Tokenization and Word Embeddings

Danna Gurari
University of Colorado Boulder
Spring 2025

https://dannagurari.colorado.edu/course/neural-networks-and-deep-learning-spring-2025/

Review

- Last week:
  - Machine learning for sequential data
  - Recurrent neural networks (RNNs)
  - Problem: learning challenges
  - Solution: Gated RNNs
  - Programming tutorial
- Assignments (Canvas):
  - Lab assignment 1 grades are out
    - Review session will be held at 4pm today on Zoom
    - Email all regrade requests to our TA, Nick Cooper (a comment in Canvas is not sufficient)
  - Problem set 3 due earlier today
  - Lab assignment 2 due in a 1.5 weeks
- Questions?

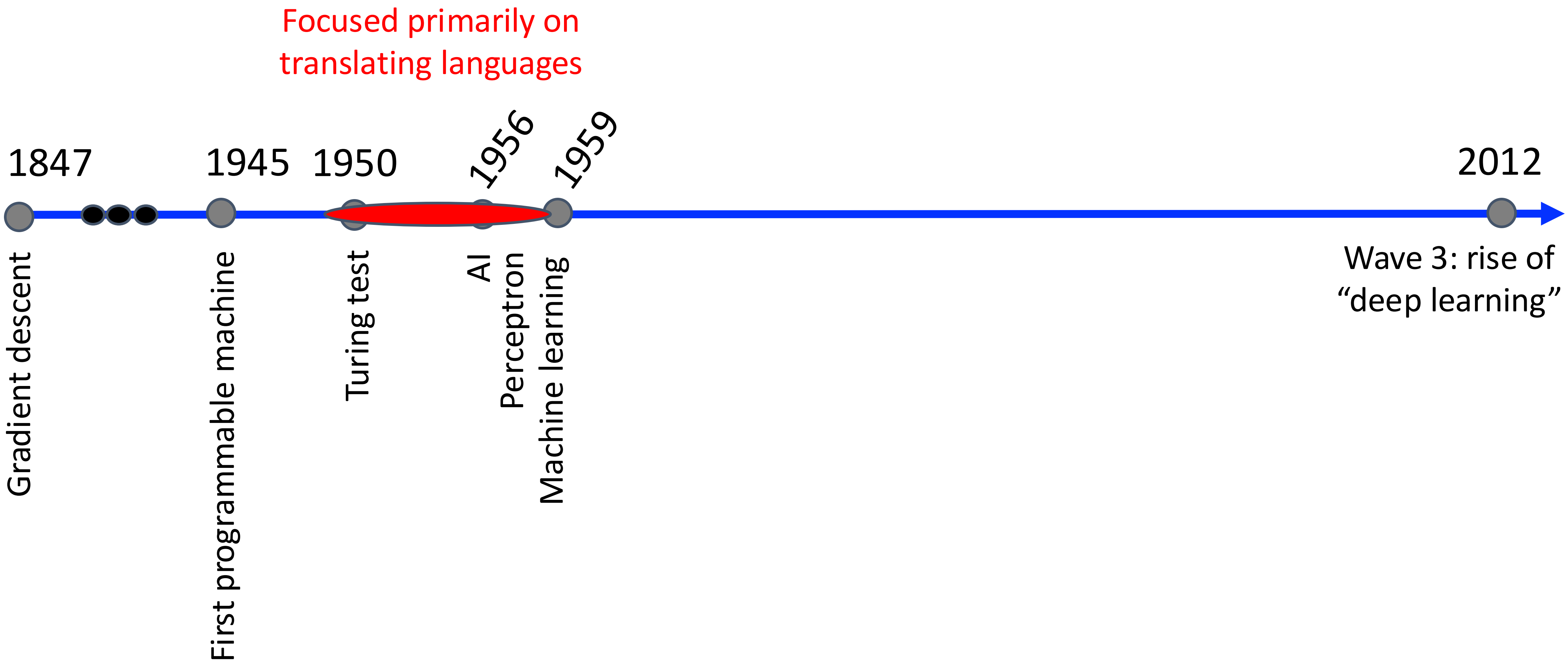
Today’s Topics

- Motivation: numeric representation of natural language
- Tokenization: how to convert text into discrete units
- Neural word embeddings: how to create dense representation
- Programming tutorial

Today’s Topics

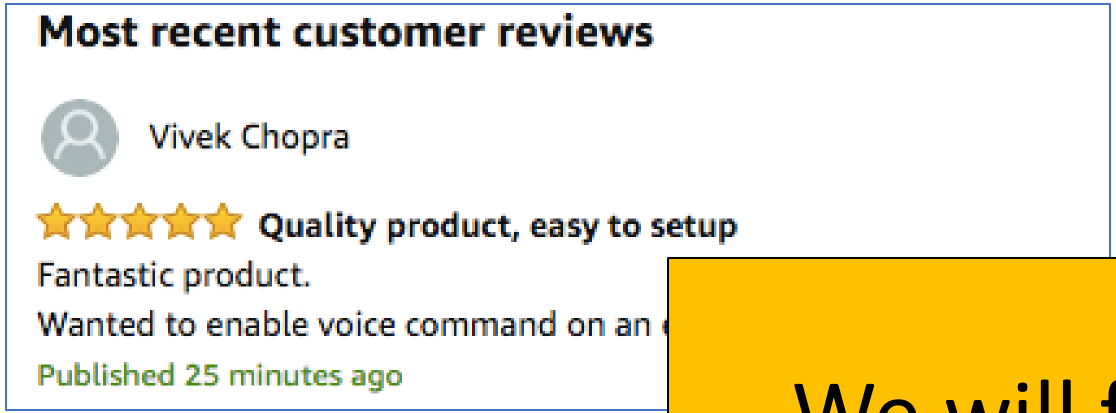
- Motivation: numeric representation of natural language
- Tokenization: how to convert text into discrete units
- Neural word embeddings: how to create dense representation
- Programming tutorial

Origins of Natural Language Processing

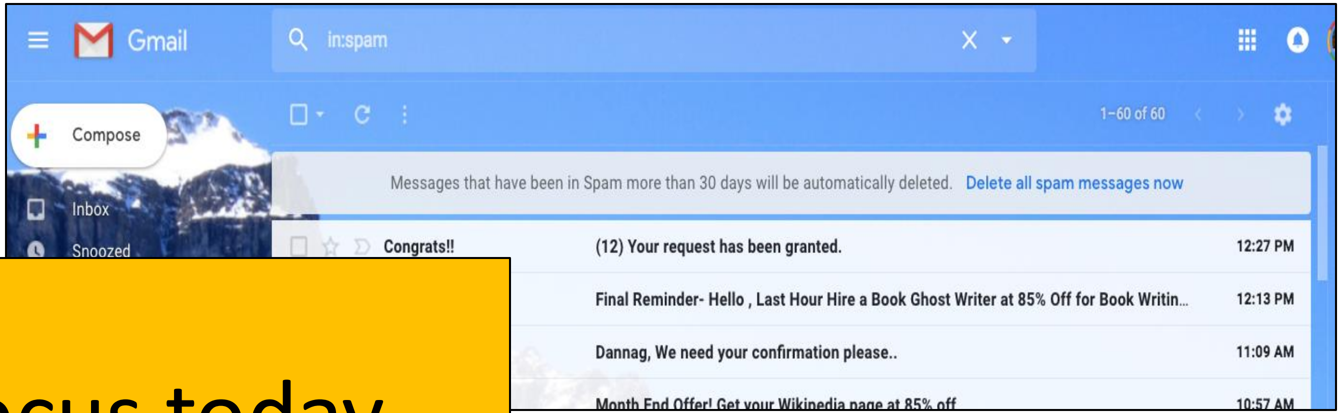


Jones. Natural Language Processing: A Historical Review. 1994.

Natural Language

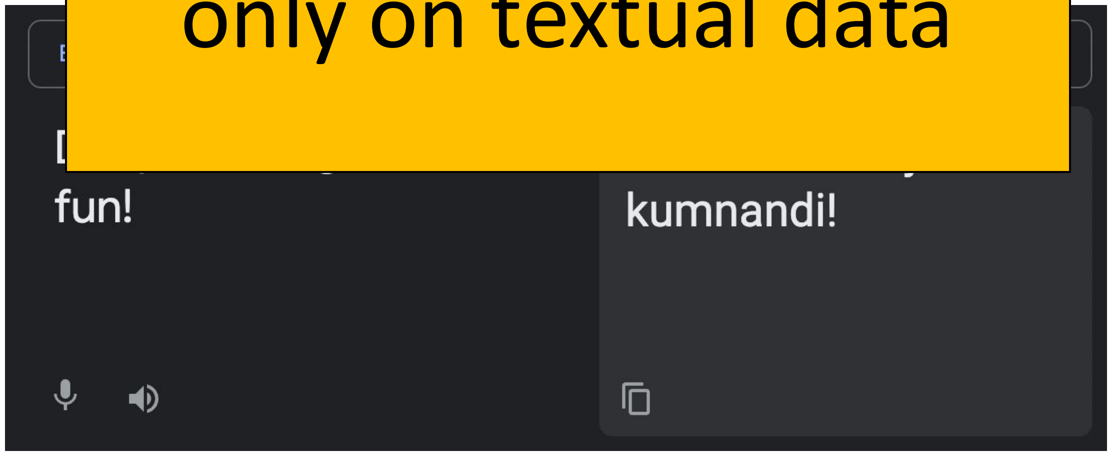


Opinion Mining



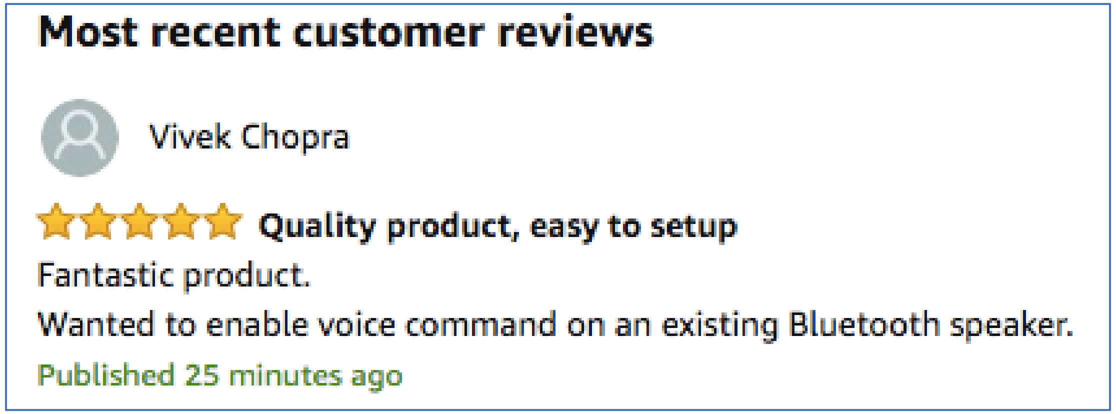
Spam Detection

We will focus today only on textual data

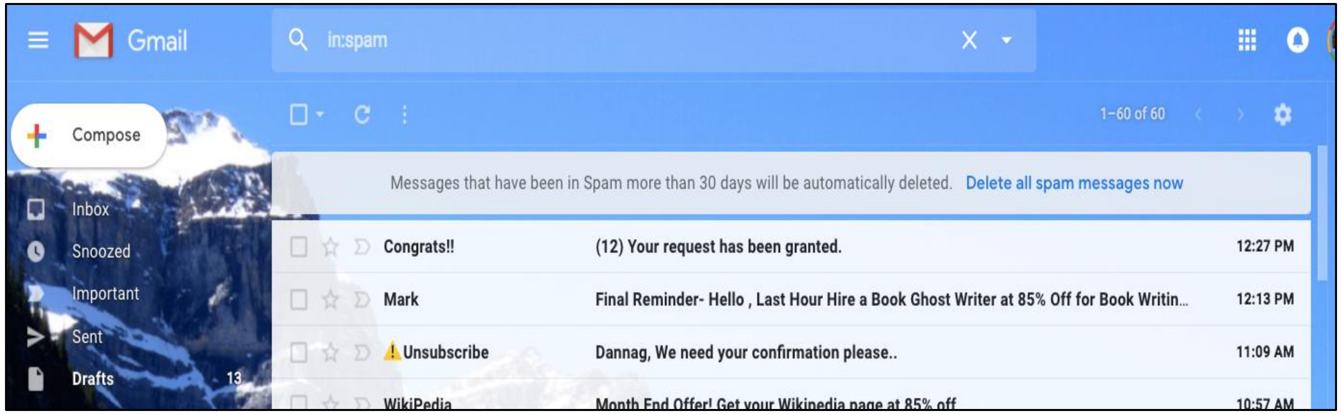


Language Translation

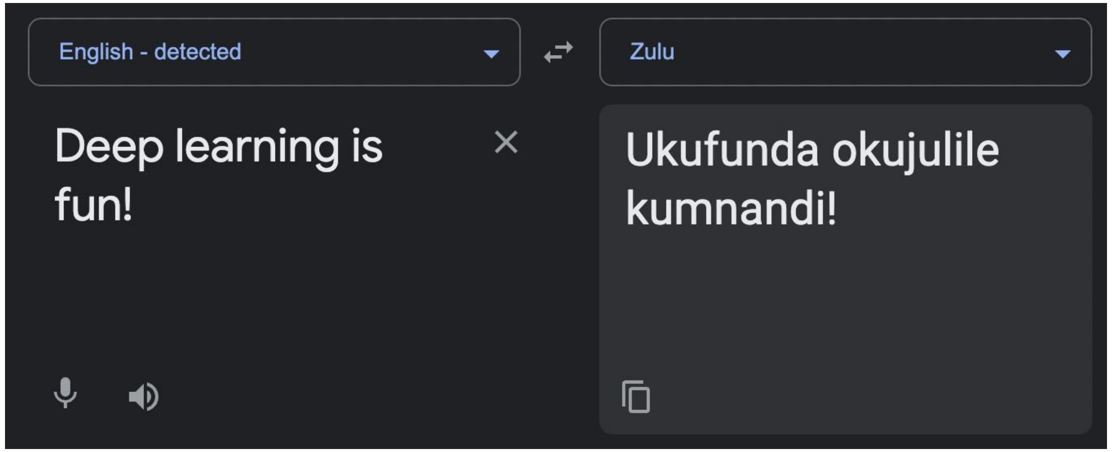
Input: String (Collection of Characters)



Opinion Mining



Spam Detection



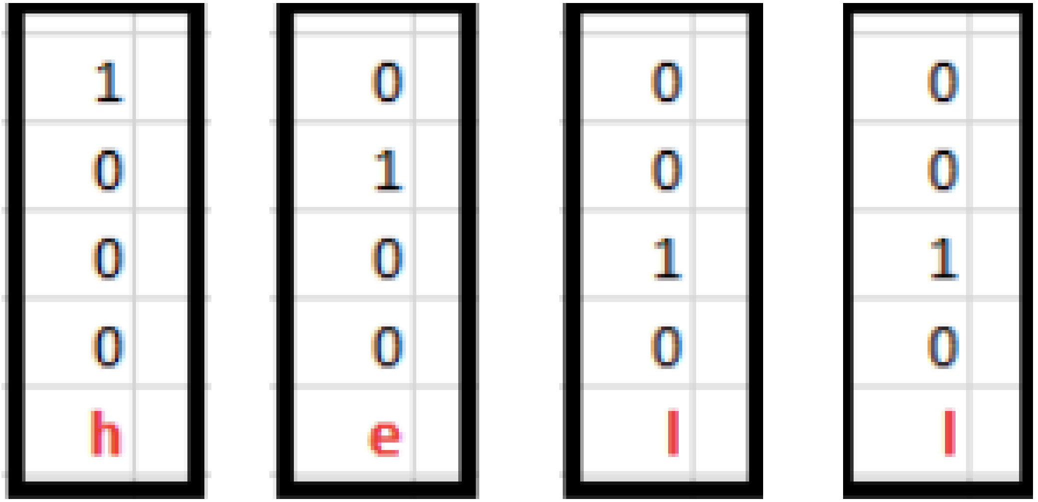
Language Translation

Input: Which “String” Feature Types Apply?

- ~~Categorical data~~
  - ~~Comes from a fixed list (e.g., education level)~~
- ~~Structured string data~~
  - ~~e.g., addresses, dates, telephone numbers,~~
- Text data

How to Feed Computers Text in the Required Numeric Format? (Recall RNN Example)

1. Tokenize training data: “hello” -> “h”, “e”, “l”, “l”, “o”
2. Learn vocabulary by identifying all unique tokens: {h, e, l, o}
3. Encode data as vectors; e.g., one hot encoding



https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/

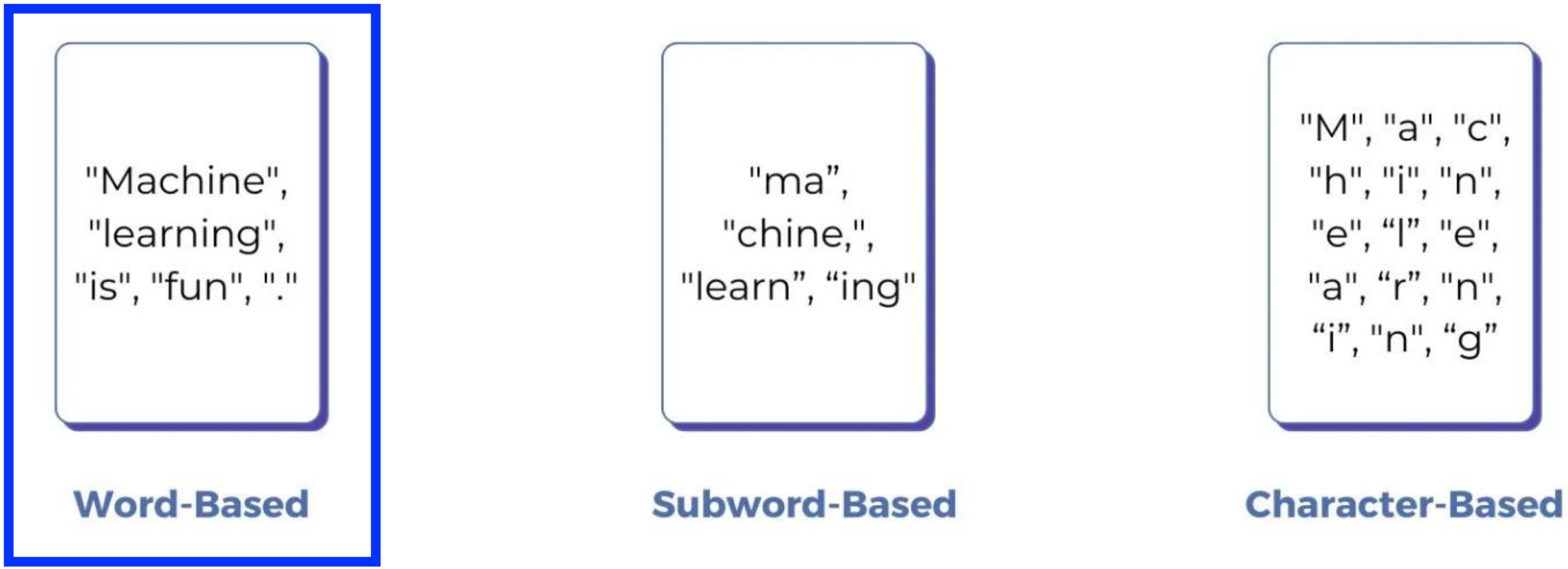
Challenge: Represent 7000+ spoken languages including individual nuances in each language?

https://ruder.io/nlp-beyond-english/

Today's Topics

- Motivation: numeric representation of natural language
- Tokenization: how to convert text into discrete units
- Neural word embeddings: how to create dense representation
- Programming tutorial

Tokenizers: Converting Text to Numeric Value



https://eastgate-software.com/jp/what-is-tokenization-in-nlp-everything-you-need-to-understand/

Tokenizer: Word-Based

- Use whitespace and punctuation to split tokens and then assign id to each token

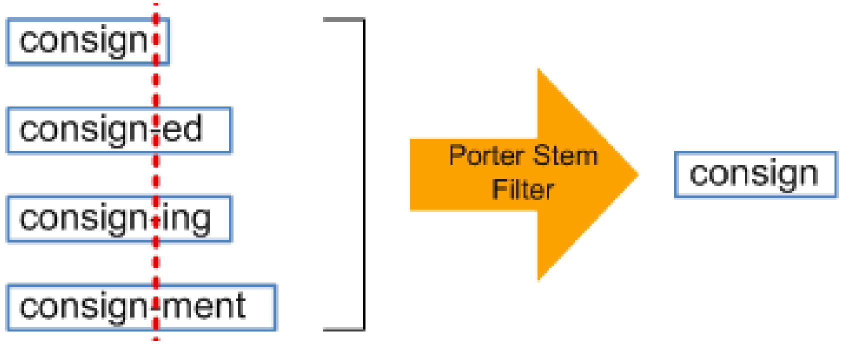
Token	a	an	at	***	bat	ball	***	zipper	zoo	***
Index	1	2	3	***	527	528	***	9,842	9,843	***

  - e.g., How many tokens should we see for “This is tokenizing.”
  - 4: [This] [is] [tokenizing] [.]

- What are limitations of this approach?
  - Huge vocabulary: ~1 million words in English alone according to Merriam- Webster (https://www.merriam-webster.com/help/faq-how-many-english-words)
  - Novel words: information is lost, because we use one token (e.g., “UNK”) for all instances

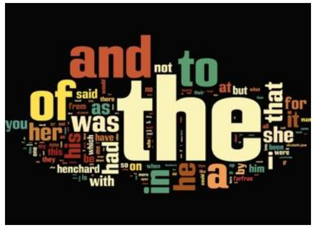
Pre-processing Ideas to Reduce Vocabulary

- Lower case all letters
- Use each word's "stem"; e.g., singular vs plural, reconcile different verb forms
  - e.g., stemming



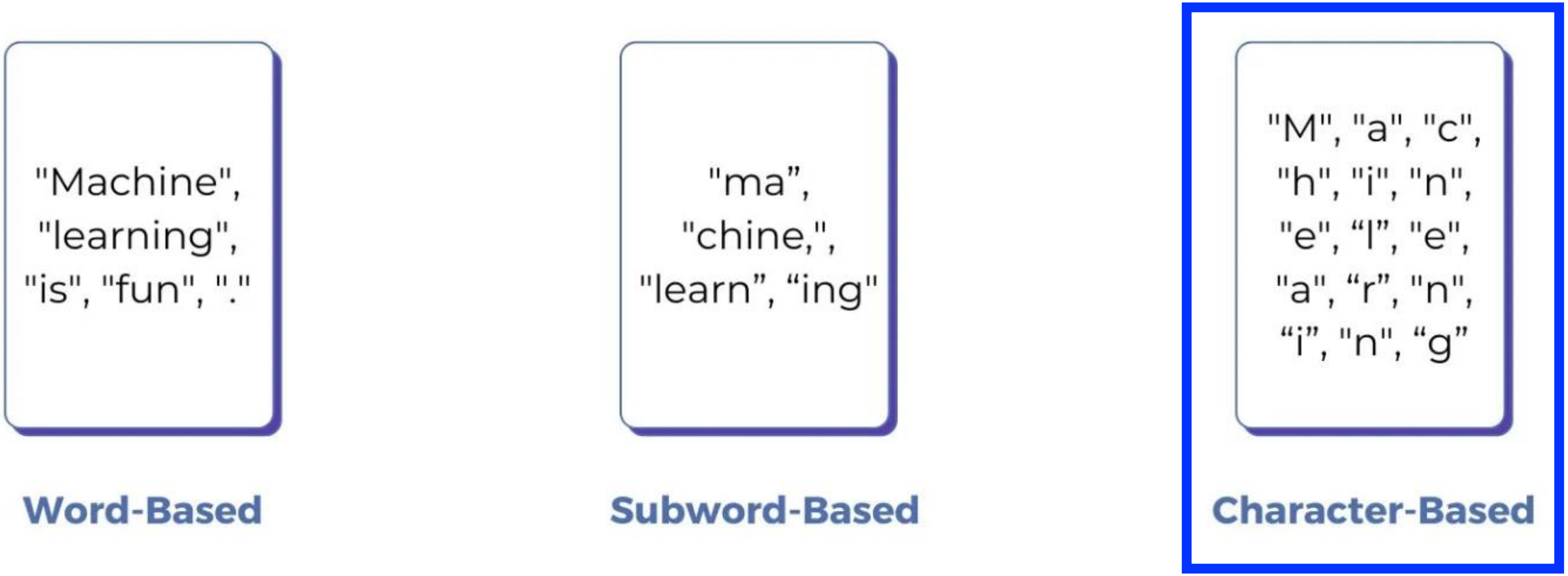
https://dzone.com/articles/using-lucene-grails

  - e.g., lemmatizing to get intended based word (e.g., caring -> care, not "car")
- Only use most popular N words, assigning UNK token to the rest
- Stop word removal: discard frequent words



https://github.com/topics/stopwords-removal

Tokenizers: Converting Text to Numeric Value



Overcome word-based limitations by using characters!

https://eastgate-software.com/jp/what-is-tokenization-in-nlp-everything-you-need-to-understand/

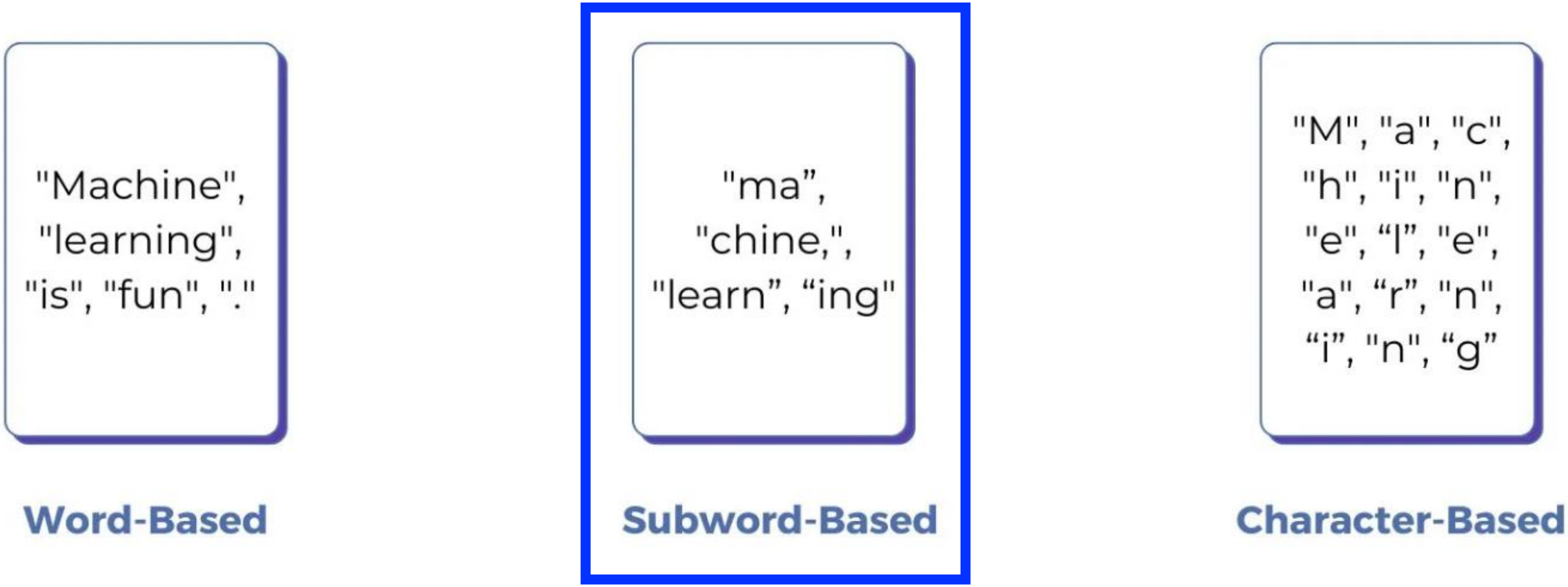
Tokenizer: Character-Based

- Id assigned to each unique character

Token	a	b	c	***	0	1	***	!	@	***
Index	1	2	3	***	27	28	***	119	120	***

  - e.g., How many tokens should we see for “This is tokenizing.”
  - 17: [T] [h] [i] [s] [i] [s] [t] [o] [k] [e] [n] [i] [z] [i] [n] [g] [.]
- Advantages of this approach:
  - Smaller vocabulary (e.g., 26 English letters plus punctuation and other symbols)
  - More known tokens (i.e., words are comprised of characters)
- What are limitations of this approach?
  - Reduced semantics**:** greater meaning ambiguity when we only see one character (e.g., “a”)
  - Longer input: greater computational expense; e.g., 17 characters vs 4 words for above example

Tokenizers: Converting Text to Numeric Value



Middle ground, widely used in modern neural networks

https://eastgate-software.com/jp/what-is-tokenization-in-nlp-everything-you-need-to-understand/

Tokenizer: Subword-Based

- Id assigned to common entities with merges/decompositions of rarer entities
  - e.g., 5 tokens for decomposition of “This is tokenizing.”: [This] [is] [token] [izing] [.]
- Compared to word-based and character-based tokenizers:
  - Middle-sized input
  - Middle-sized vocabulary
  - Middling semantics
  - Unknown tokens are rare

Popular Example: Byte Pair Encoding

1. Identify all tokens in the training data with their frequency
2. Define vocabulary size; e.g., 14
3. Add all characters in the tokenized input to the vocabulary; e.g.,

Character sequence	Cost
C o s t	2
b e s t	2
m e n u	1
m e n	1
c a m e l	1

https://static.packt-cdn.com/downloads/9781838821593_ColorImages.pdf

Popular Example: Byte Pair Encoding

1. Identify all tokens in the training data with their frequency
2. Define vocabulary size; e.g., 14
3. Add all characters in the tokenized input to the vocabulary; e.g.,
4. Until vocabulary is filled, add merged highest frequency symbol pairs

e.g., What are the highest frequency symbol pairs?

Character sequence	Cost	Vocabulary
C o s t	2	a, b, c, e, l, m, n, o, s, t, u
b e s t	2	
m e n u	1	
m e n	1	
c a m e l	1	

https://static.packt-cdn.com/downloads/9781838821593_ColorImages.pdf

Popular Example: Byte Pair Encoding

1. Identify all tokens in the training data with their frequency
2. Define vocabulary size; e.g., 14
3. Add all characters in the tokenized input to the vocabulary; e.g.,
4. Until vocabulary is filled, add merged highest frequency symbol pairs

e.g., What are the highest frequency symbol pairs?

Character sequence	Cost	Vocabulary
C o s t	2	a, b, c, e, l, m, n, o, s, t, u, st
b e s t	2	
m e n u	1	
m e n	1	
c a m e l	1	

https://static.packt-cdn.com/downloads/9781838821593_ColorImages.pdf

Popular Example: Byte Pair Encoding

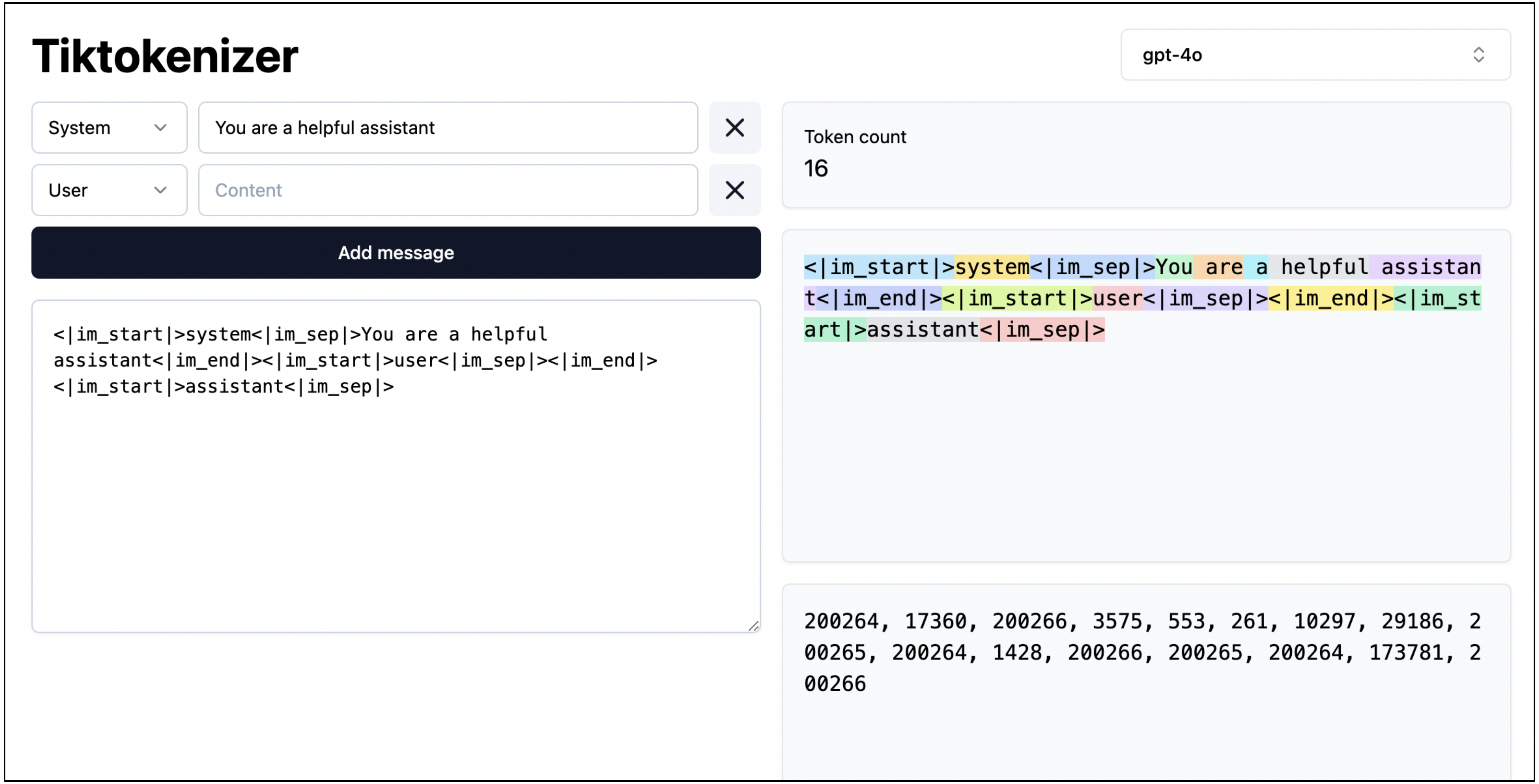
1. Identify all tokens in the training data with their frequency
2. Define vocabulary size; e.g., 14
3. Add all characters in the tokenized input to the vocabulary; e.g.,
4. Until vocabulary is filled, add merged highest frequency symbol pairs

e.g., What are the highest frequency symbol pairs?

Character sequence	Cost	Vocabulary
C o s t	2	a, b, c, e, l, m, n, o, s, t, u, st, me, **men**
b e s t	2	
m e n u	1	
m e n	1	
c a m e l	1	

https://static.packt-cdn.com/downloads/9781838821593_ColorImages.pdf

Popular Focus Today: Improving Tokenization



https://tiktokenizer.vercel.app/

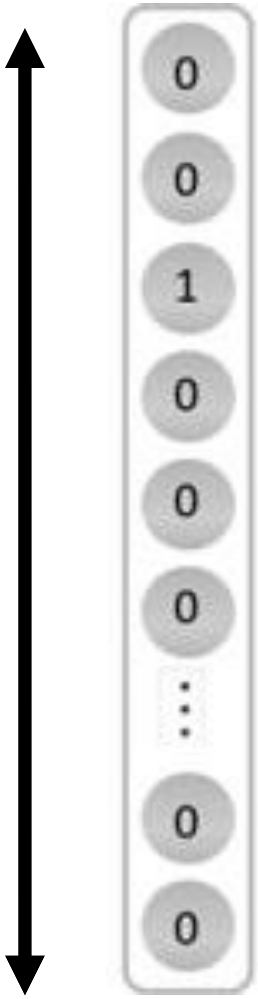
Today’s Topics

- Motivation: numeric representation of natural language
- Tokenization: how to convert text into discrete units
- Neural word embeddings: how to create dense representation
- Programming tutorial

Problems with One-Hot Encoding Words?

Dimensionality = vocabulary size

e.g., English has ~170,000 words
with ~10,000 commonly used words

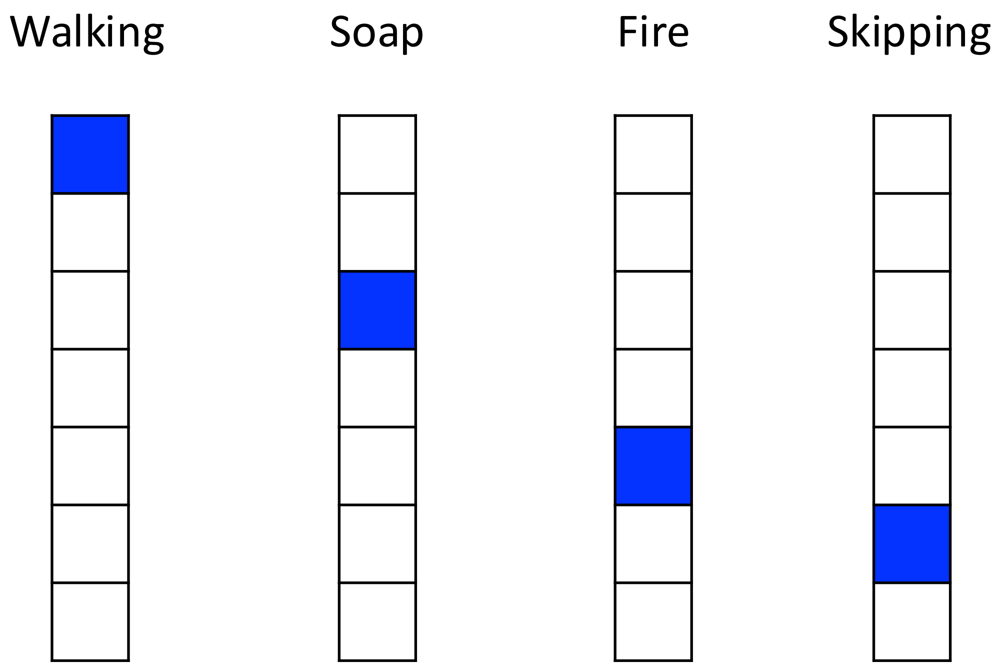


- Huge memory burden
- Computationally expensive

Kamath, Liu, and Whitaker. Deep Learning for NLP and Speech Recognition. 2019

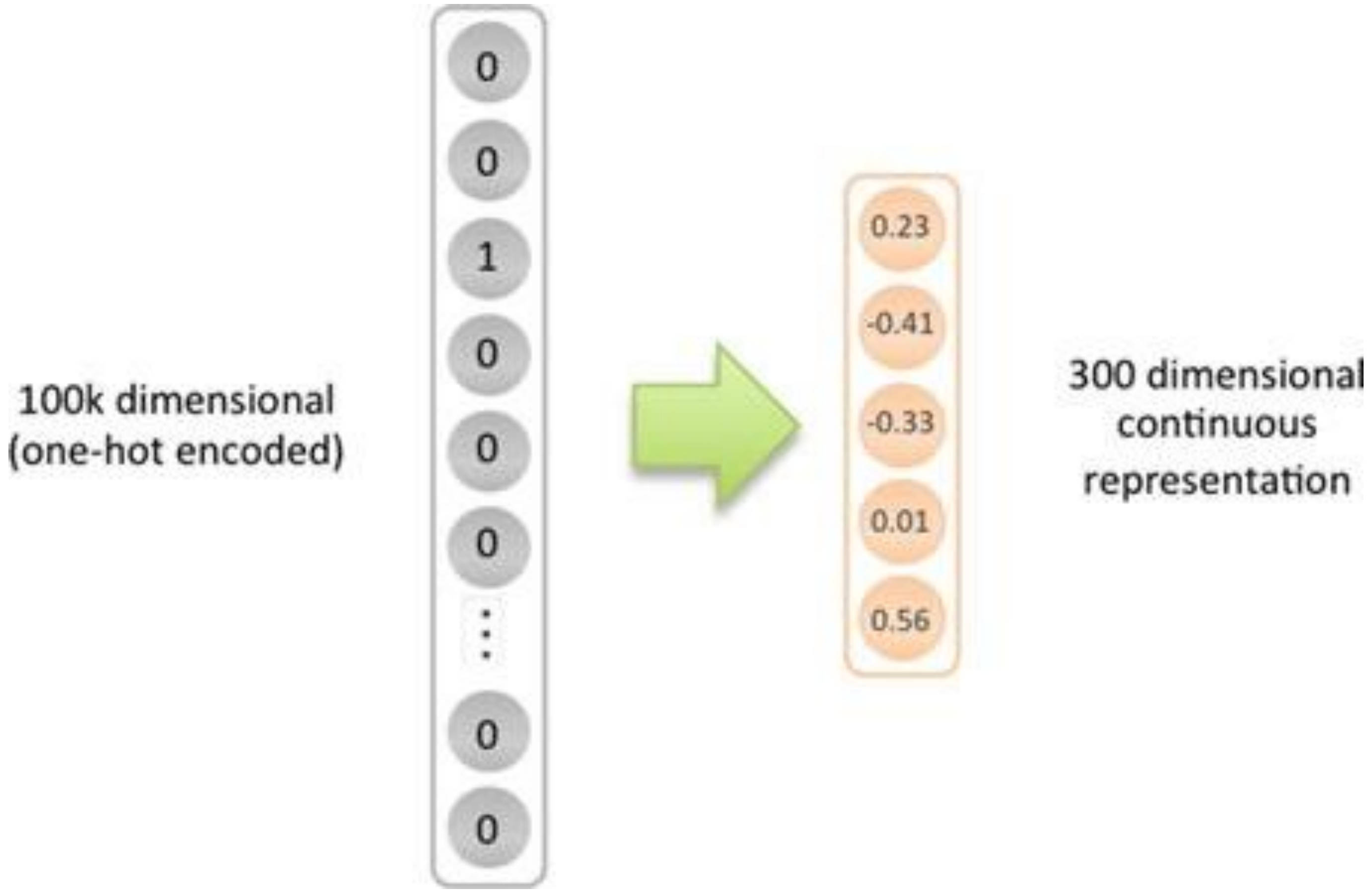
Limitation of One-Hot Encoding Words

- No notion of which words are similar, yet such understanding can improve generalization
  - e.g., “walking”, “running”, and “skipping” are all suitable for “He was ____ to school.”



The distance between all words is equal!

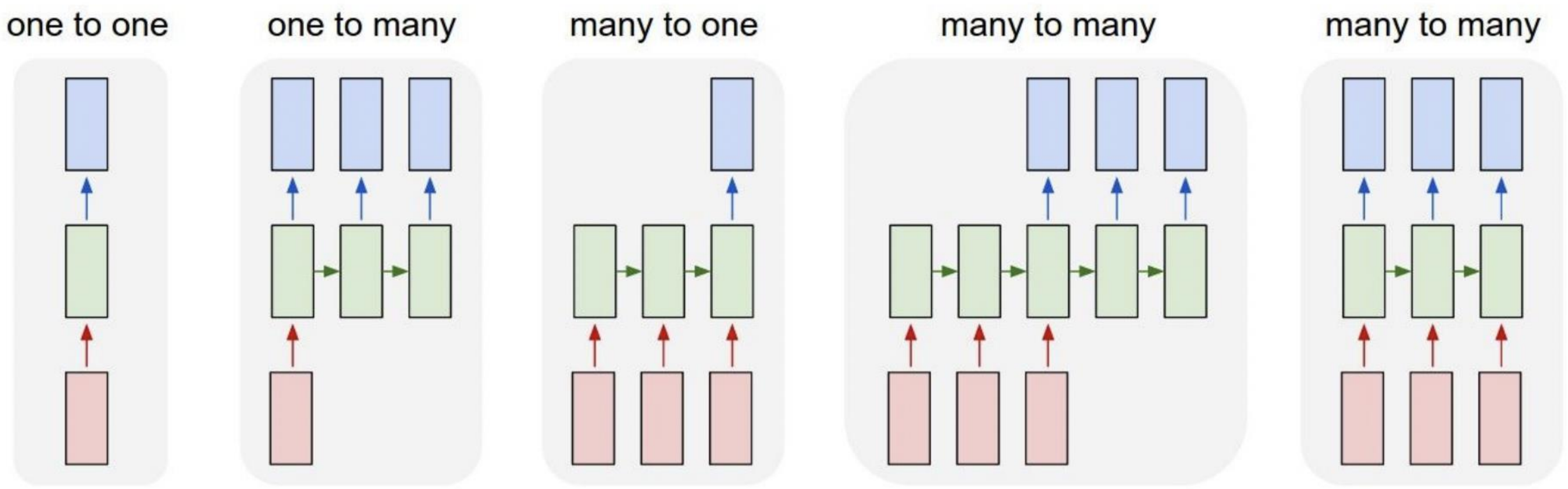
Idea: Represent Each Word Compactly in a Space Where Vector Distance Indicates Word Similarity



Kamath, Liu, and Whitaker. Deep Learning for NLP and Speech Recognition. 2019

Potential Use (Of Numerous)

- Convert words into compact vectors as input to neural networks; e.g., RNNs



https://www.analyticsvidhya.com/blog/2017/12/introduction-to-recurrent-neural-networks/

Inspiration: Distributional Semantics

"The distributional hypothesis says that the meaning of a word is derived from the context in which it is used, and words with similar meaning are used in similar contexts."

- Origins: Harris in 1954 and Firth in 1957

Inspiration: Distributional Semantics

"The distributional hypothesis says that the meaning of a word is derived from the context in which it is used, and words with similar meaning are used in similar contexts."

Kamath, Liu, and Whitaker. Deep Learning for NLP and Speech Recognition. 2019

Inspiration: Distributional Semantics

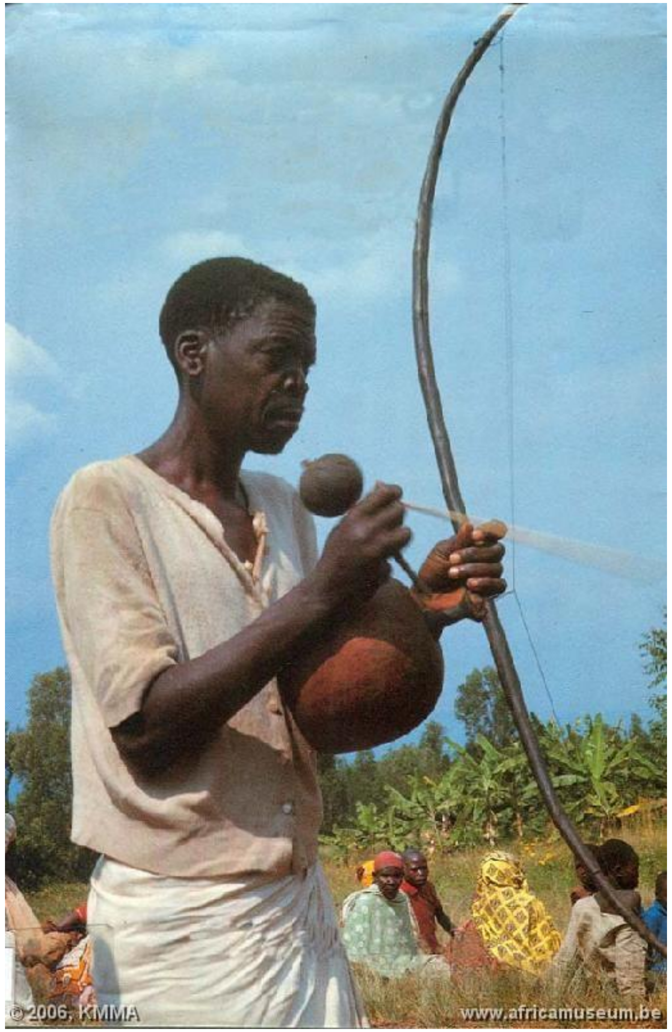
- What is the meaning of berimbau based on context?

  Background music from a berimbau offers a beautiful escape.

  Many people danced around the berimbau player.

  I practiced for many years to learn how to play the berimbau.

- Idea: context makes it easier to understand a word's meaning



[Adapted from slides by Lena Voita]

https://capoeirasongbook.wordpress.com/instruments/berimbau/

Inspiration: Distributional Semantics

"The distributional hypothesis says that the meaning of a word is derived from the context in which it is used, and words with similar meaning are used in similar contexts."

Kamath, Liu, and Whitaker. Deep Learning for NLP and Speech Recognition. 2019

Inspiration: Distributional Semantics

- What other words could fit into these contexts?

1. Background music from a _______ offers a beautiful escape.
2. Many people danced around the _______ player.
3. I practiced for many years to learn how to play the _______.

Hypothesis is that words with similar row values have similar meanings

	1.	2.	3.
Berimbau	1	1	1
Soap	0	0	0
Fire	0	0	0
Guitar	1	1	1

Contexts

1 if a word can appear in the context
0 otherwise

[Adapted from slides by Lena Voita]

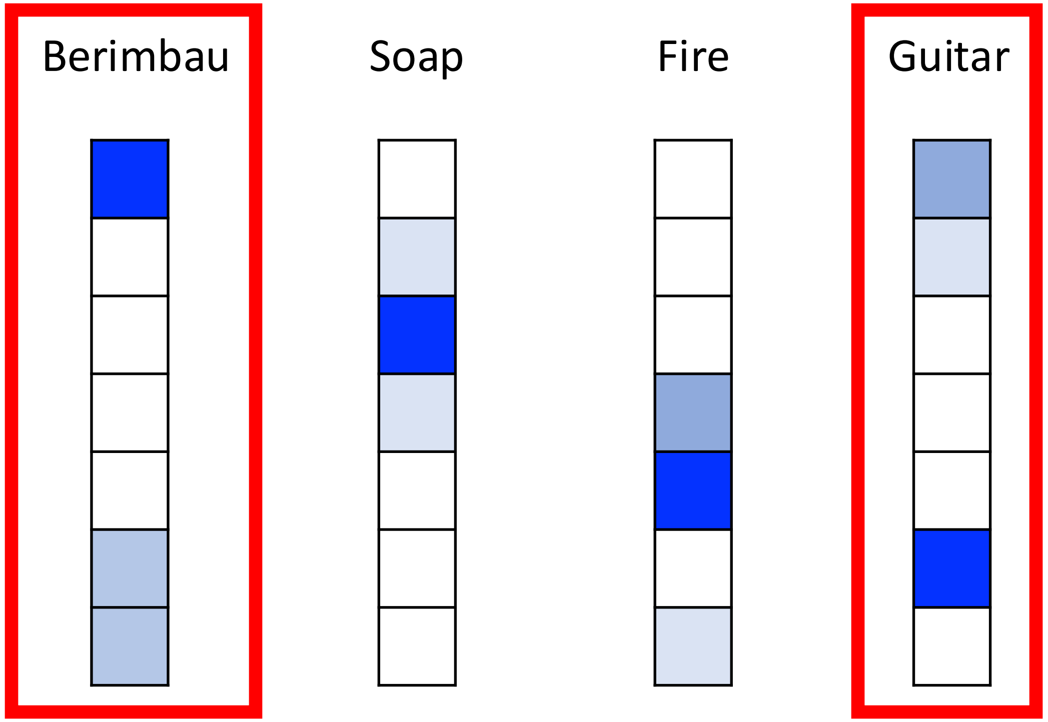
Inspiration: Distributional Semantics

"The distributional hypothesis says that the meaning of a word is derived from the context in which it is used, and words with similar meaning are used in similar contexts."

Kamath, Liu, and Whitaker. Deep Learning for NLP and Speech Recognition. 2019

Approach

- Learn a dense (lower-dimensional) vector for each word by characterizing its **context**, which inherently will reflect similarity/differences to other words



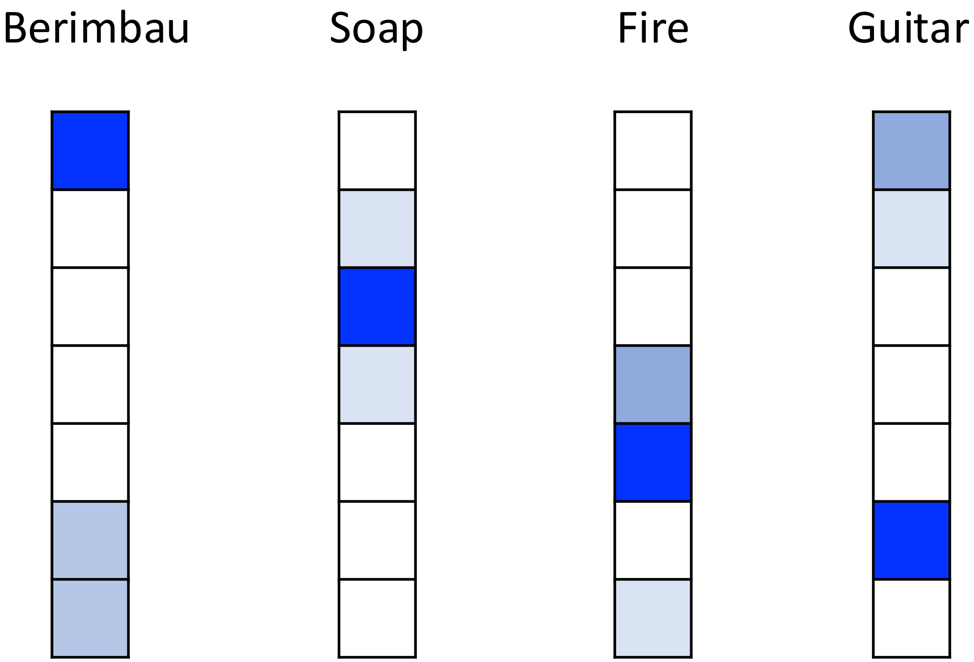
Berimbau and guitar are the closest word pair

The distance between each pair of words differs!

Note: many ways to measure distance (e.g., cosine distance)

Approach

- Learn a dense (lower-dimensional) vector for each word by characterizing its **context**, which inherently will reflect similarity/differences to other words



We embed words in a shared space so they can be compared with a few features

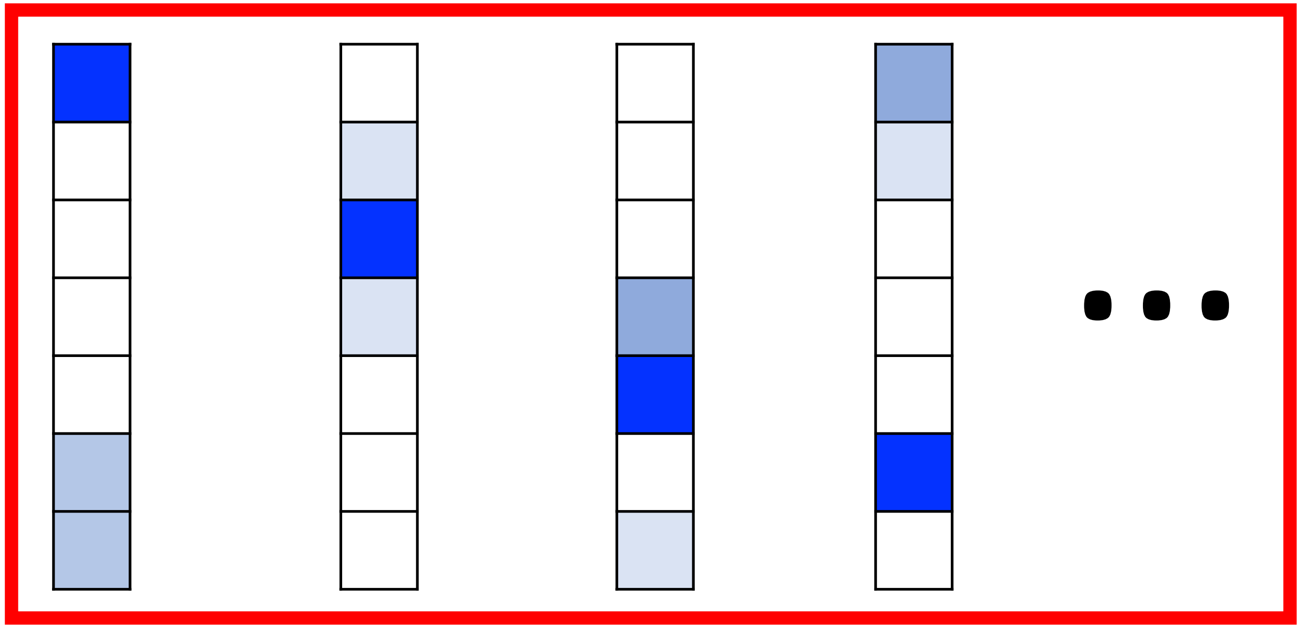
What features would discriminate these words?

Approach

- Learn a dense (lower-dimensional) vector for each word by characterizing its **context**, which inherently will reflect similarity/differences to other words

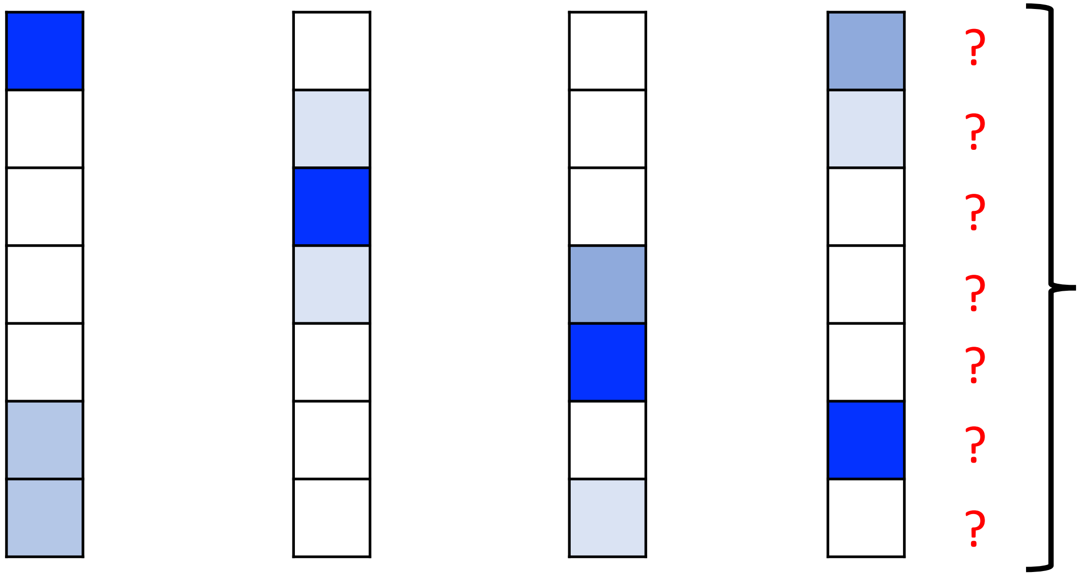
Approach: Learn Word Embedding Space

- An **embedding space** represents a finite number of words, decided in training
- A **word embedding** is represented as a vector indicating its context
- The dimensionality of all word embeddings in an embedding space match
  - What is the word embedding dimensionality for the shown example?



Approach: Learn Word Embedding Space

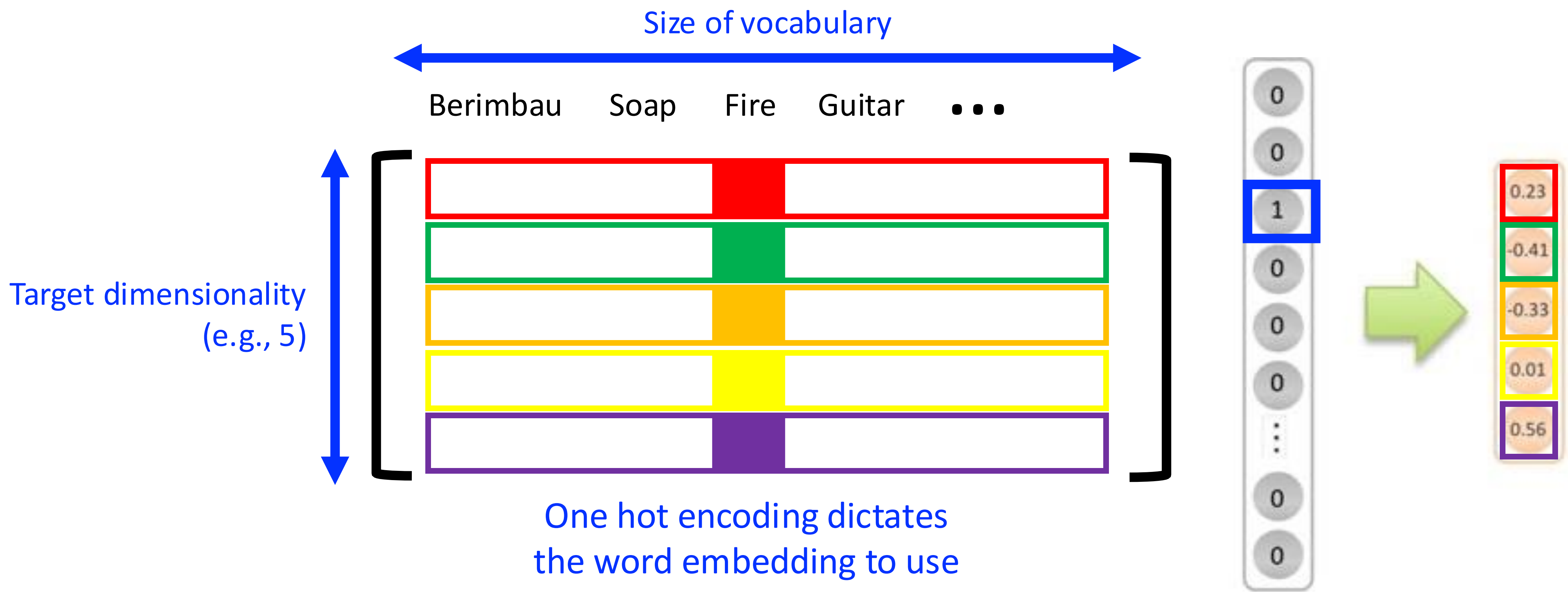
- An embedding space represents a finite number of words, defined in training
- A word embedding is represented as a vector indicating its context
- The dimensionality of all word embeddings in an embedding space match



In practice, the learned discriminating features are hard to interpret

Embedding Matrix

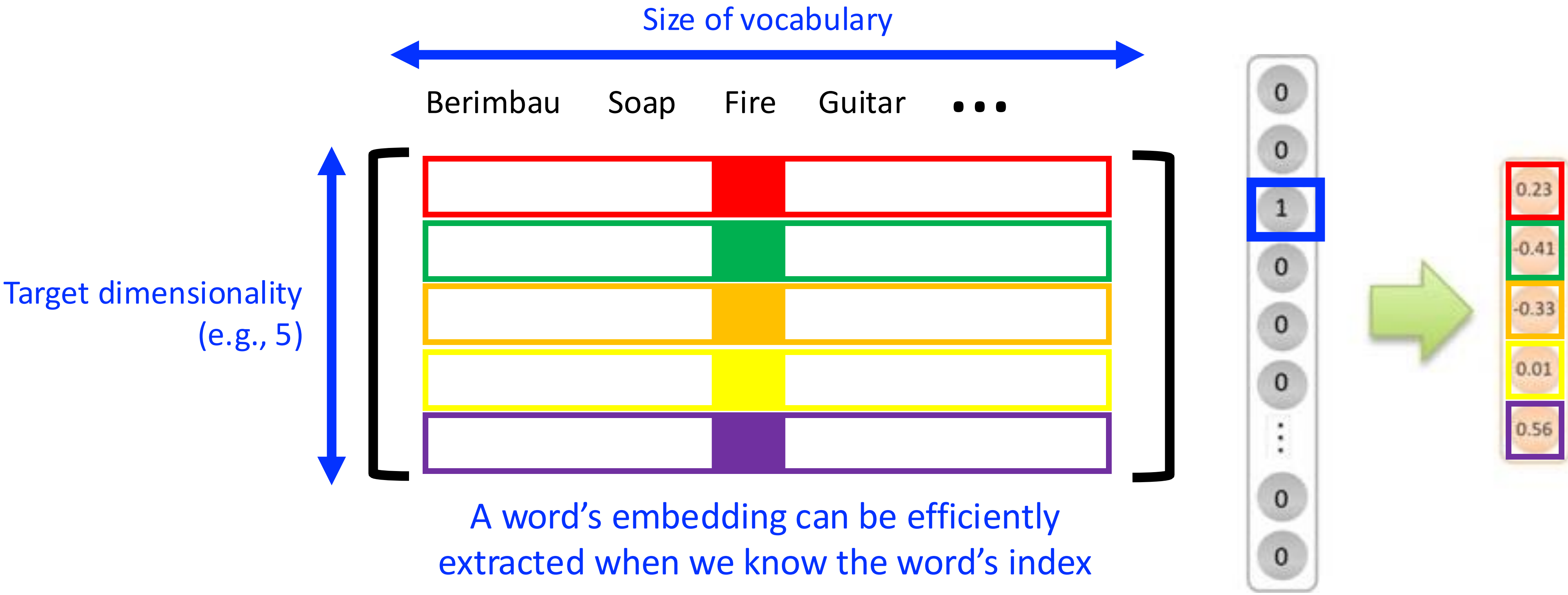
- The embedding matrix converts an input word into a dense vector



Kamath, Liu, and Whitaker. Deep Learning for NLP and Speech Recognition. 2019

Embedding Matrix

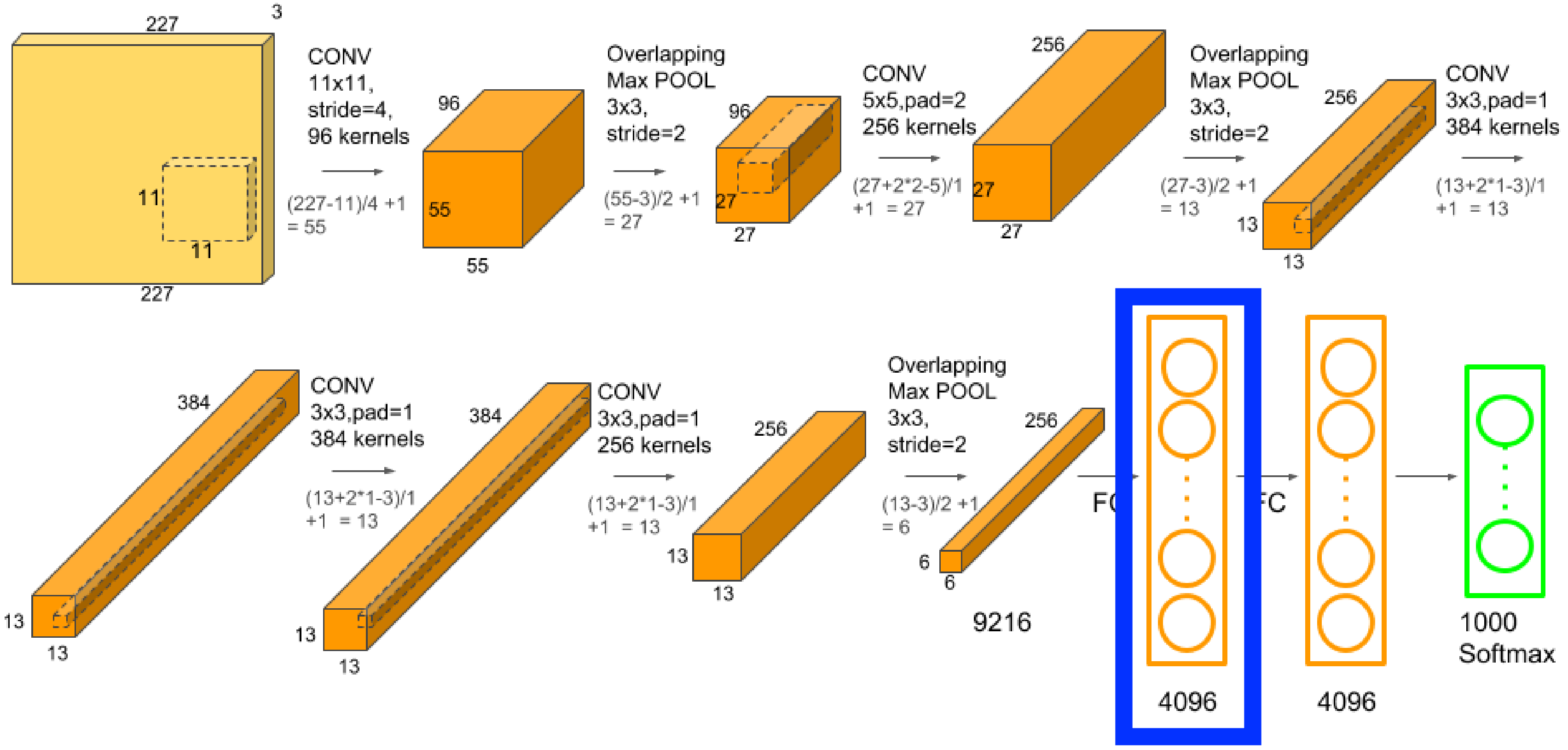
- It converts an input word into a dense vector



Kamath, Liu, and Whitaker. Deep Learning for NLP and Speech Recognition. 2019

Word Embedding Analogous to a CNN Pretrained Feature

- e.g., FC6 layer of AlexNet

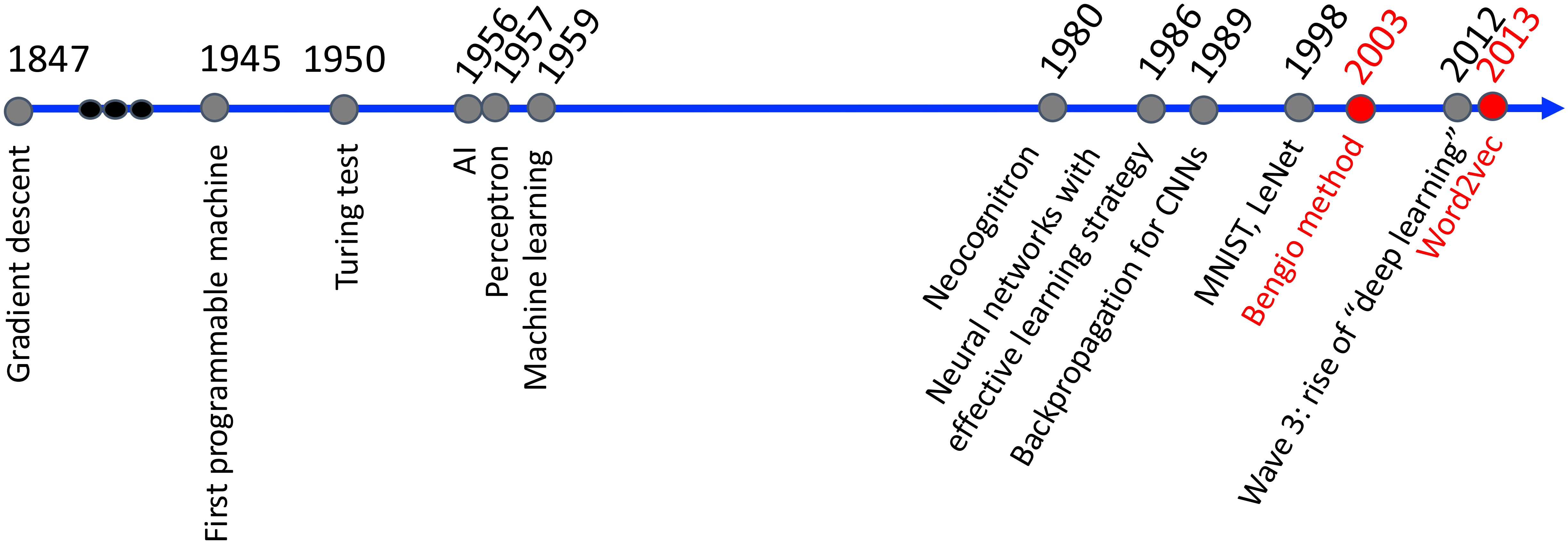


https://www.learnopencv.com/wp-content/uploads/2018/05/AlexNet-1.png

Popular Word Embeddings

- Bengio method
- Word2vec (skip-gram model)
- And more…

Historical Context



Popular Word Embeddings

- Bengio method
- Word2vec (skip-gram model)
- And more…

Idea: Learn Word Embeddings That Help Predict Viable Next Words

e.g.,

1. Background music from a ________
2. Many people danced around the ________
3. I practiced for many years to learn how to play the ________

Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Task: Predict Next Word Given Previous Ones

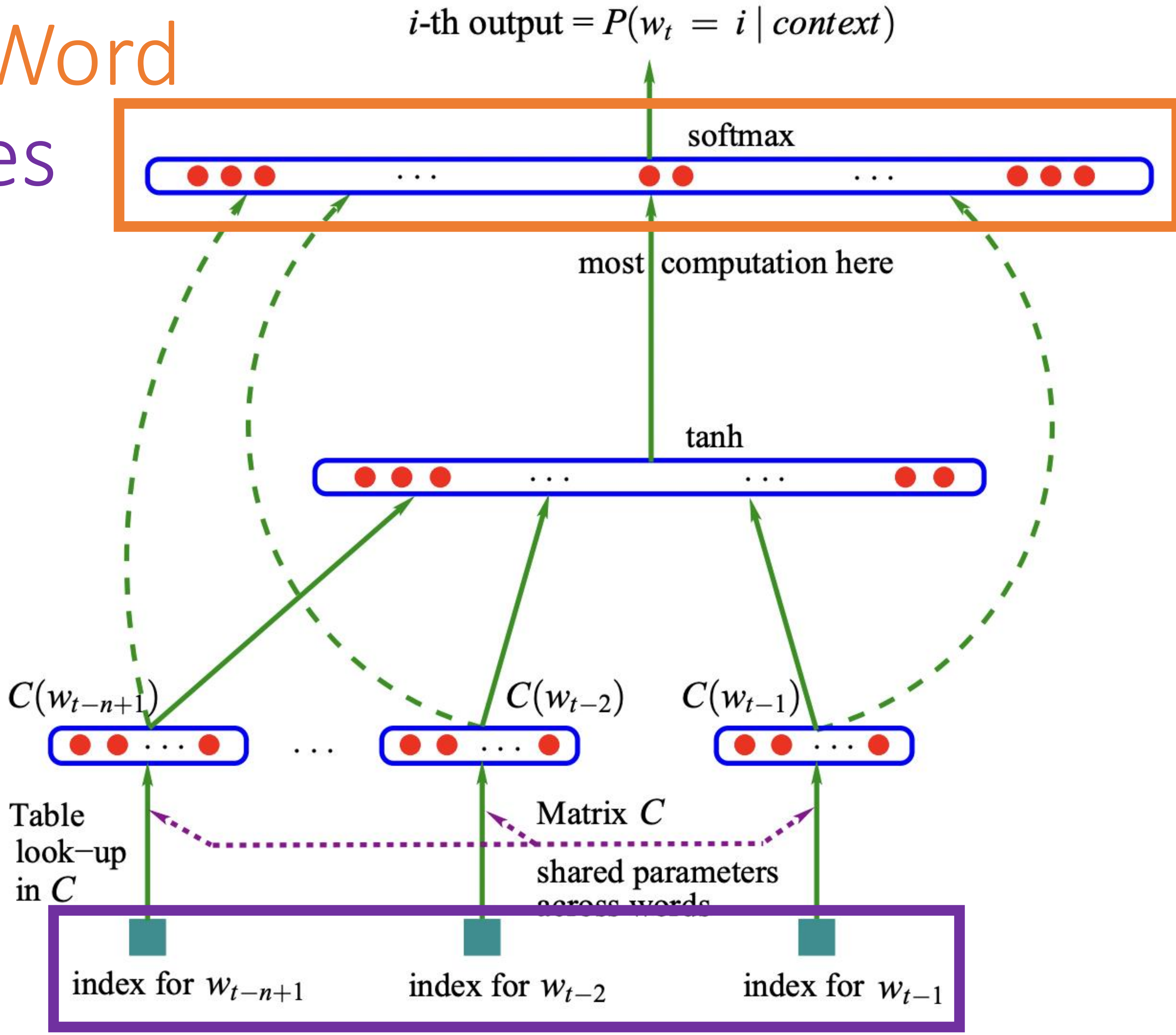
e.g.,

1. Background music from a ________
2. Many people danced around the ________
3. I practiced for many years to learn how to play the ________

Task: Predict Next Word Given Previous Ones

e.g., a vocabulary size of 17,000 was used in experiments

What is the dimensionality of the output layer?
- 17,000 (each indexed position indicates probability of a word)



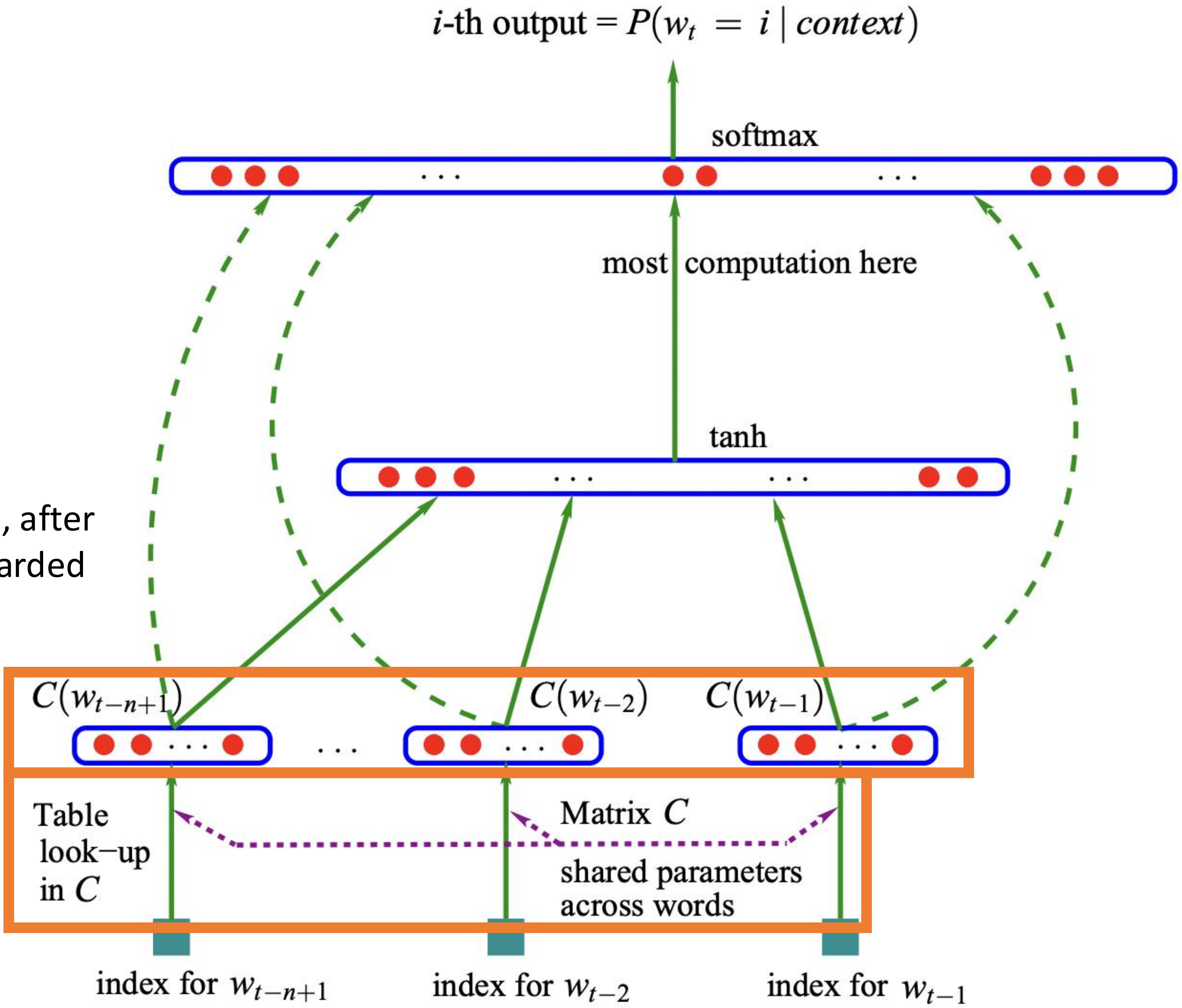
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Architecture

Note: the goal is to learn an embedding matrix and, after training, the rest of the neural network can be discarded

Word embeddings:

Embedding matrix:



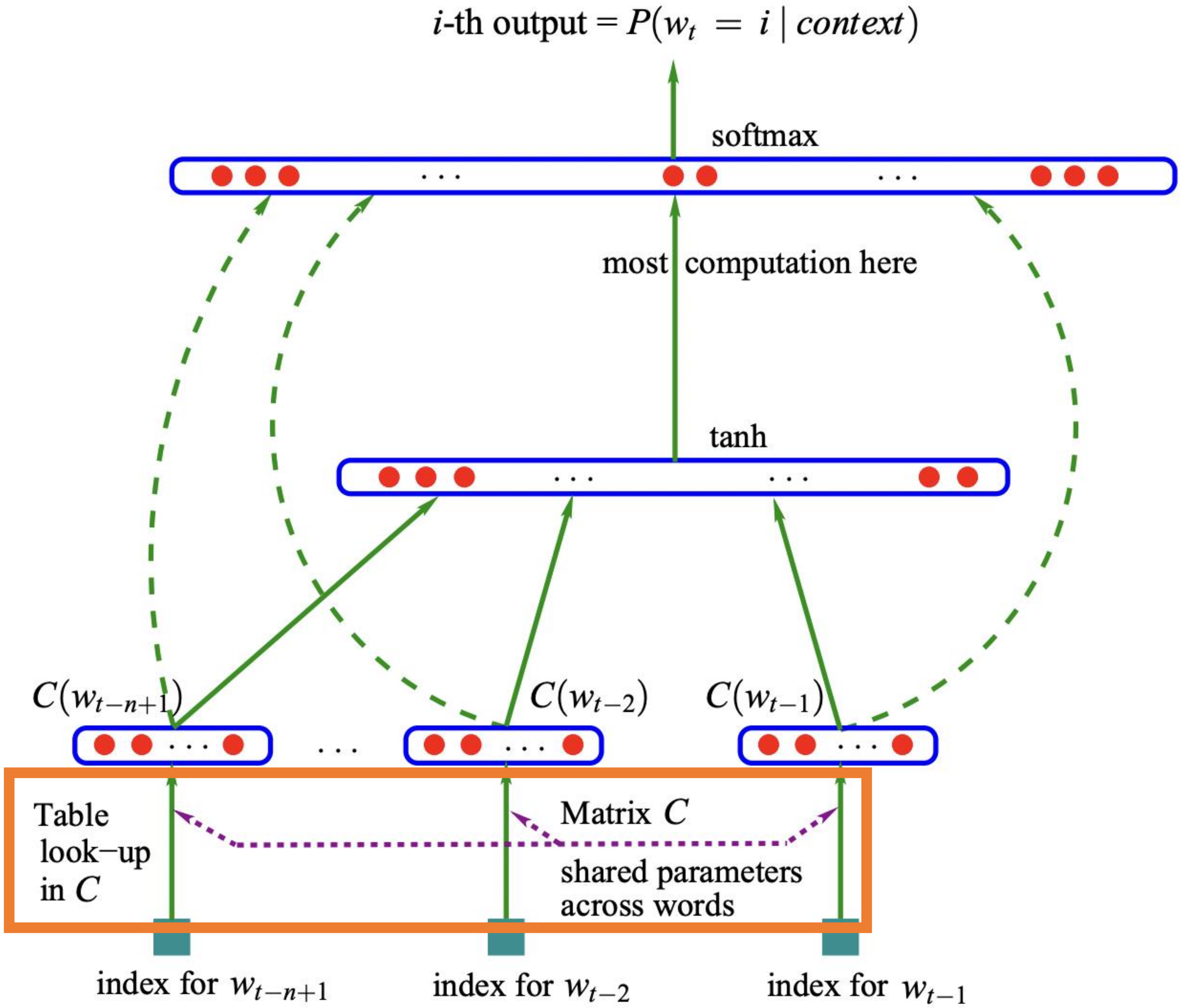
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Architecture

e.g., a vocabulary size of 17,000 was used with embedding sizes of 30, 60, and 100 in experiments

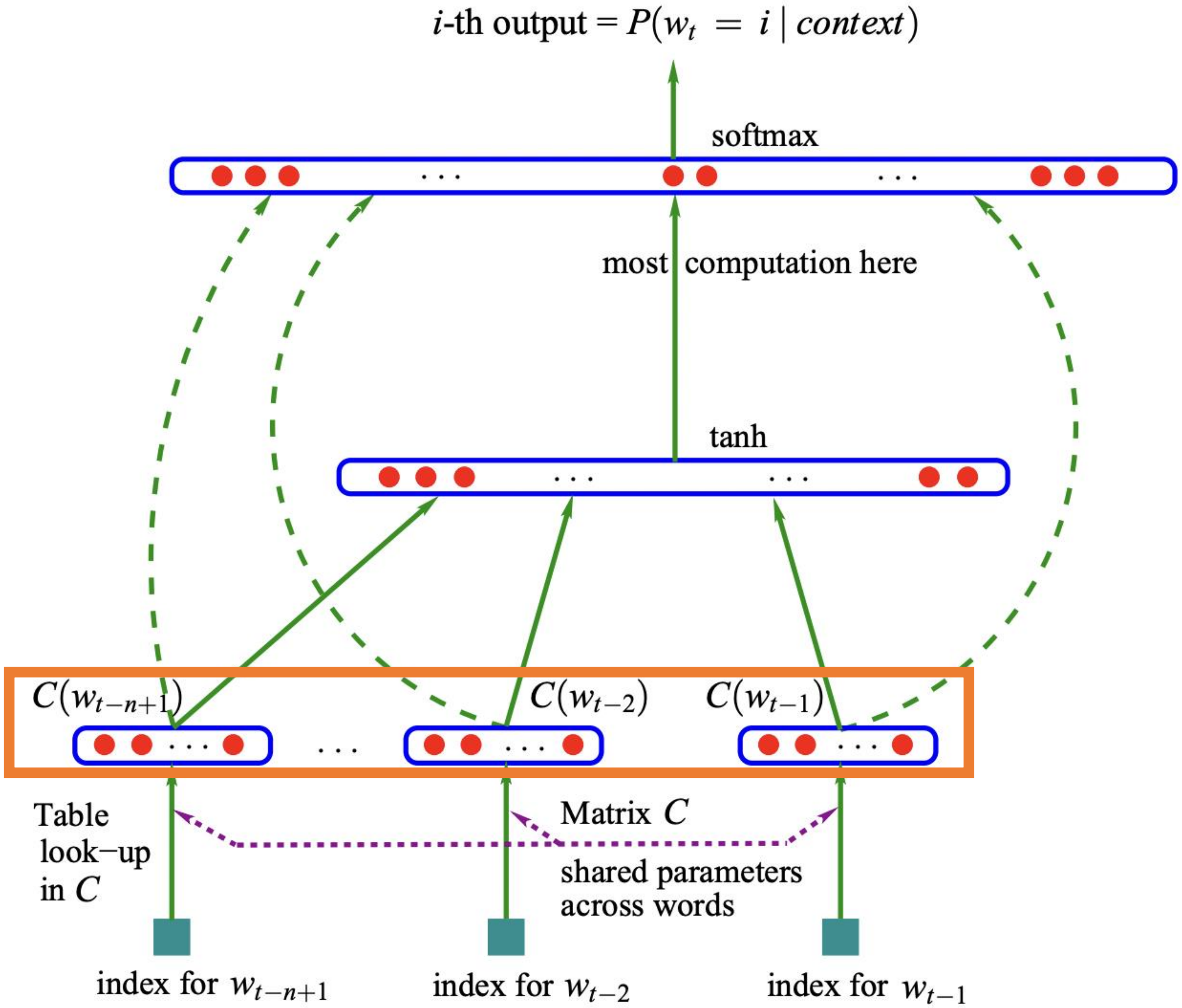
Assume a 30-d word embedding
 - what are the dimensions of the embedding matrix C?

30 x 17,000 (i.e., 510,000 weights)



Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Architecture



e.g., a vocabulary size of 17,000 was used with embedding sizes of 30, 60, and 100 in experiments

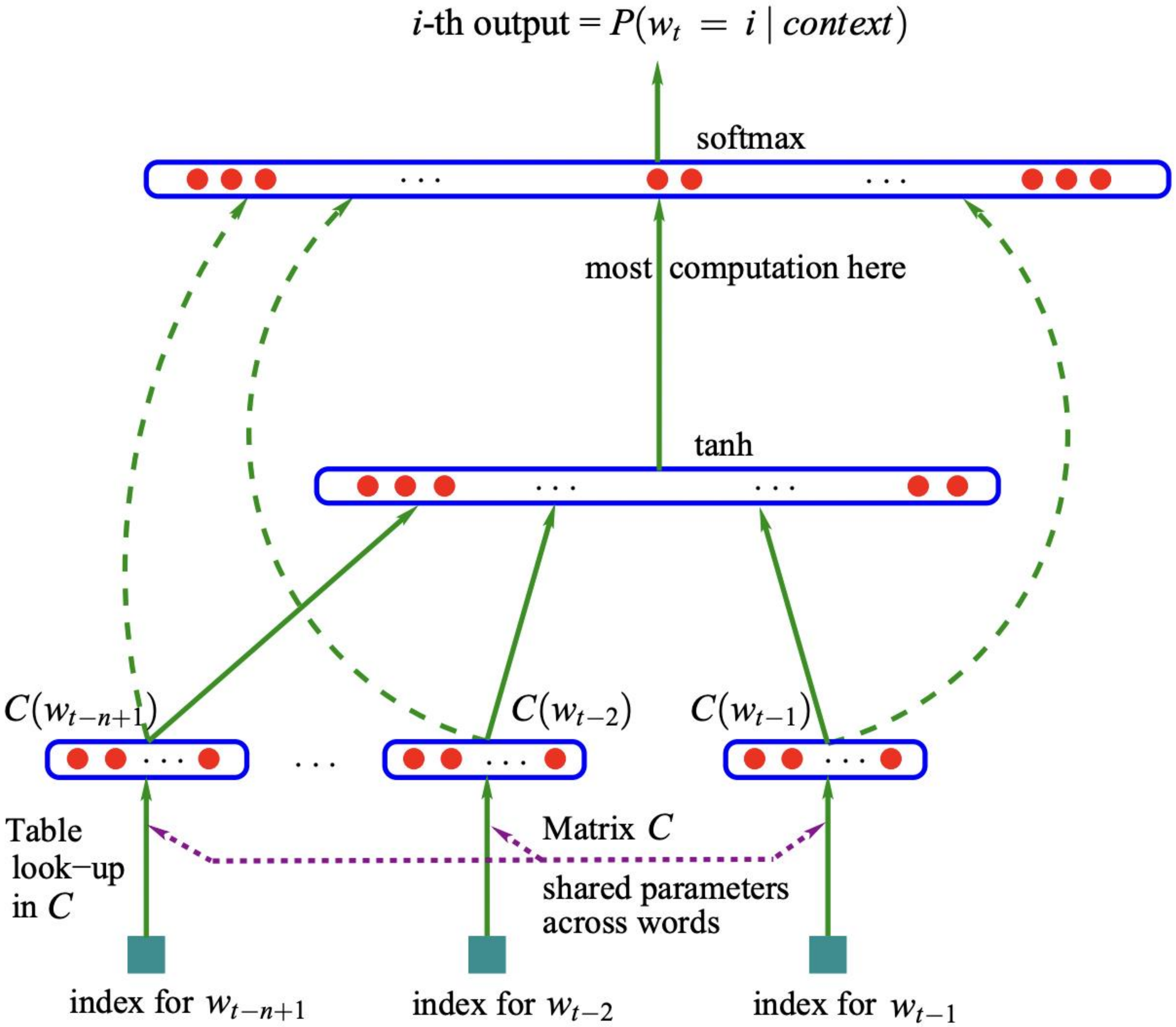
Assume a 30-d word embedding
 - what are the dimensions of each word embedding?

1 x 30

Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Architecture

Projection layer followed by a hidden layer with non-linearity



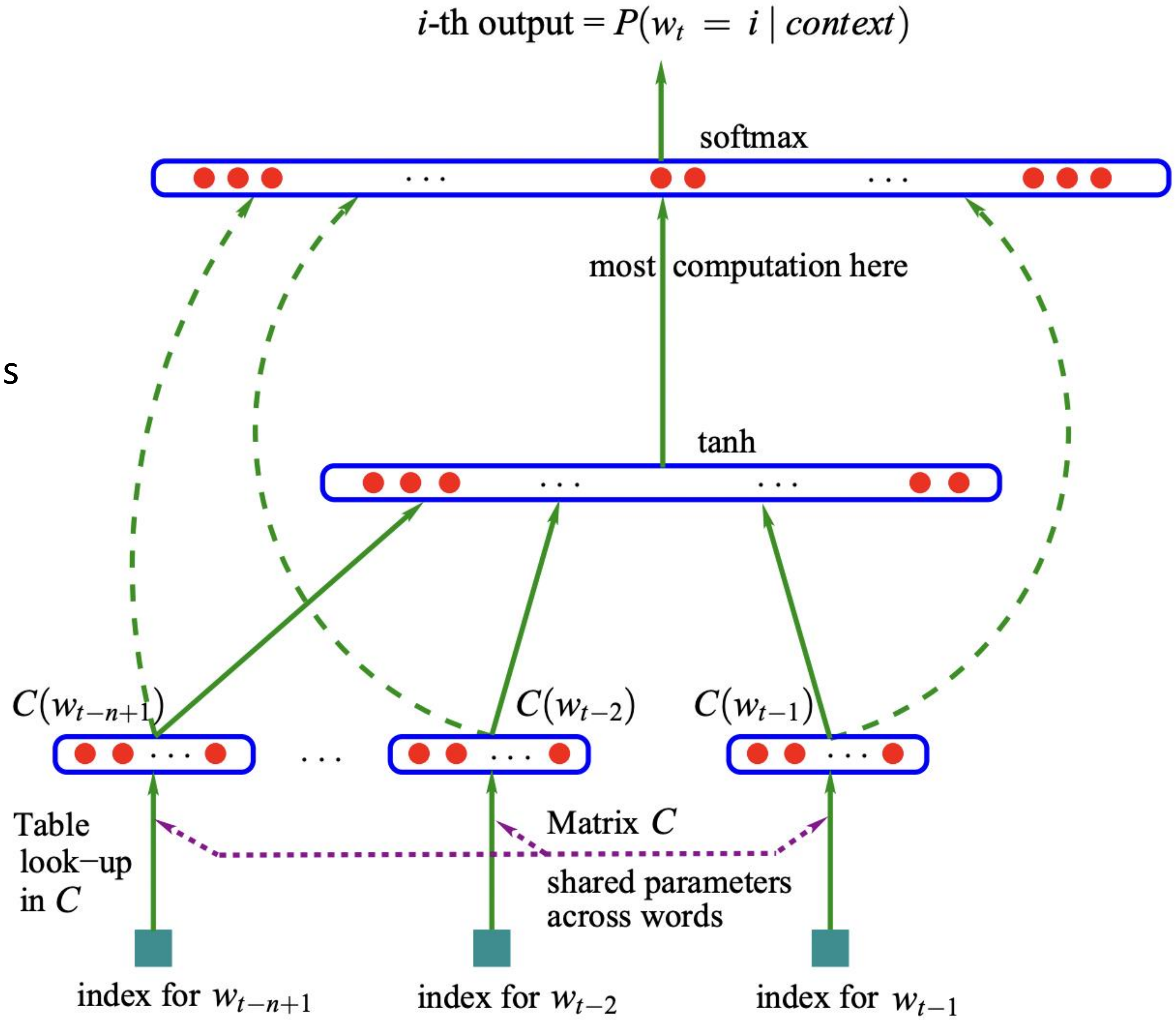
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Training

Use sliding window on input data; e.g., 3 words

Background music from a berimbau offers a beautiful escape…

Input: tried 1, 3, 5, and 8 input words and used 2 datasets with ~1 million and ~34 million words respectively



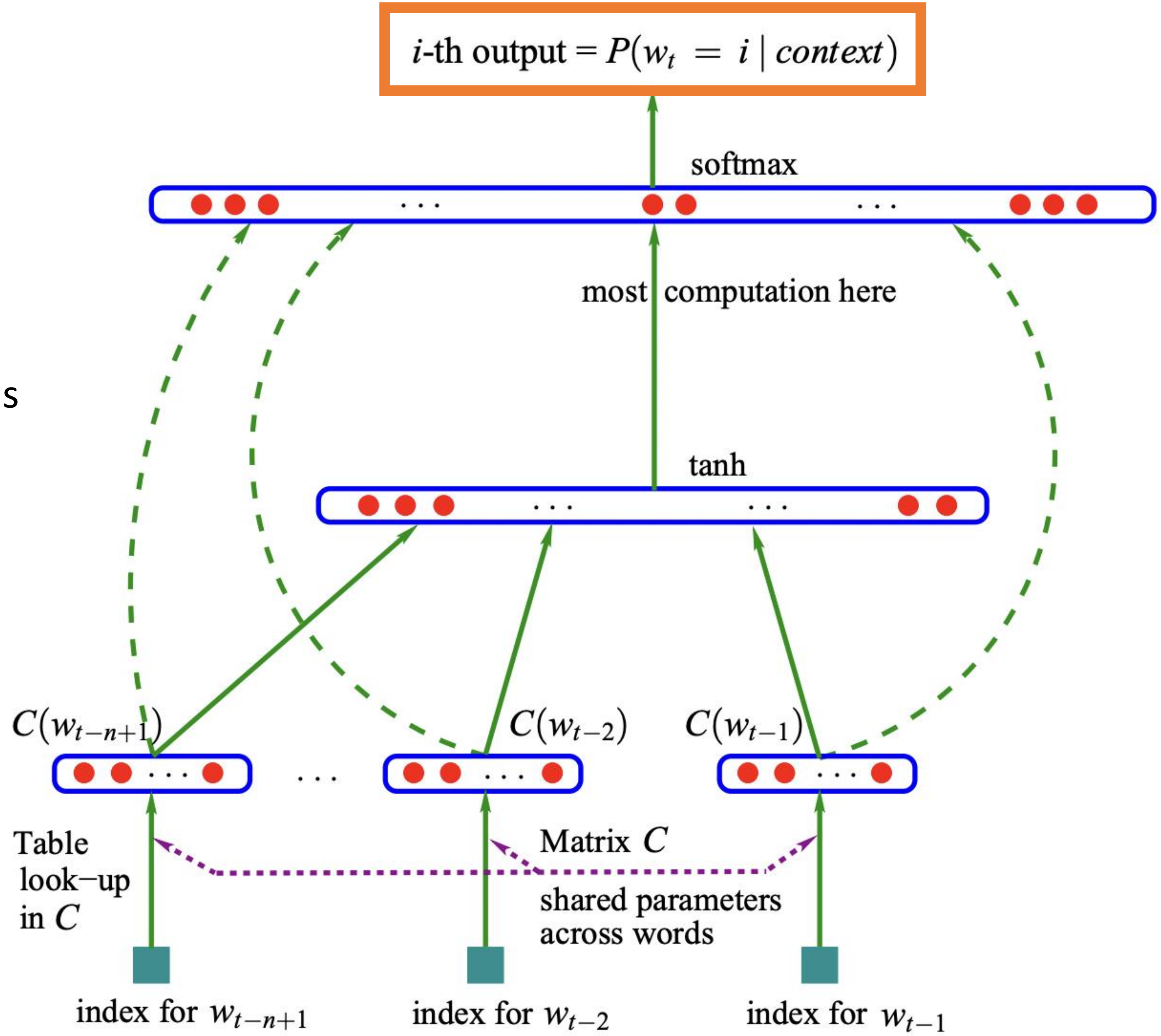
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Training

Use sliding window on input data; e.g., 3 words

Background music from a berimbau offers a beautiful escape…

Input: tried 1, 3, 5, and 8 input words and used 2 datasets with ~1 million and ~34 million words respectively



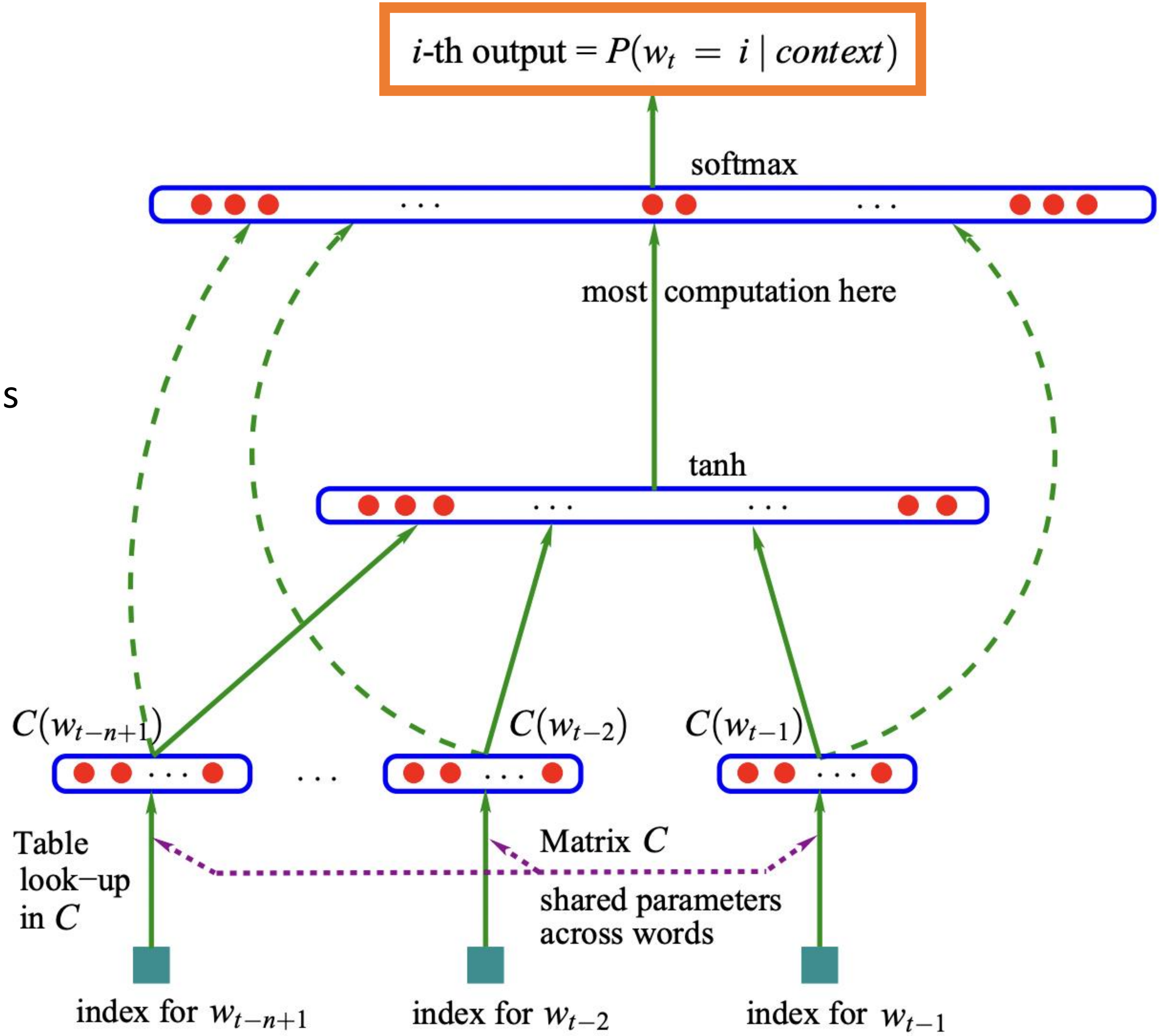
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Training

Use sliding window on input data; e.g., 3 words

Background music from a berimbau offers a beautiful escape…

Input: tried 1, 3, 5, and 8 input words and used 2 datasets with ~1 million and ~34 million words respectively



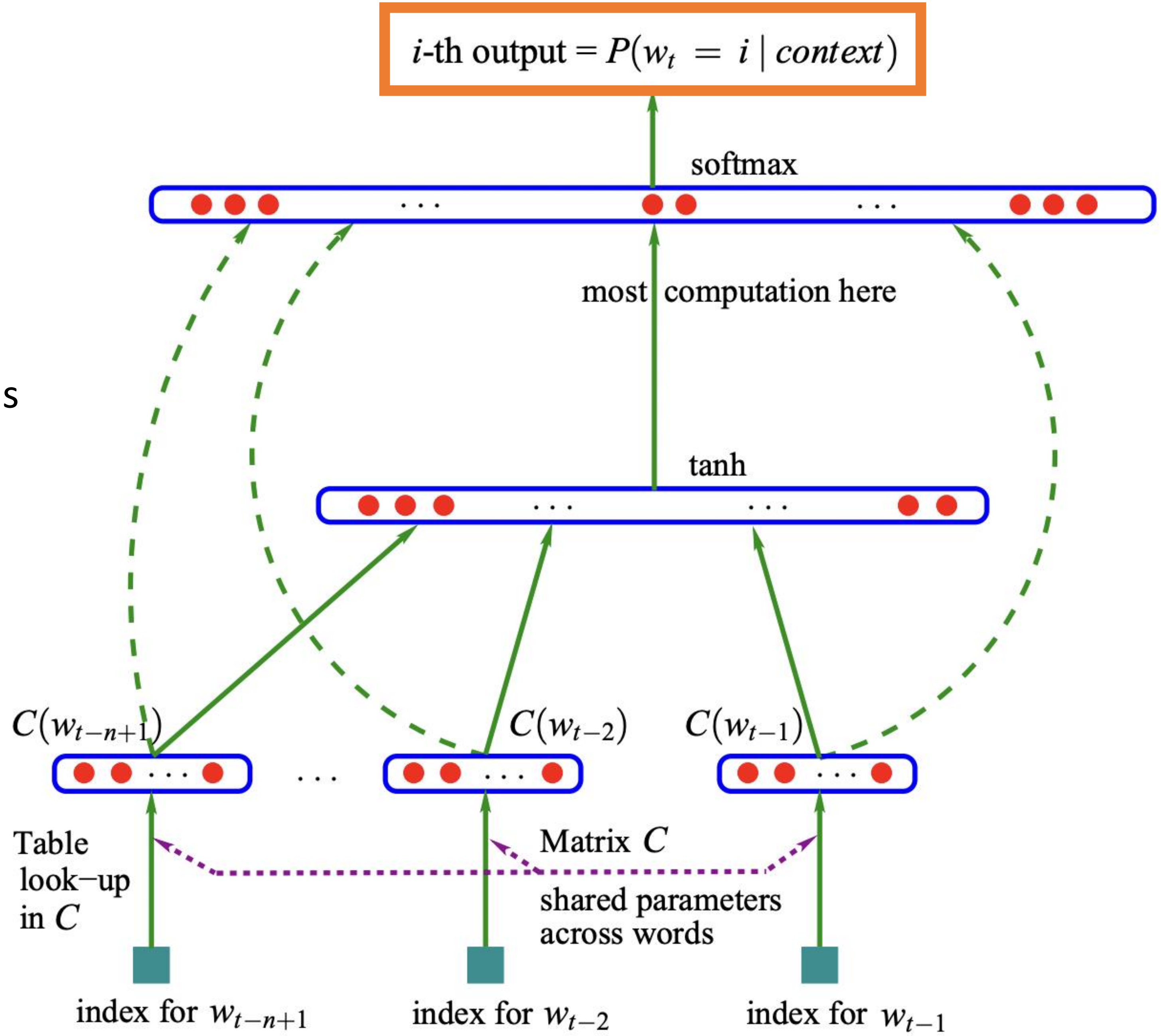
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Training

Use sliding window on input data; e.g., 3 words

Background music from a berimbau offers a beautiful escape…

Input: tried 1, 3, 5, and 8 input words and used 2 datasets with ~1 million and ~34 million words respectively



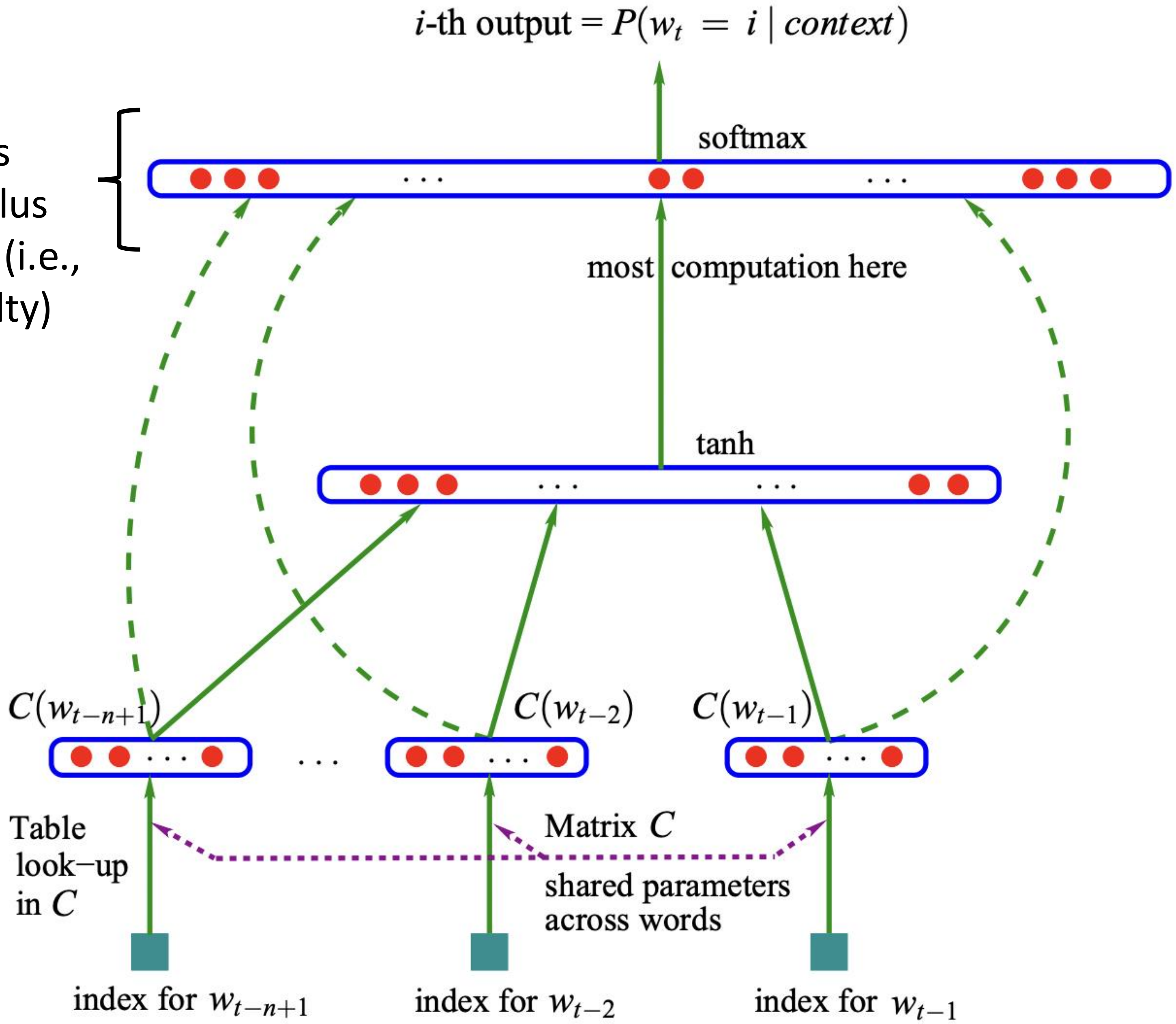
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Training

Cost function: minimize cross entropy loss plus regularization (i.e., L2 norm penalty)

Word embedding iteratively updated

Input: tried 1, 3, 5, and 8 input words and used 2 datasets with ~1 million and ~34 million words respectively



Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Summary: Word Embeddings Learn Context of Previous Words Needed to Predict Next Word

e.g.,

1. Background music from a ________
2. Many people danced around the ________
3. I practiced for many years to learn how to play the ________

Popular Word Embeddings

- Bengio method
- Word2vec (skip-gram model)
- And more…

Idea: Learn Word Embeddings That Know What Are Viable Surrounding Words

e.g.,

1. ___ ___ ___ ___ berimbau ___ ___ ___ ___
2. ___ berimbau ___

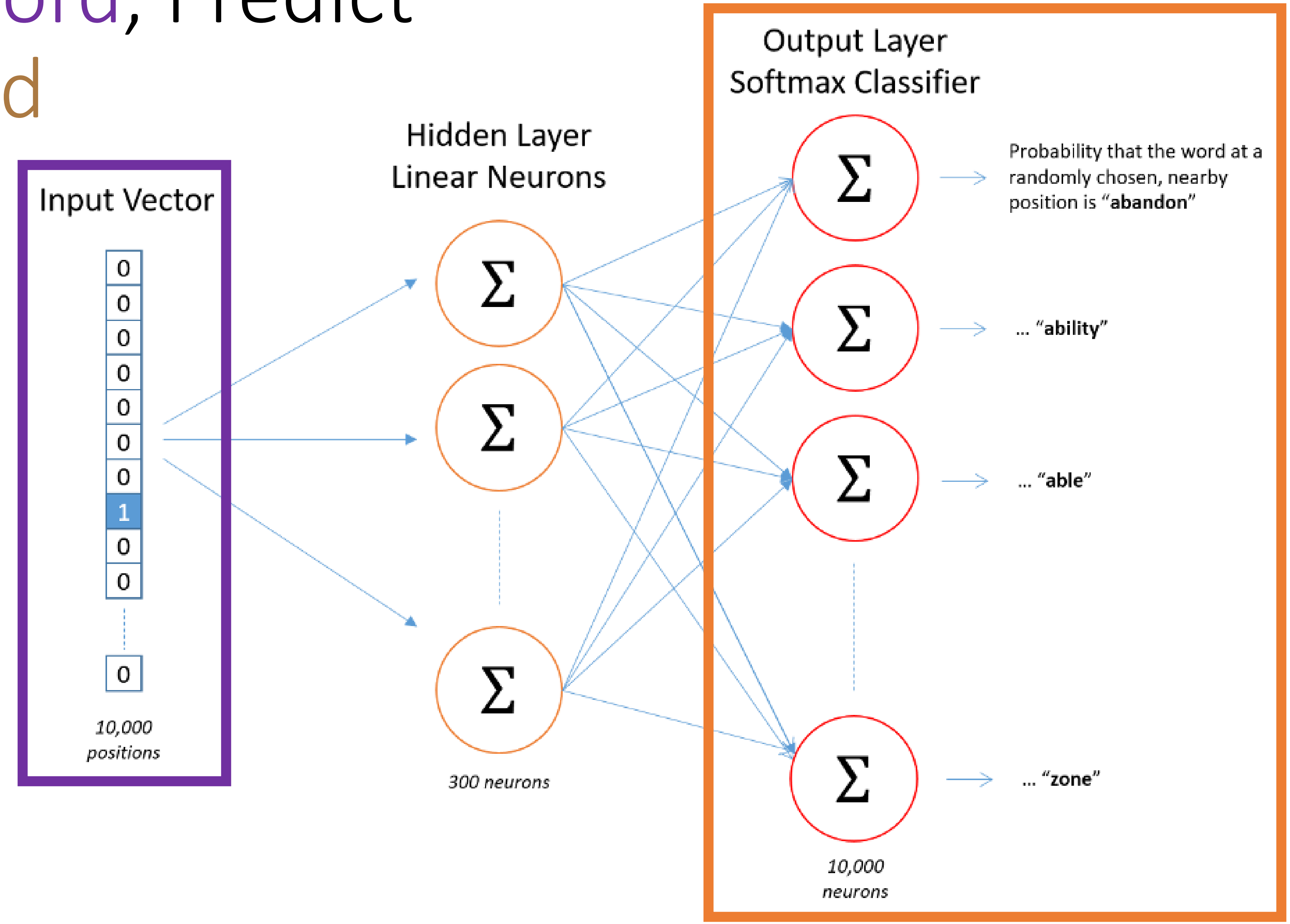
Mikolov et al. Efficient Estimation of Word Representations in Vector Space. arXiv 2013.

Task: Given Word, Predict a Nearby Word

e.g.,

1. ___ ___ ___ ___ berimbau ___ ___ ___ ___
2. ___ berimbau ___

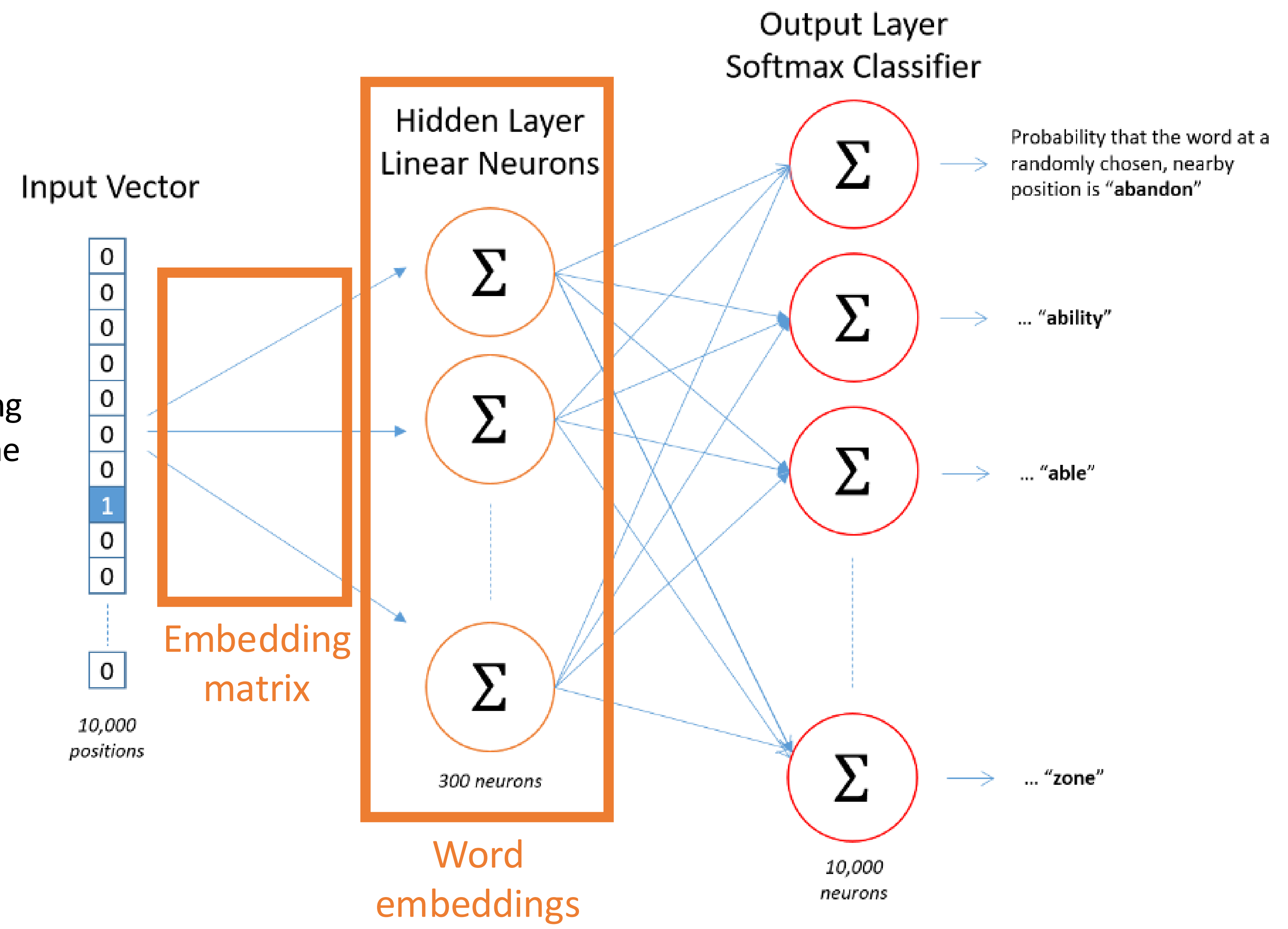
Task: Given Word, Predict a Nearby Word



https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b

Architecture

Recall: the goal is to learn an embedding matrix and, after training, the rest of the neural network can be discarded



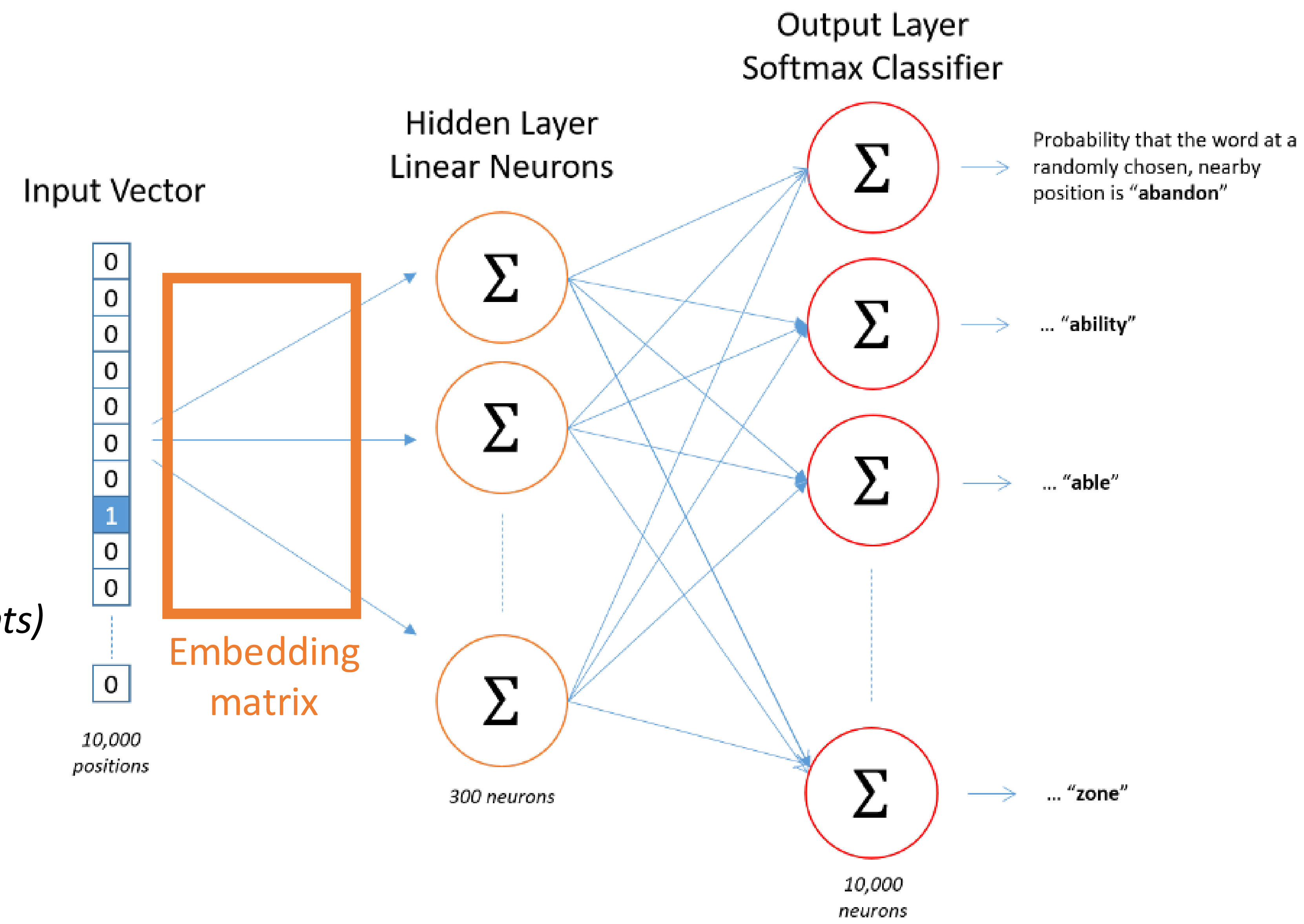
https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b

Architecture

e.g., a vocabulary size of 10,000 is used with embedding sizes of 300

What are the dimensions of the embedding matrix?

300 x 10,000 (i.e., 3,000,000 weights)



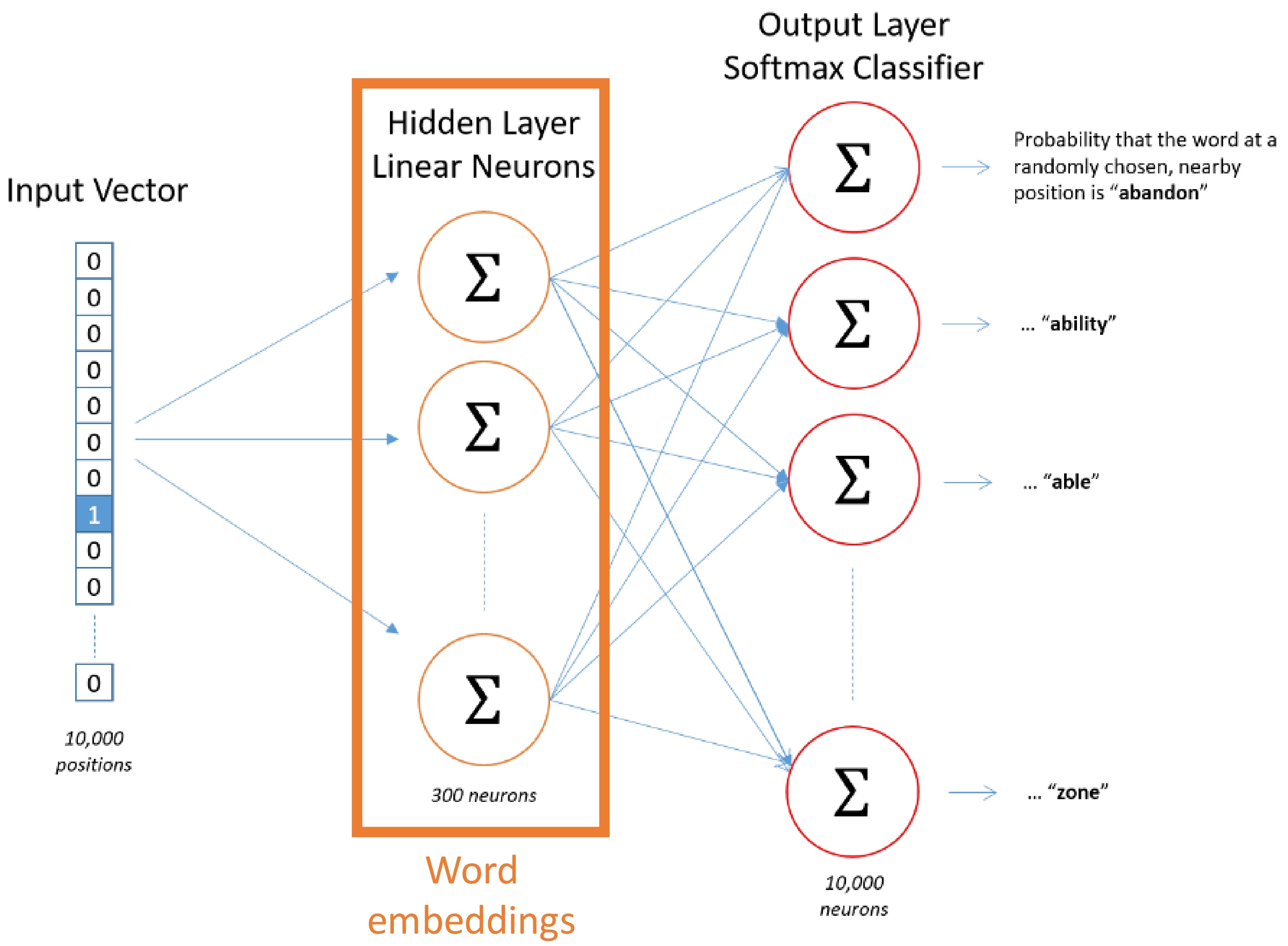
https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b

Architecture

e.g., a vocabulary size of 10,000 is used with embedding sizes of 300

What are the dimensions of each word embedding?

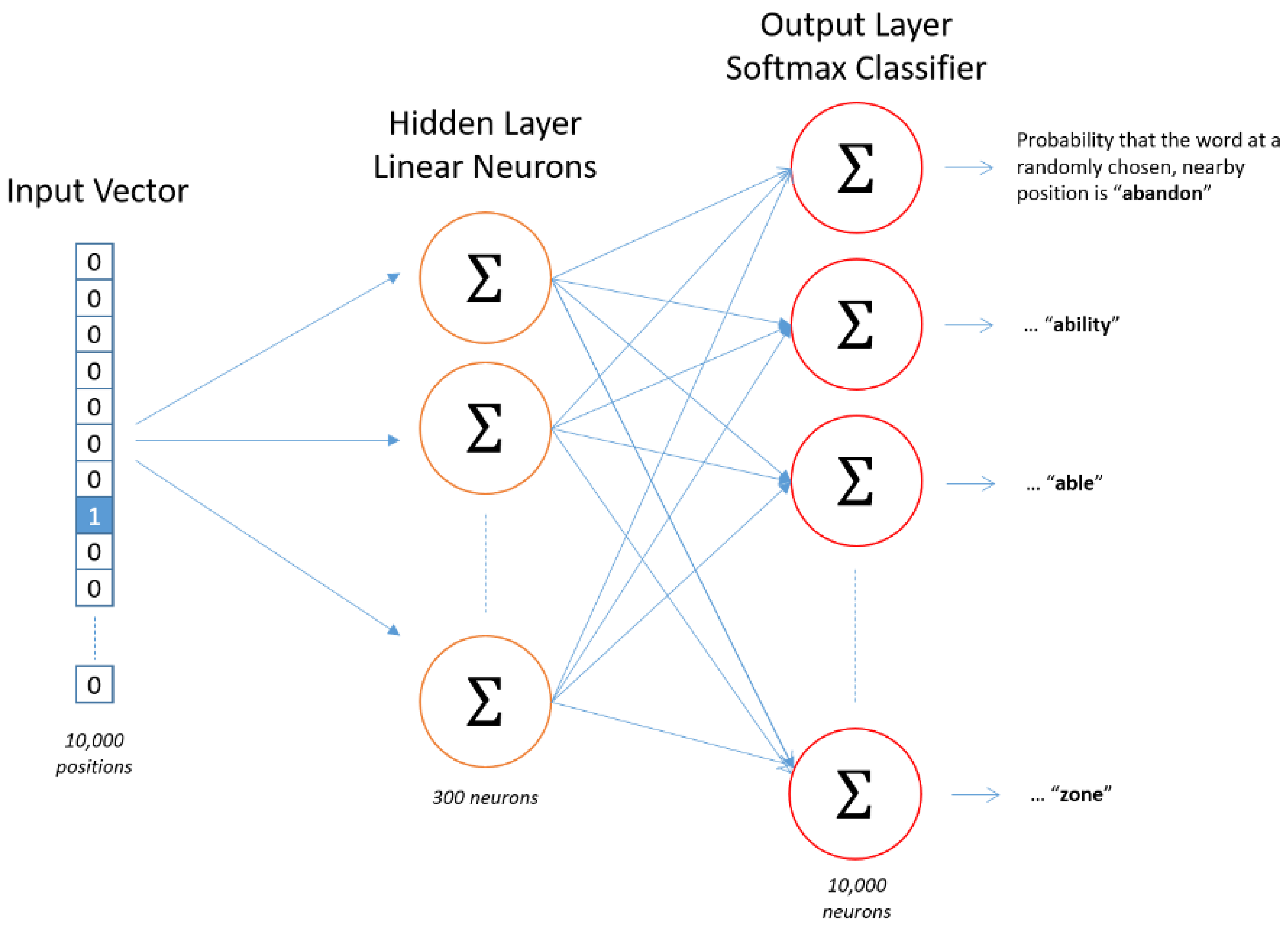
1 x 300



https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b

Architecture

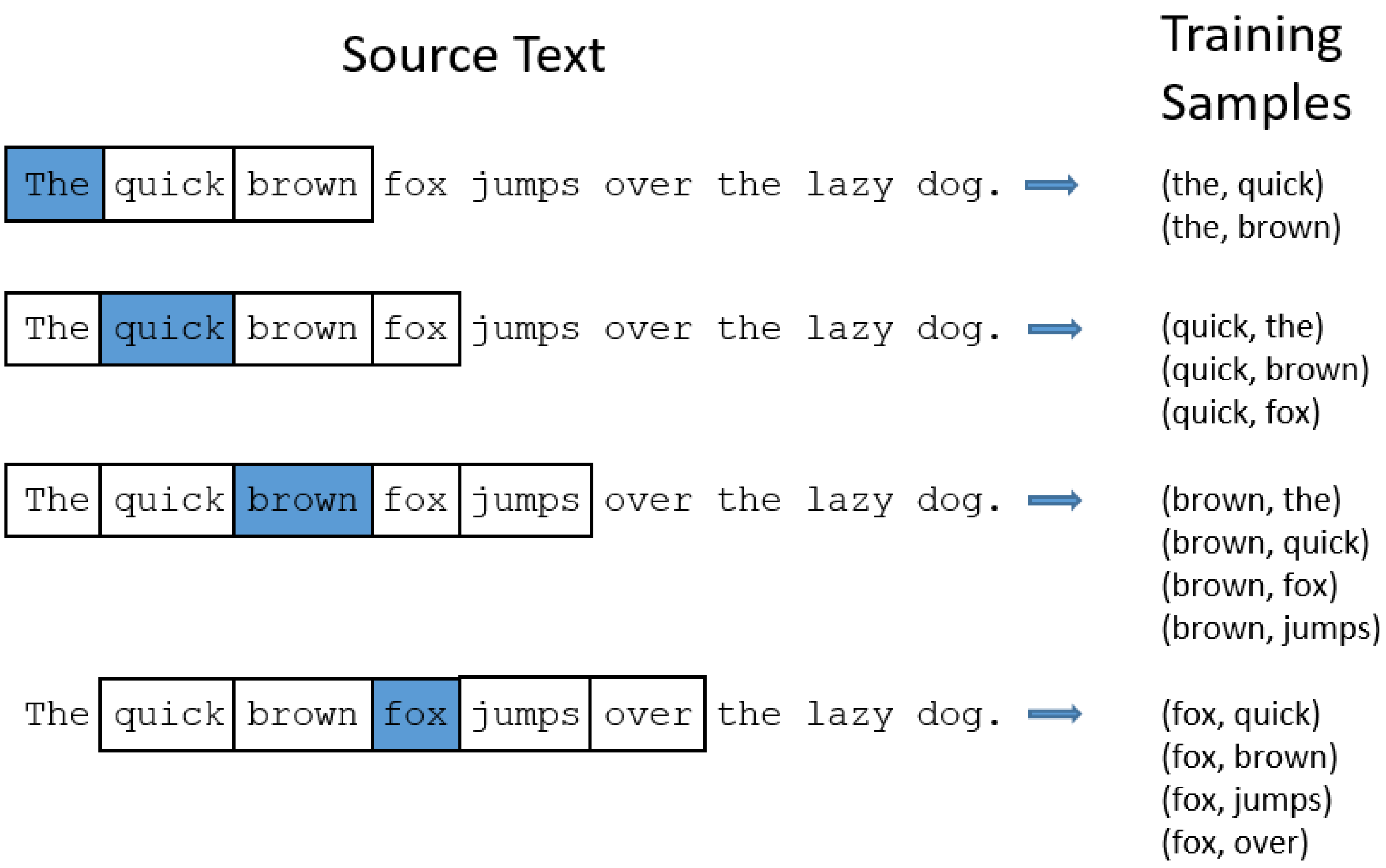
A shallower, simpler architecture than the Bengio approach (i.e., lacks a non-linear hidden layer)!



https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b

Training

Sliding window run over input to sample neighbors of each target word



https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b

Hyperparameters: What Works Well?

- Word embedding dimensionality?
  - Dimensionality set between 100 and 1,000

- Context window size?
  - ~10

Mikolov et al. Efficient Estimation of Word Representations in Vector Space. arXiv 2013.

Very Exciting/Surprising Finding

- Vector arithmetic with word embeddings can solves many analogies

  (Full test list: http://download.tensorflow.org/data/questions-words.txt)

- Semantic relationships (meaning of words in a sentence):
  - Italy + (Paris - France) = Rome

- Syntactic relationships (rules for words in a sentence)
  - smallest + (big – small) = biggest
  - think + (read – reading) = thinking
  - mouse + (dollars – dollar) = mice

Mikolov et al. Efficient Estimation of Word Representations in Vector Space. arXiv 2013.

Summary: Word Embeddings Are Learned that Support Predicting Viable Surrounding Words!

e.g.,

1. ___ ___ ___ ___ berimbau ___ ___ ___ ___
2. ___ berimbau ___

Popular Word Embeddings

- Bengio method
- Word2vec (skip-gram model)
- And more…

Variants for Learning Embeddings

- Capture global context rather than just local context of previous or surrounding words; e.g.,
  - GloVe for Global Vectors (Pennington et al., 2014)
- Capture that the same word can have different meanings under different contexts; e.g., The bat… …swung? …flew?
  - Elmo was a pioneer for language models (Peters et al., arXiv 2018)
- Support multiple languages; e.g.,
  - Fast-text (Bojanowski et al., 2016)

Word Embedding Limitations/Challenges

- Distinguish antonyms from synonyms
  - Antonyms are often close in embeddings space since they often occur in similar contexts: “I hate math” vs “I love math” or “Turn right” vs “Turn left”

- Gender bias:

Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings

Tolga Bolukbasi[1], **Kai-Wei Chang**[2], **James Zou**[2], **Venkatesh Saligrama**[1,2], **Adam Kalai**[2]
[1]Boston University, 8 Saint Mary’s Street, Boston, MA
[2]Microsoft Research New England, 1 Memorial Drive, Cambridge, MA
tolgab@bu.edu, kw@kwchang.net, jamesyzou@gmail.com, srv@bu.edu, adam.kalai@microsoft.com

Word Embedding Limitations/Challenges

- Distinguish antonyms from synonyms
  - Antonyms are often close in embeddings space since they often occur in similar contexts: “I hate math” vs “I love math” or “Turn right” vs “Turn left”

- Gender bias:

Extreme *she*	**Extreme *he***
1. homemaker	1. maestro
2. nurse	2. skipper
3. receptionist	3. protege
4. librarian	4. philosopher
5. socialite	5. captain
6. hairdresser	6. architect
7. nanny	7. financier
8. bookkeeper	8. warrior
9. stylist	9. broadcaster
10. housekeeper	10. magician

Gender stereotype *she*-*he* analogies

sewing-carpentry	registered nurse-physician	housewife-shopkeeper
nurse-surgeon	interior designer-architect	softball-baseball
blond-burly	feminism-conservatism	cosmetics-pharmaceuticals
giggle-chuckle	vocalist-guitarist	petite-lanky
sassy-snappy	diva-superstar	charming-affable
volleyball-football	cupcakes-pizzas	lovely-brilliant

Gender appropriate *she*-*he* analogies

queen-king	sister-brother	mother-father
waitress-waiter	ovarian cancer-prostate cancer	convent-monastery

Bolukbasi et al. Neurips 2016.

Word Embedding Limitations/Challenges

- Distinguish antonyms from synonyms
  - Antonyms are often close in embeddings space since they often occur in similar contexts: “I hate math” vs “I love math” or “Turn right” vs “Turn left”

- Gender bias

- What additional language biases do you think could be learned?

- Still not as compact as phrase-level or sentence-level embeddings

Today’s Topics

- Motivation: numeric representation of natural language
- Tokenization: how to convert text into discrete units
- Neural word embeddings: how to create dense representation
- Programming tutorial

Today’s Topics

- Motivation: numeric representation of natural language
- Tokenization: how to convert text into discrete units
- Neural word embeddings: how to create dense representation
- Programming tutorial


The End