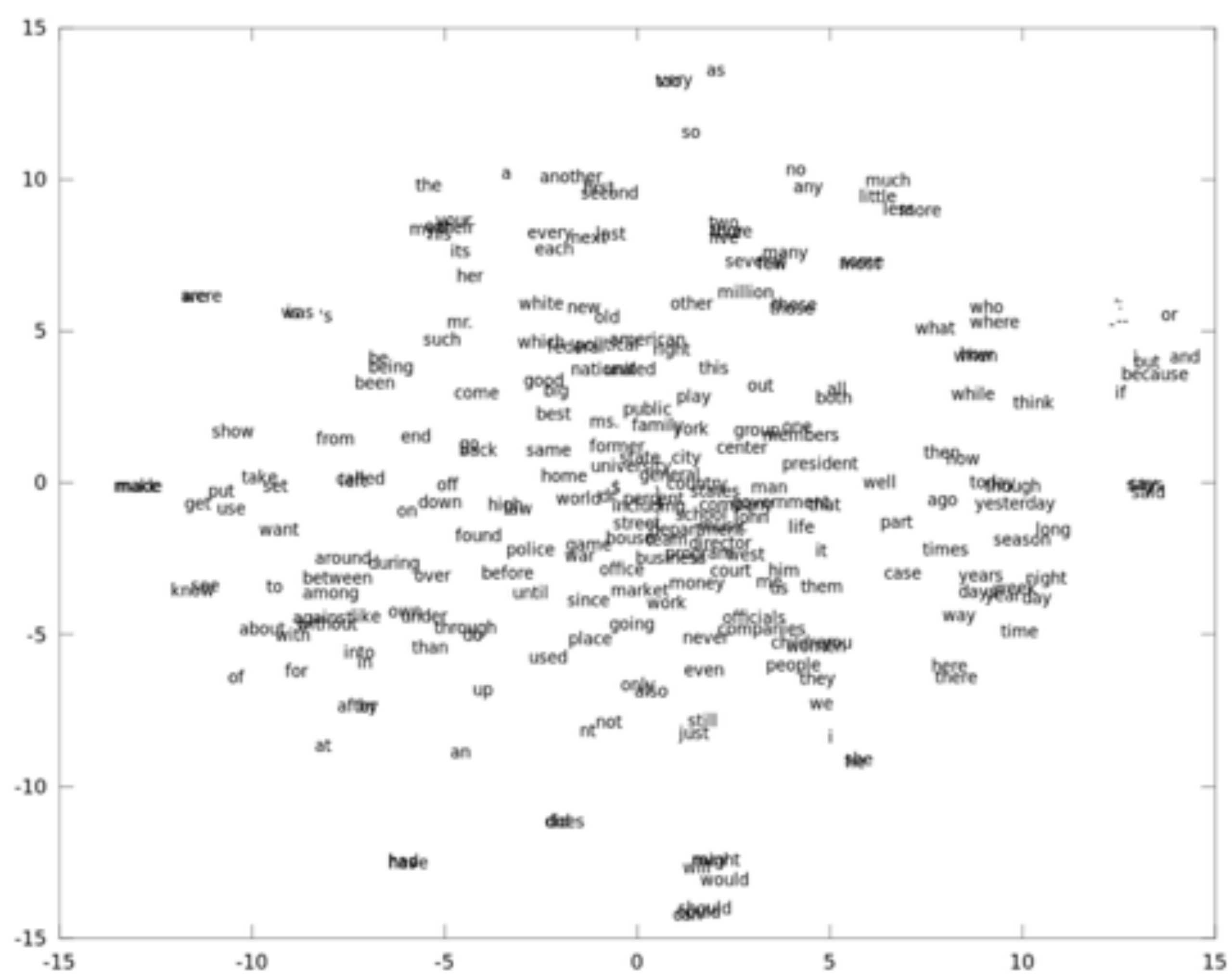


Things you need to do in assignment 1

- Part 2
  - code/derive the partial derivative of cross-entropy cost with respect to softmax input
  - code/derive the gradients of the weight matrix using partial derivatives from backdrop
  - can be done in just 5 simple lines of code and NO for-loops
  - use *checking.check_gradients* to verify the correctness of the 5 lines of code

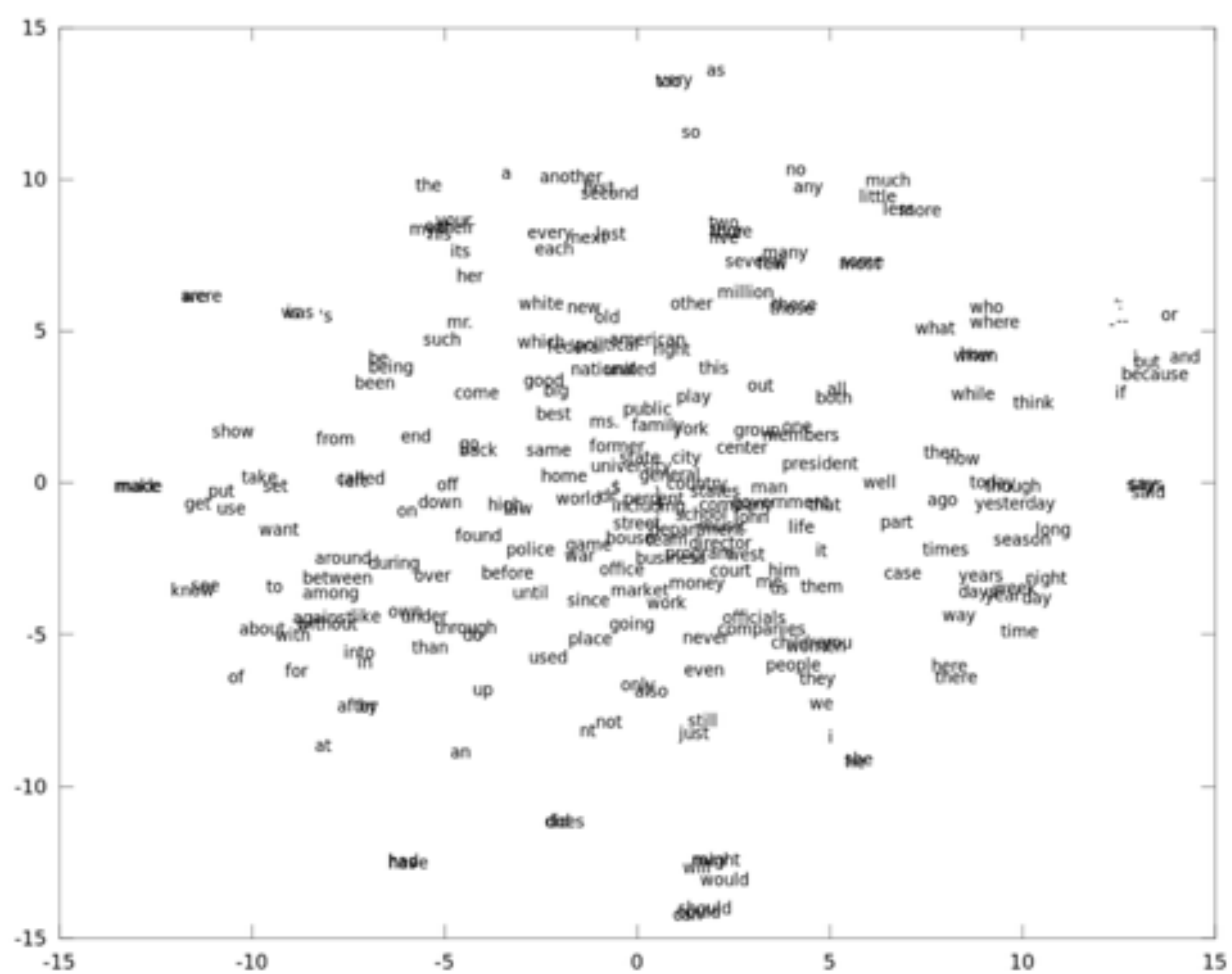
Things you need to do in assignment 1

- Part 3
    - analyze the trained model



t-SNE embedding

- It projects 16D learnt word embedding to 2D for plotting visualization only. (*display_nearest_words*, *word_distance* uses the 16D word embedding)



Word Distance

- It only makes sense to compare the relative distances between words, i.e.
  - distance(A, B) and distance(A, C)
  - distance(A, B) and distance(A, w), distance(B,w)
  - NOT distance(A,B) and distance(C,D)

Things you need to do in assignment 1

- Part 3
  - Think about how the model would put two words close together in embedding space
  - Think about what the task the model is trying to achieve and how that affects the word representation that is being learned.
  - Think about what kind of similarity the nearest words in the 16D embedding space have

Due: Tuesday, Feb. 3

at the start of lecture