Introduction to NLP and Word Embeddings

Danna Gurari
University of Colorado Boulder
Spring 2022

Review

- Last week:
  - Machine learning for sequential data
  - Recurrent neural networks (RNNs)
  - Gated RNNs
  - Programming tutorial
- Assignments (Canvas):
  - Problem set 3 due earlier today
  - Lab assignment 3 due in 1.5 weeks
- Questions?

Today’s Topics

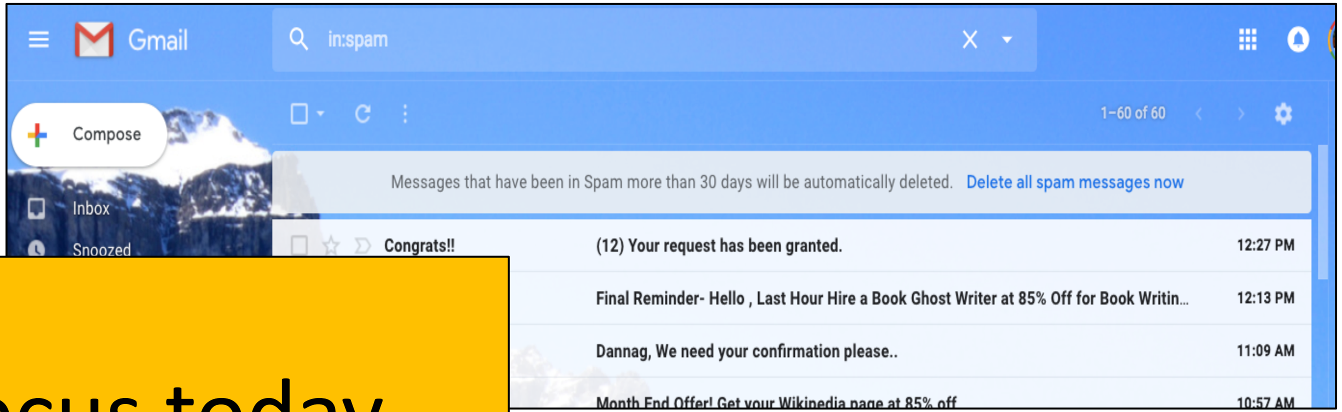
- Introduction to natural language processing
- Text representation
- Neural word embeddings
- Programming tutorial

Today's Topics

- Introduction to natural language processing
- Text representation
- Neural word embeddings
- Programming tutorial

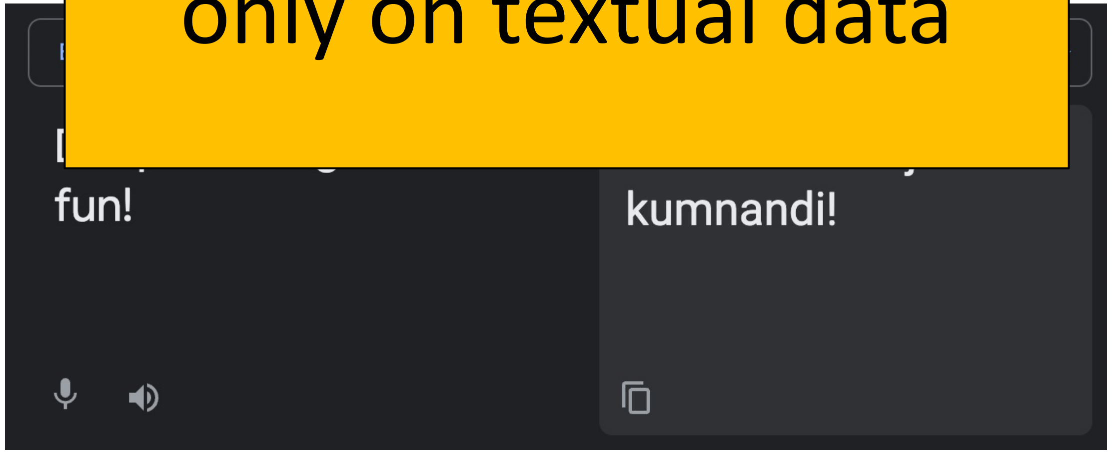
NLP: Computers that Can Understand (and So Also Communicate in) Human Language

Opinion Mining



Spam Detection

We will focus today only on textual data

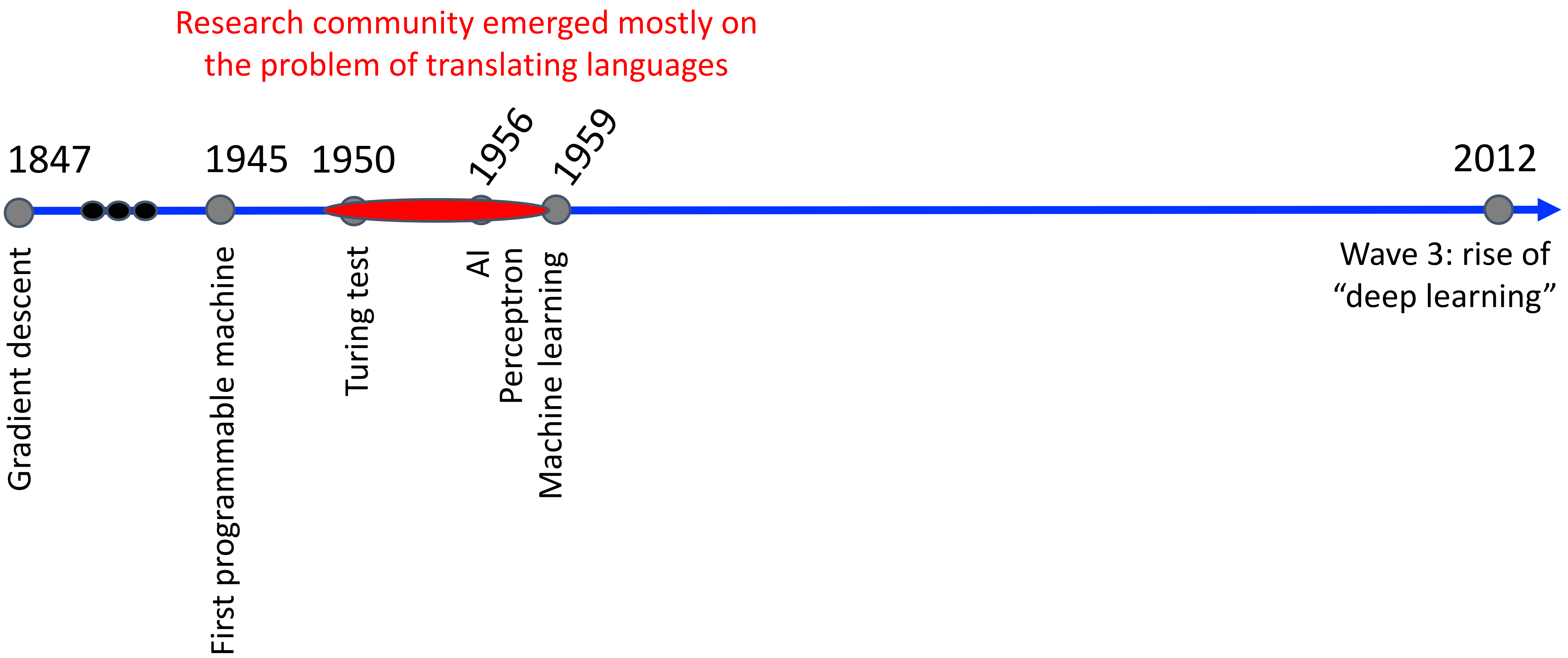


Language Translation

Why Discuss NLP With RNNs?

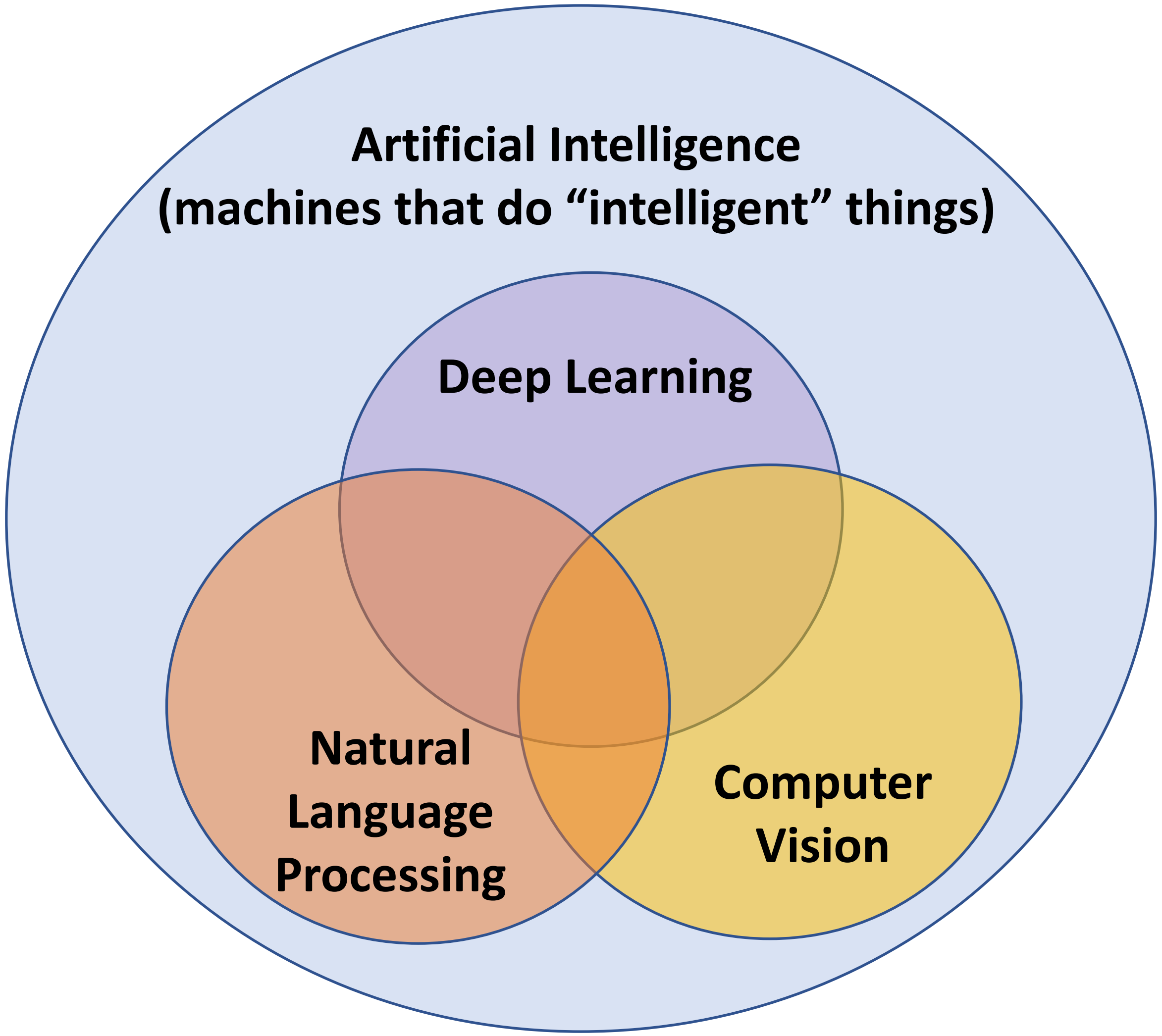
- RNNs have a strong track record for NLP problems
- Text data's representation (i.e., sequential data) is a natural match for RNNs

Historical Context: Origins of NLP



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

NLP in Context



Key Challenge: Replicate Language Understanding for **So Many Tasks**!

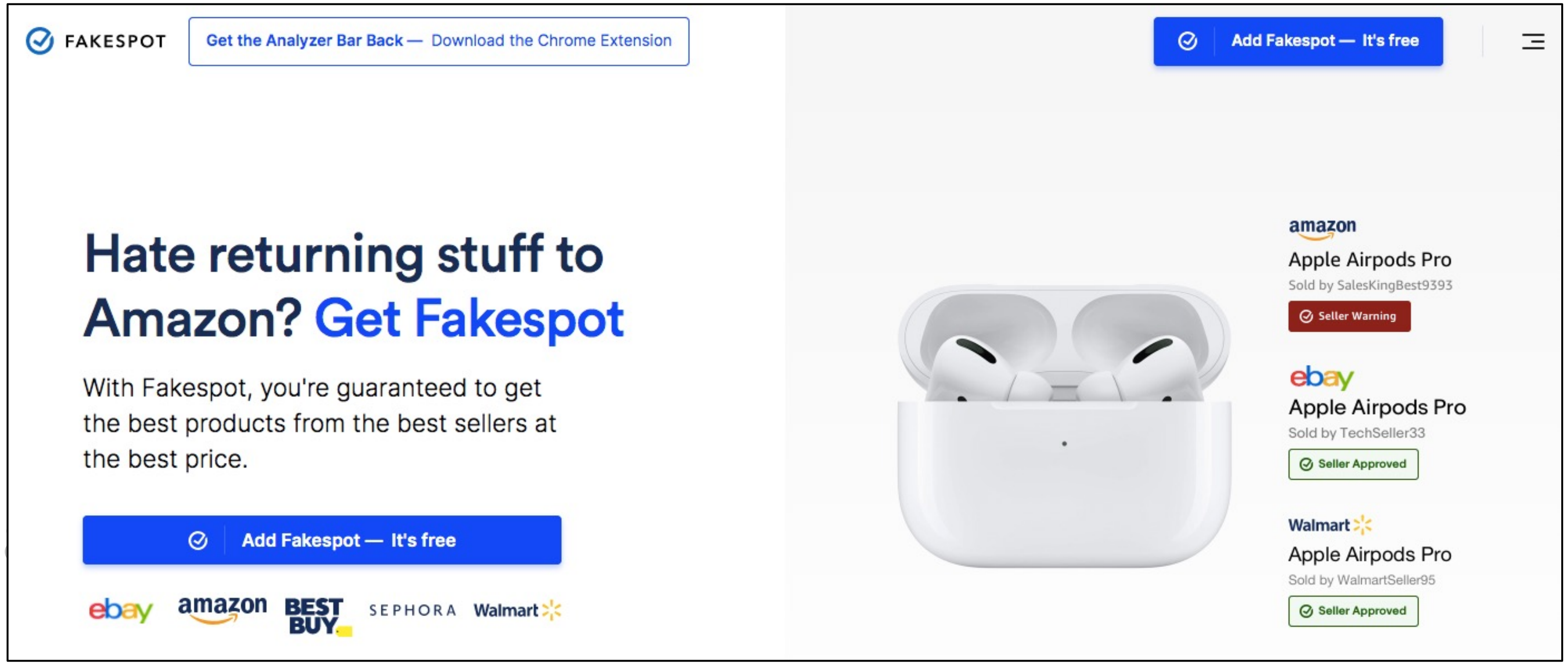
- Text classification
- Machine translation
- Question answering
- Automatic summarization
- And more…

Key Challenge: Replicate Language Understanding for **So Many Tasks**!

- Text classification
- Machine translation
- Question answering
- Automatic summariza
- And more…

Key Challenge: Replicate Language Understanding for **So Many Tasks**!

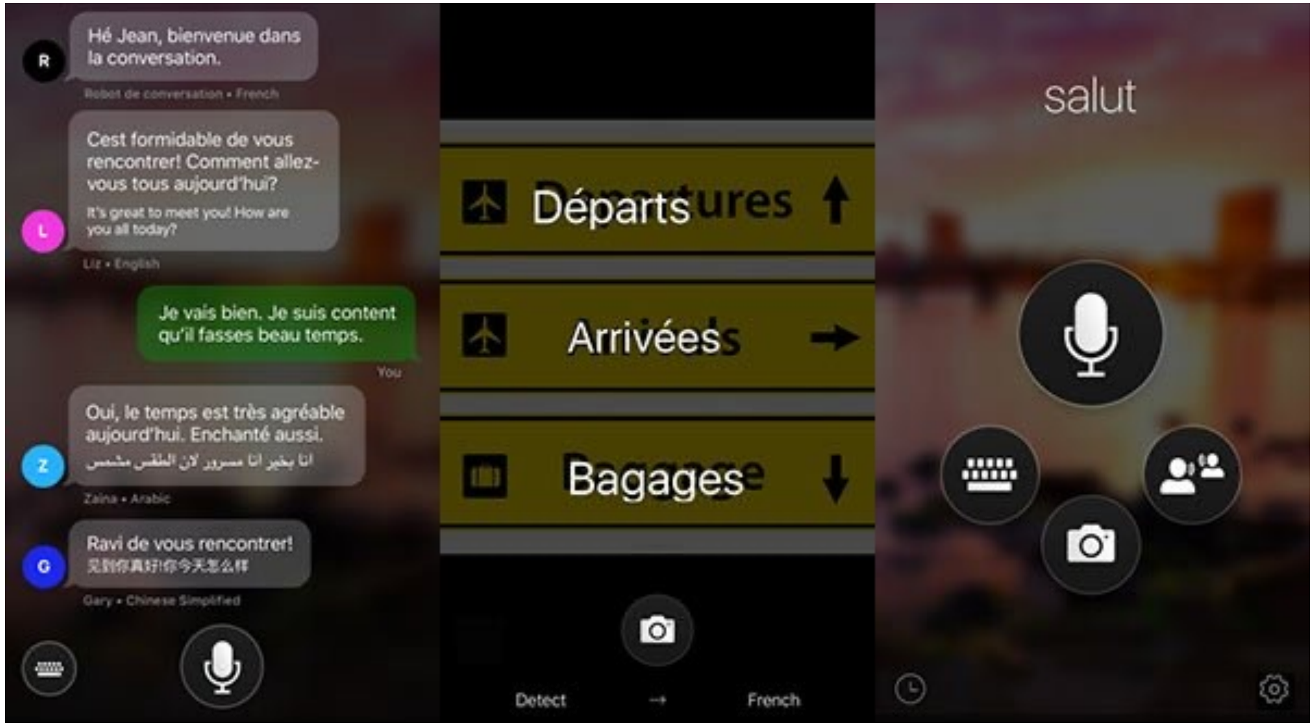
- Text classification
- Machine translation
- Question answering
- Automatic summariz
- And more…



Key Challenge: Replicate Language Understanding for **So Many Tasks**!

- Text classification
- Machine translation
- Question answering
- Automatic summarization
- And more…

e.g., Microsoft translator

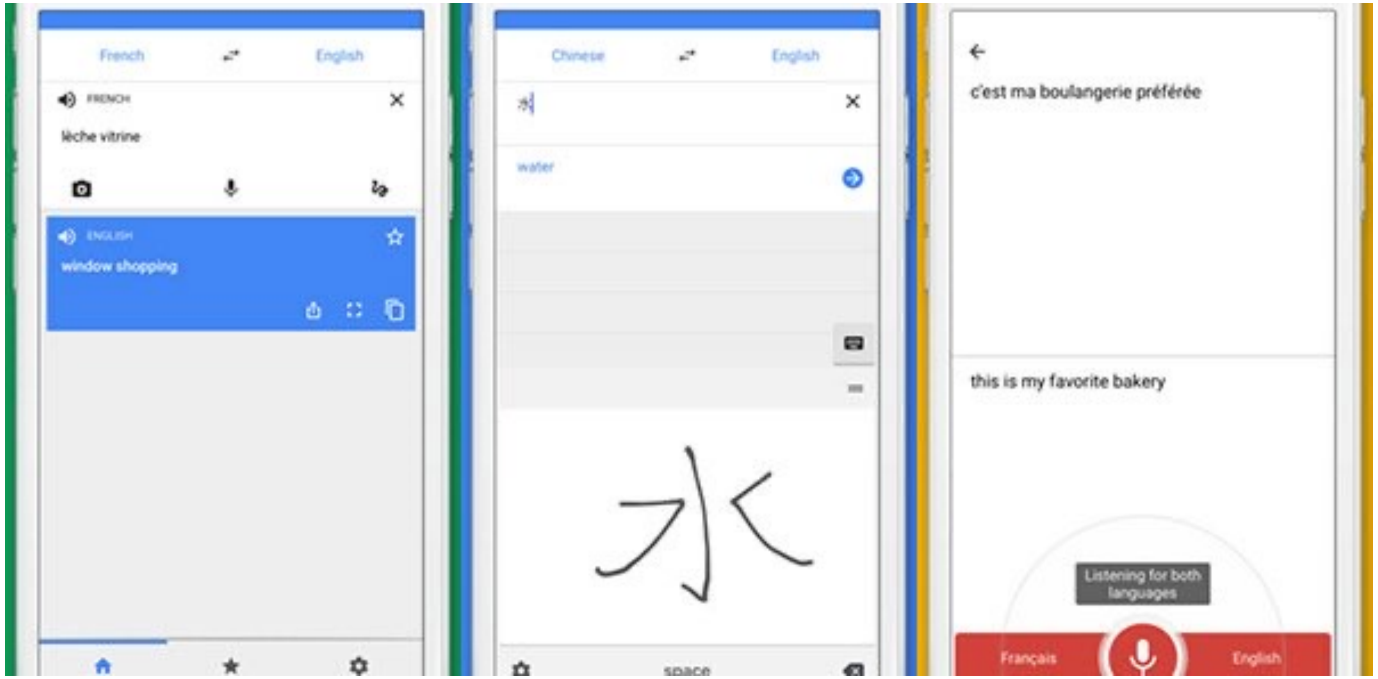


https://uncubed.com/daily/best-translation-apps-for-travel-in-2019/

Key Challenge: Replicate Language Understanding for **So Many Tasks**!

- Text classification
- Machine translation
- Question answering
- Automatic summarization
- And more…

e.g., Google translate



https://uncubed.com/daily/best-translation-apps-for-travel-in-2019/

Key Challenge: Replicate Language Understanding for **So Many Tasks**!

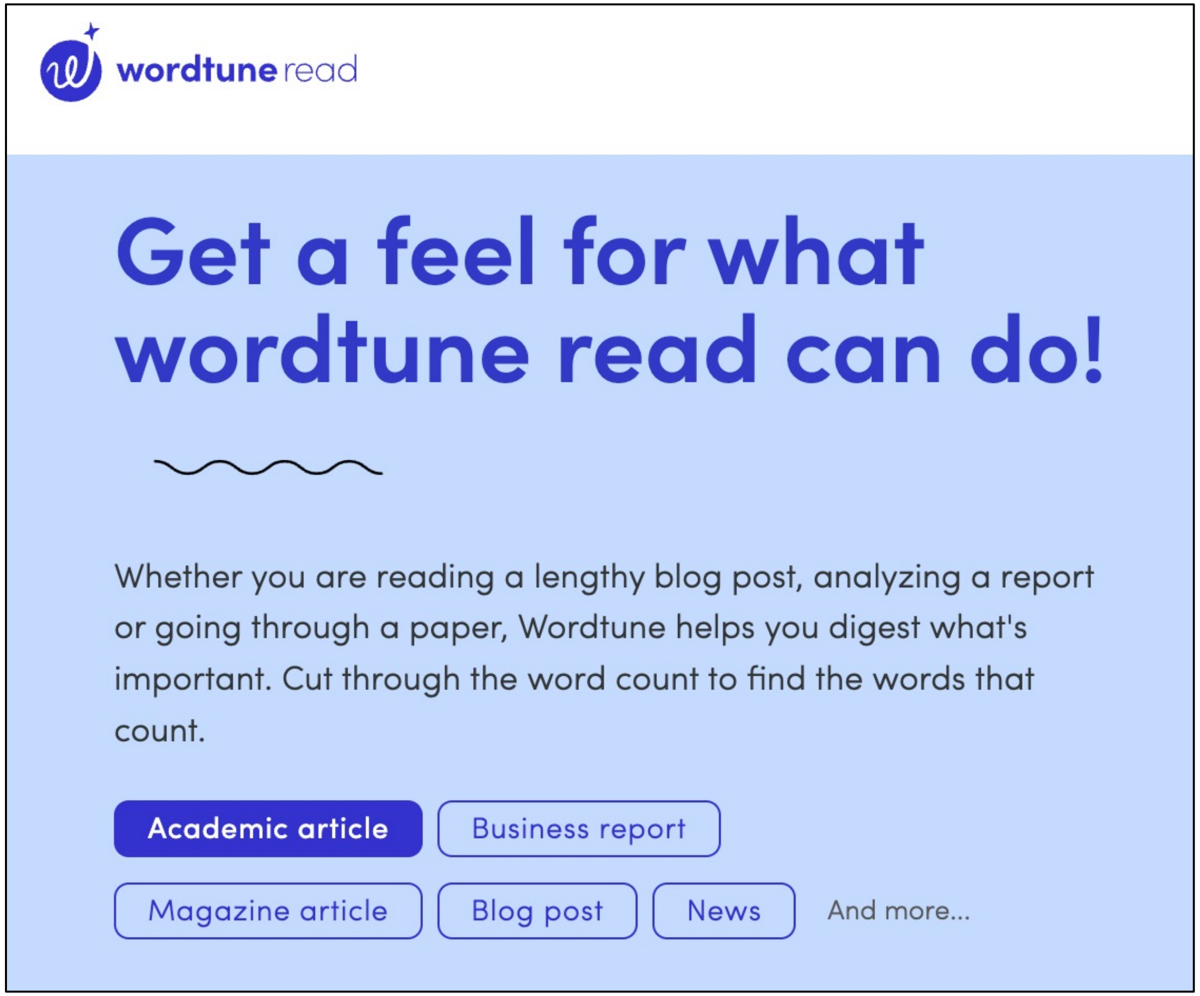
- Text classification
- Machine translation
- Question answering
- Automatic summarization
- And more…

e.g., IBM Watson question answering system (and Jeopardy winner)

https://www.nytimes.com/2011/02/17/science/17jeopardy-watson.html

Key Challenge: Replicate Language Understanding for **So Many Tasks**!

- Text classification
- Machine translation
- Question answering
- Automatic summarization
- And more…



Key Challenge: Replicate Language Understanding for **So Many Tasks**!

- Text classification
- Machine translation
- Question answering
- Automatic summarization
- And more…

Other Key Challenges: Replicate Language Understanding for So Many Languages/Individuals!

- Need a computable characterization of all human languages that simultaneously captures nuances from individuals; e.g., 7000+ languages spoken around the world

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

Today’s Topics

- Introduction to natural language processing
- Text representation
- Neural word embeddings
- Programming tutorial

Input: String (Collection of Characters)

Most Relevant

Lives in Austin, Texas

Keith C. McCormic Let the food pantries have it instead of monetizing it.

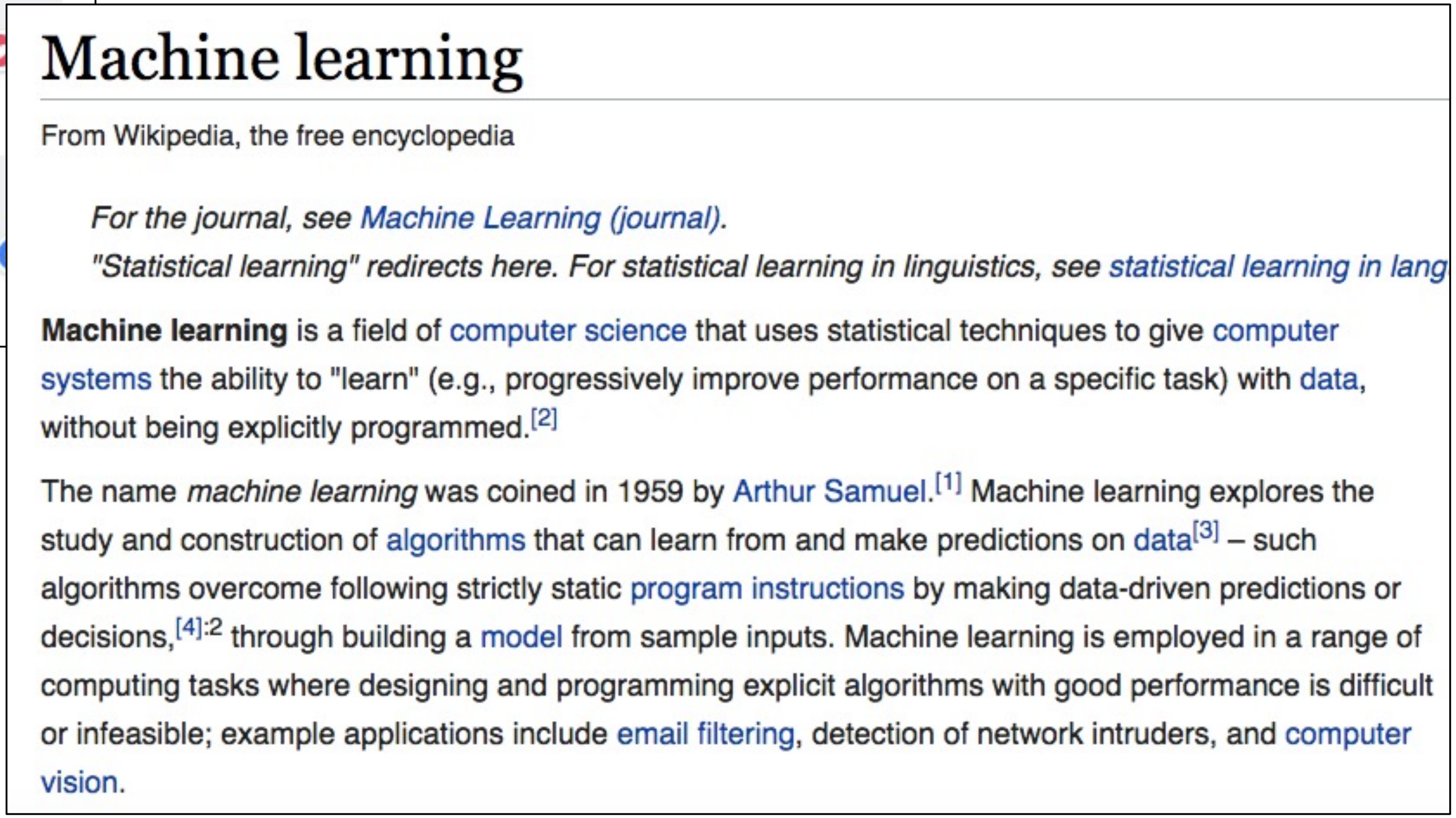
Like · Reply · 1d

5 Replies

Caty O'Neil Webb The promo code isn't working but I found another one on line GETFIFTY% .

Like · Reply · 1d · Edited

2 Replies



Machine learning

From Wikipedia, the free encyclopedia

For the journal, see Machine Learning (journal).

"Statistical learning" redirects here. For statistical learning in linguistics, see statistical learning in lang

Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" (e.g., progressively improve performance on a specific task) with data, without being explicitly programmed.[2]

The name *machine learning* was coined in 1959 by Arthur Samuel.[1] Machine learning explores the study and construction of algorithms that can learn from and make predictions on data[3] – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions,[4]:2 through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders, and computer vision.

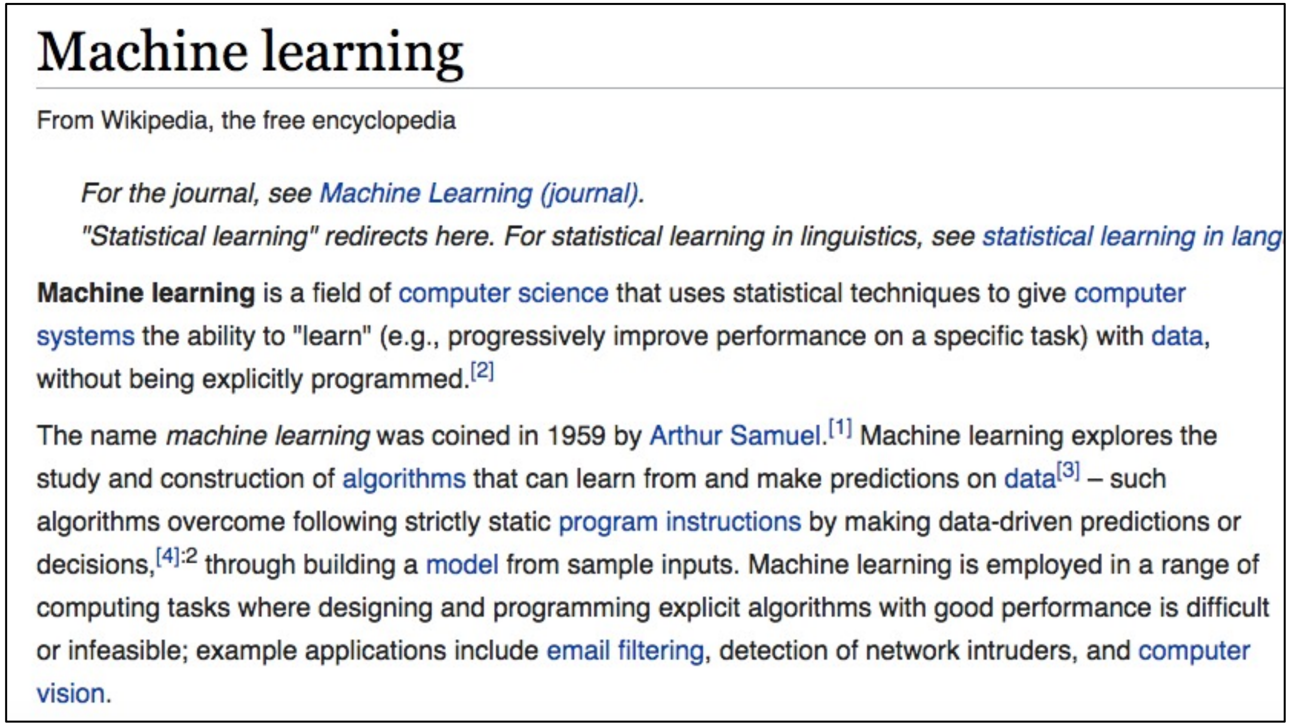
- Common terms
  - **Corpus:** dataset
  - **Document:** example

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 Describe Text to a Computer?

- Challenge: input often varies in length



Machine learning

From Wikipedia, the free encyclopedia

For the journal, see Machine Learning (journal).
"Statistical learning" redirects here. For statistical learning in linguistics, see statistical learning in lang

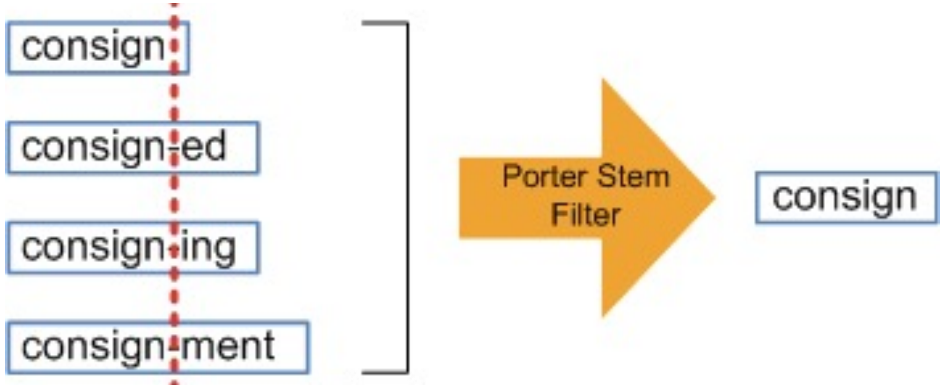
Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" (e.g., progressively improve performance on a specific task) with data, without being explicitly programmed.[2]

The name *machine learning* was coined in 1959 by Arthur Samuel.[1] Machine learning explores the study and construction of algorithms that can learn from and make predictions on data[3] – such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions,[4]:2 through building a model from sample inputs. Machine learning is employed in a range of computing tasks where designing and programming explicit algorithms with good performance is difficult or infeasible; example applications include email filtering, detection of network intruders, and computer vision.

- Solution: convert text to numeric format that DL algorithms can handle

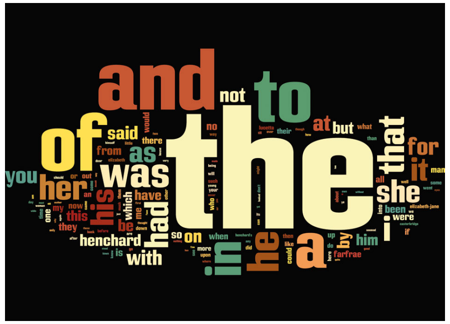
Implementation Details – Possible Pre-processing

- Lower case all letters
- Stemming: use each word's stem; e.g., singular to plural, resolve different verb forms
  - e.g.,



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

- Stop word removal: discard frequent words



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

Converting Text to Vectors

1. Tokenize training data
2. Learn vocabulary
3. Encode data as vectors

Converting Text to Vectors

1. Tokenize training data; convert data into sequence of tokens (e.g., data ->“This is tokening”)
2. Learn vocabulary
3. Encode data as vectors

Two common approaches:

Character Level

[T] [h] [i] [s] [i] [s] [t] [o] [k] [e] [n] [i] [z] [i] [n] [g] [.]

Word Level

[This] [is] [tokenizing] [.]

https://nlpiation.medium.com/how-to-use-huggingfaces-transformers-pre-trained-tokenizers-e029e8d6d1fa

Converting Text to Vectors

1. Tokenize training data
2. Learn vocabulary by identifying all unique tokens in the training data
3. Encode data as vectors

Two common approaches:

Character Level

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

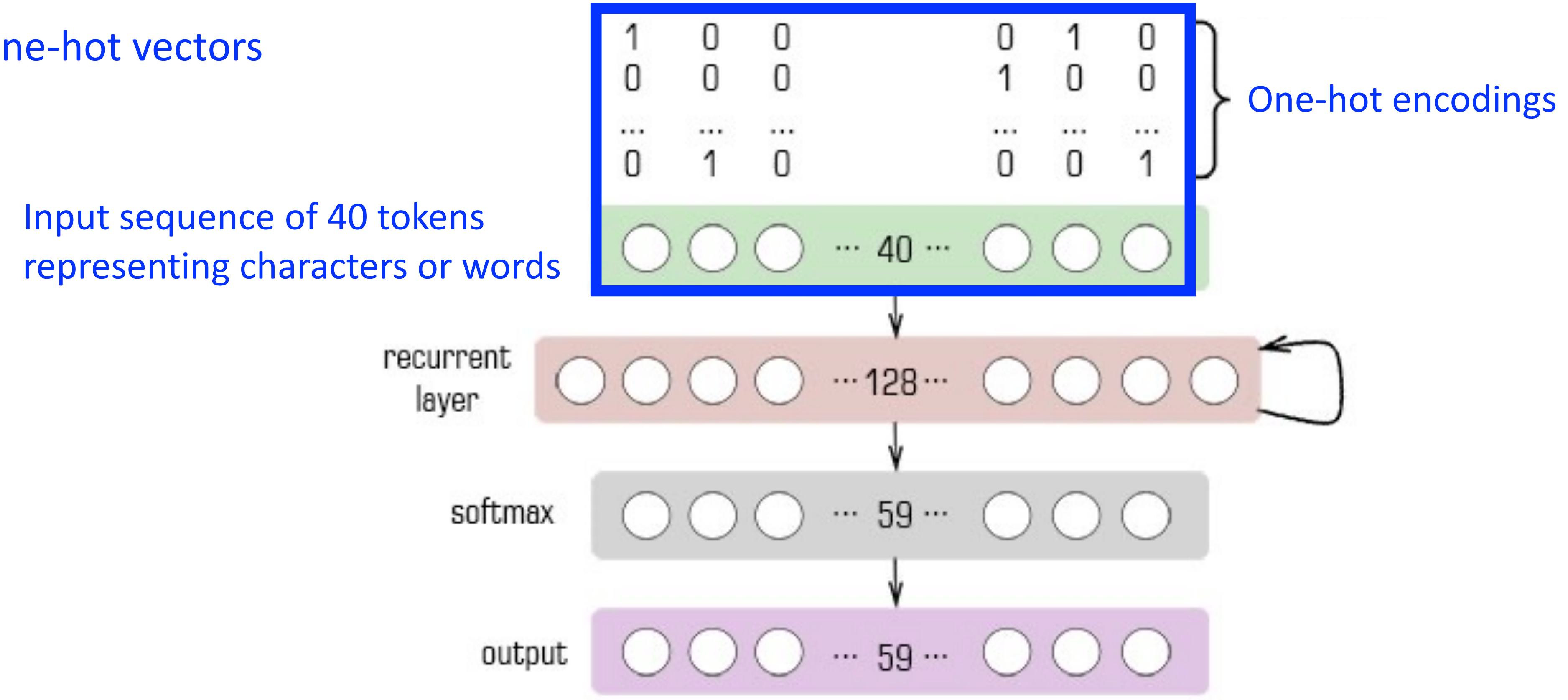
Word Level

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

https://nlpiation.medium.com/how-to-use-huggingfaces-transformers-pre-trained-tokenizers-e029e8d6d1fa

Converting Text to Vectors

1. Tokenize training data
2. Learn vocabulary by identifying all unique tokens in the training data
3. Encode data as one-hot vectors



https://github.com/DipLernin/Text_Generation

Converting Text to Vectors

What are the pros and cons for using word tokens instead of character tokens?

Character Level

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

Word Level

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

- Pros: length of input/output sequences is shorter, simplifies learning semantics
- Cons: "UNK" word token needed for out of vocabulary words; vocabulary can be large

https://nlpiation.medium.com/how-to-use-huggingfaces-transformers-pre-trained-tokenizers-e029e8d6d1fa

Converting Text to Vectors

Character Level

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

Word Level

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

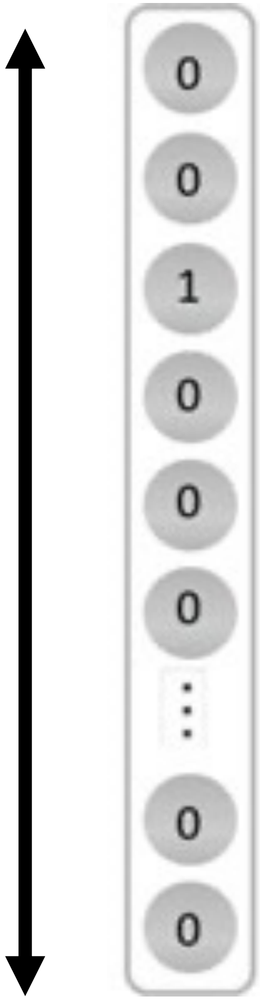
Word level representations are more commonly used

https://nlpiation.medium.com/how-to-use-huggingfaces-transformers-pre-trained-tokenizers-e029e8d6d1fa

Problems with One-Hot Encoding Words?

Dimensionality = vocabulary size

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

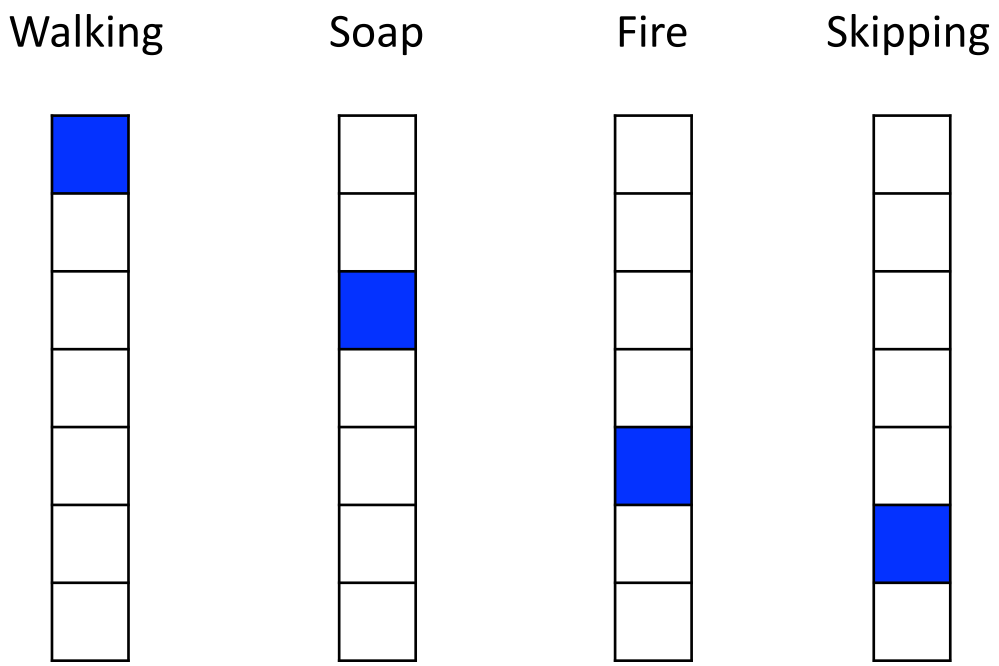


- Huge memory burden
- Computationally expensive

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.”

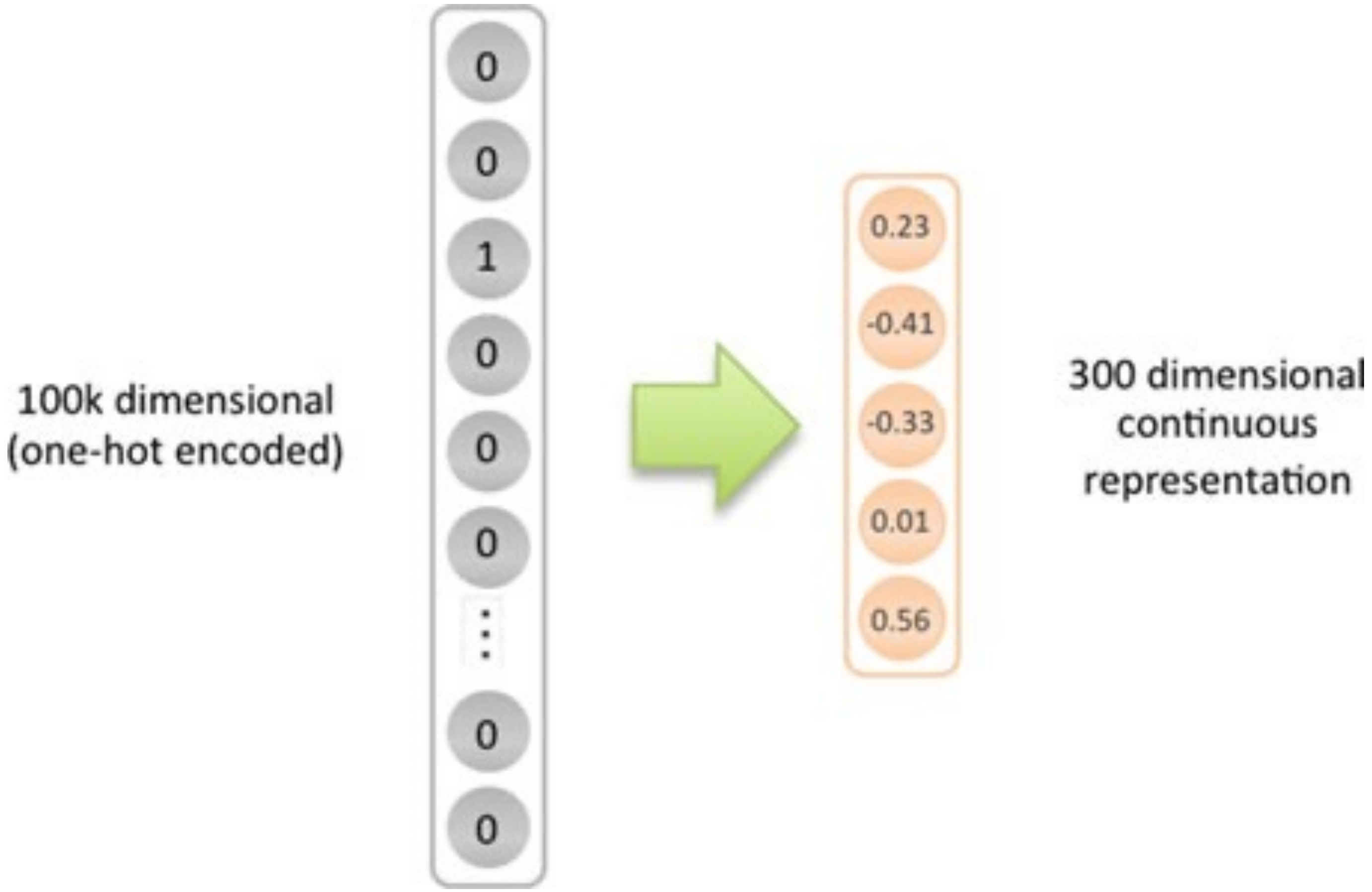


The distance between all words is equal!

Today’s Topics

- Introduction to natural language processing
- Text representation
- Neural word embeddings
- Programming tutorial

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.

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

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

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

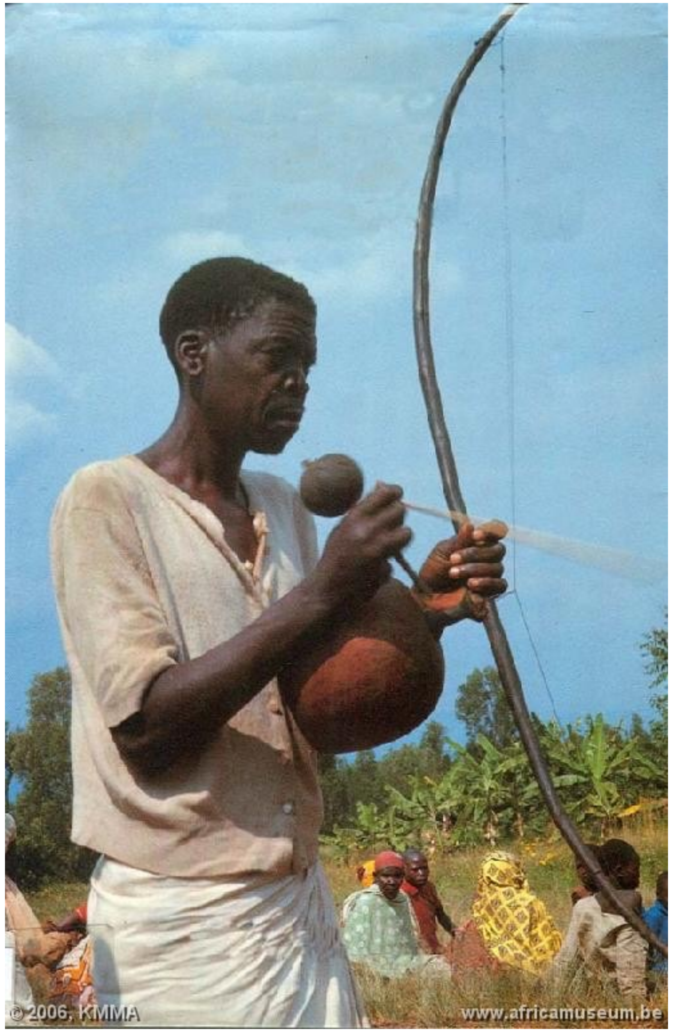
- What is the meaning of berimbau based on context?

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 context?

    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]

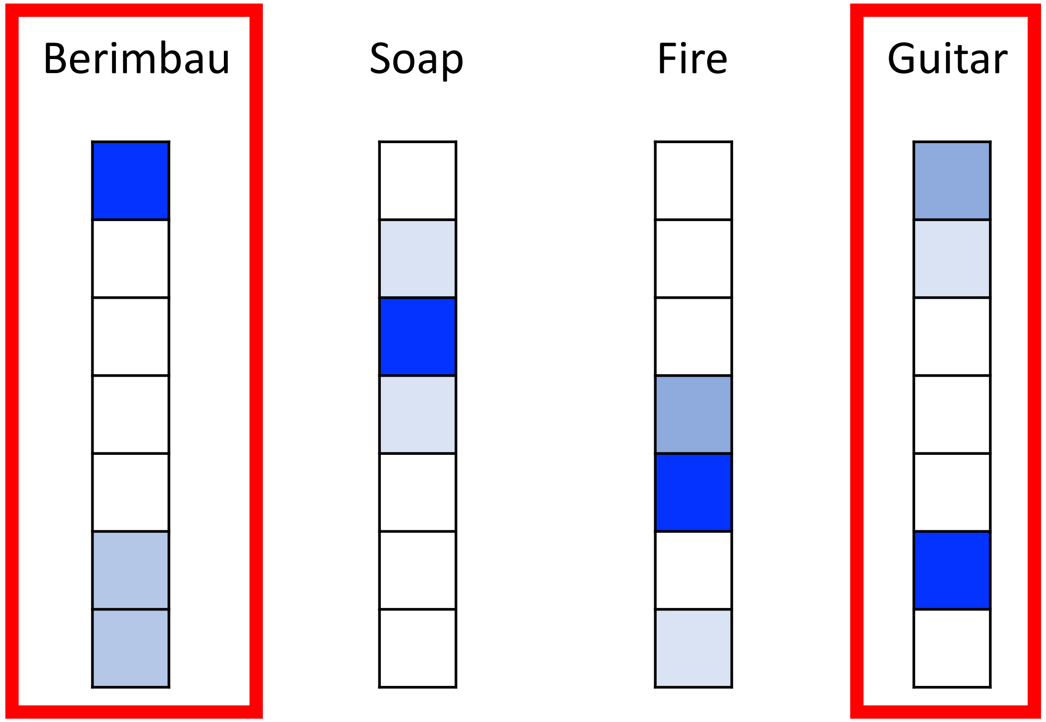
Inspiration: Distributional Semantics

“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



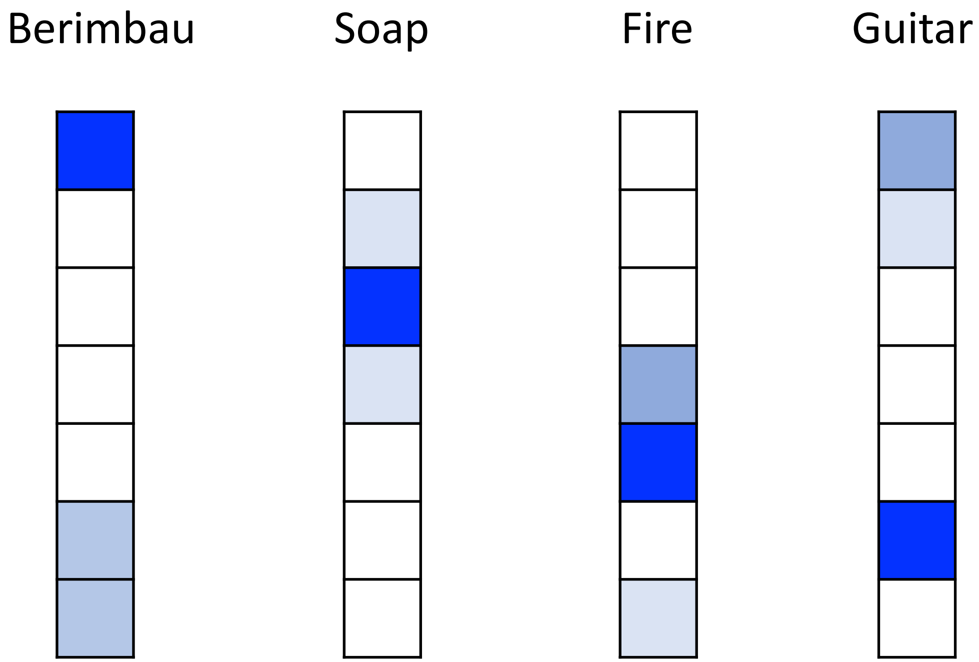
Berimbau and guitar are the closest word pair

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

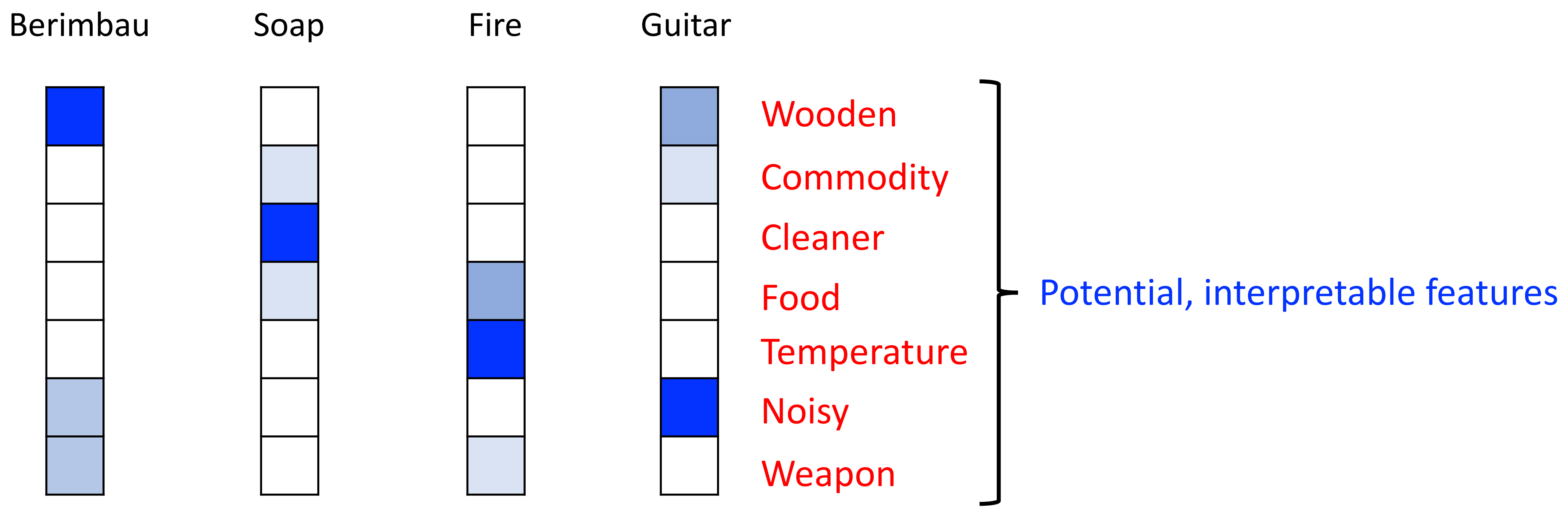


We embed words in a shared space so they can be compared with a few features

What features would discriminate these words?

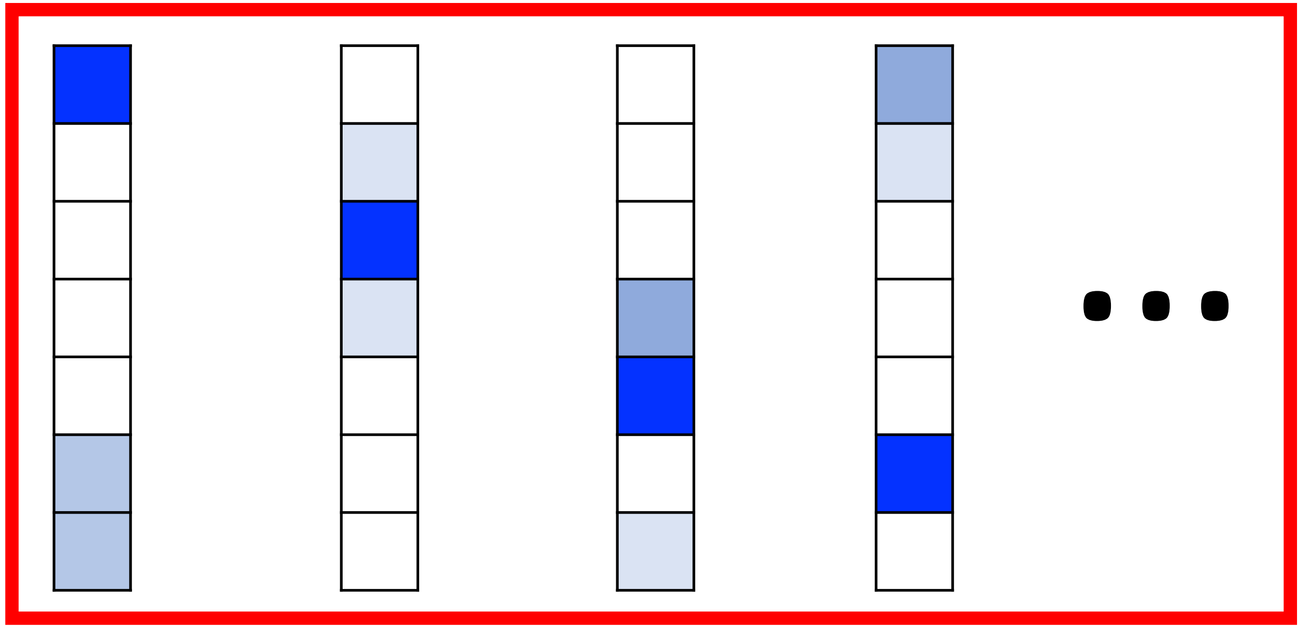
Approach

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



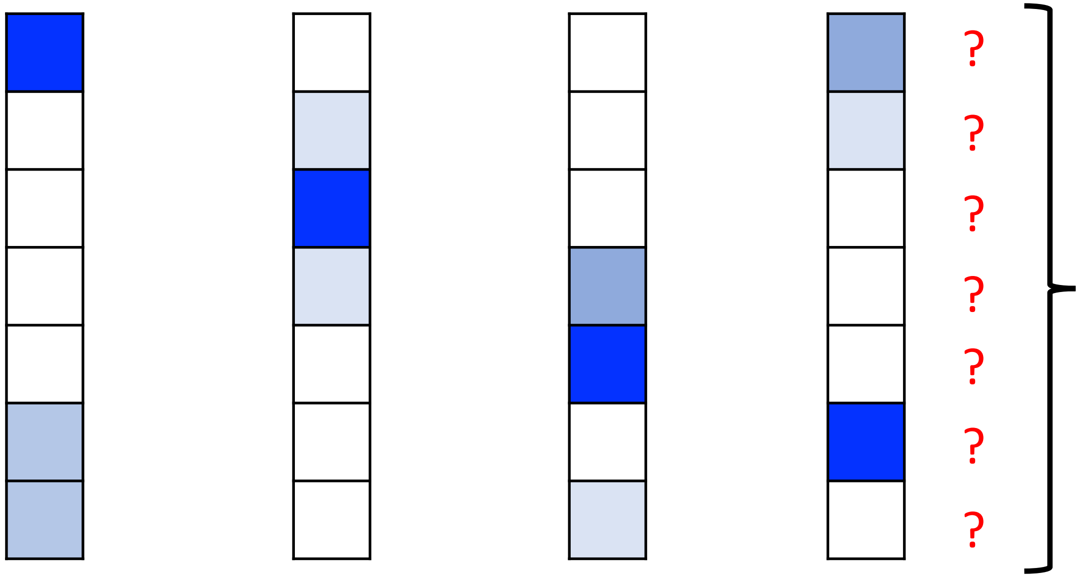
Approach: Learn Word Embedding Space

- 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 dimensionality for the shown example?



Approach: Learn Word Embedding Space

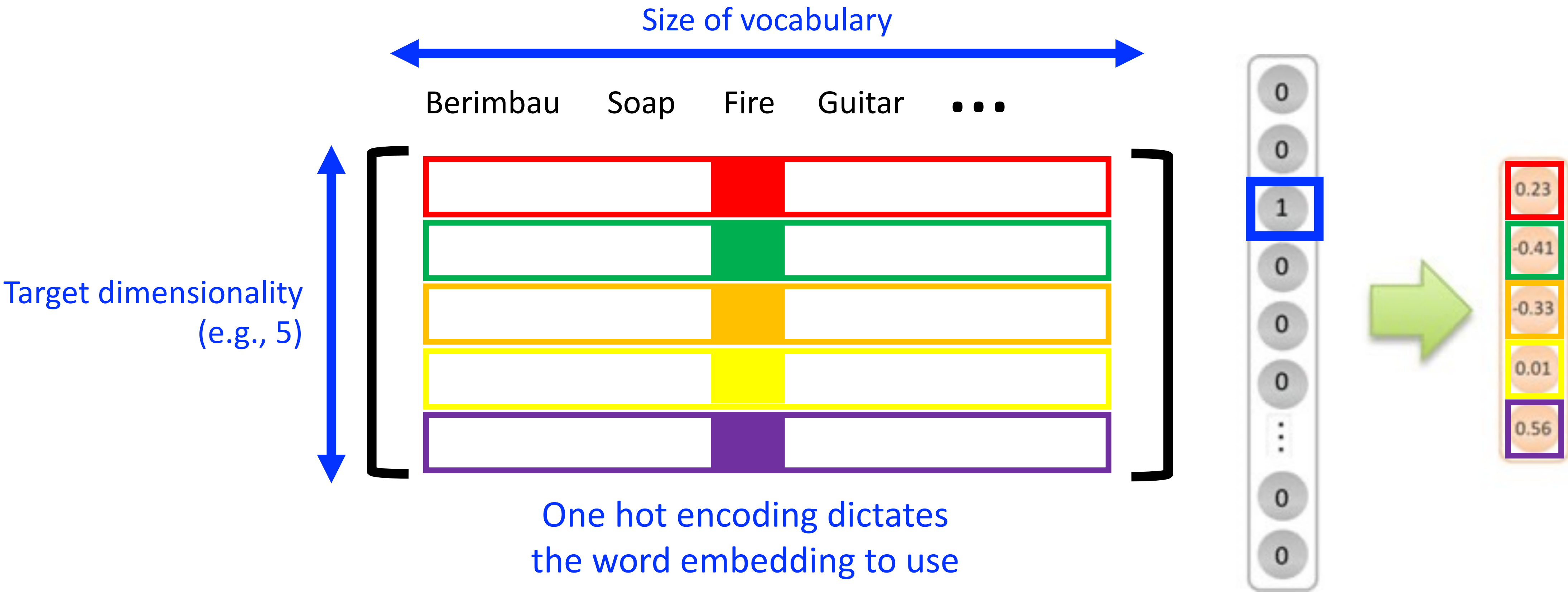
- An embedding space represents a finite number of words, defined in training
- A word embedding is represented as a vector indicating its context
- The dimensionality of all word embeddings in an embedding space match



In practice, the learned discriminating features are hard to interpret

Embedding Matrix

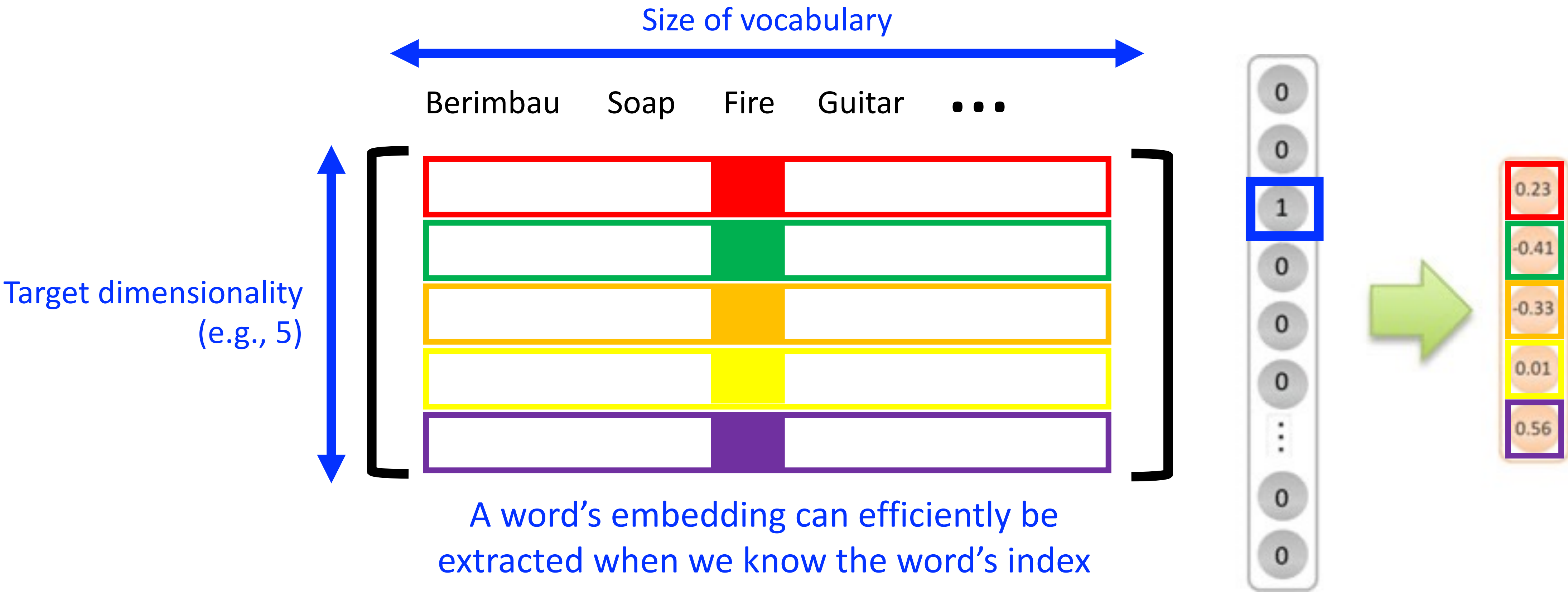
- The embedding matrix converts an input word into a dense vector



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

Embedding Matrix

- It converts an input word into a dense vector



A word's embedding can efficiently be extracted when we know the word's index

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

Popular Word Embeddings

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

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

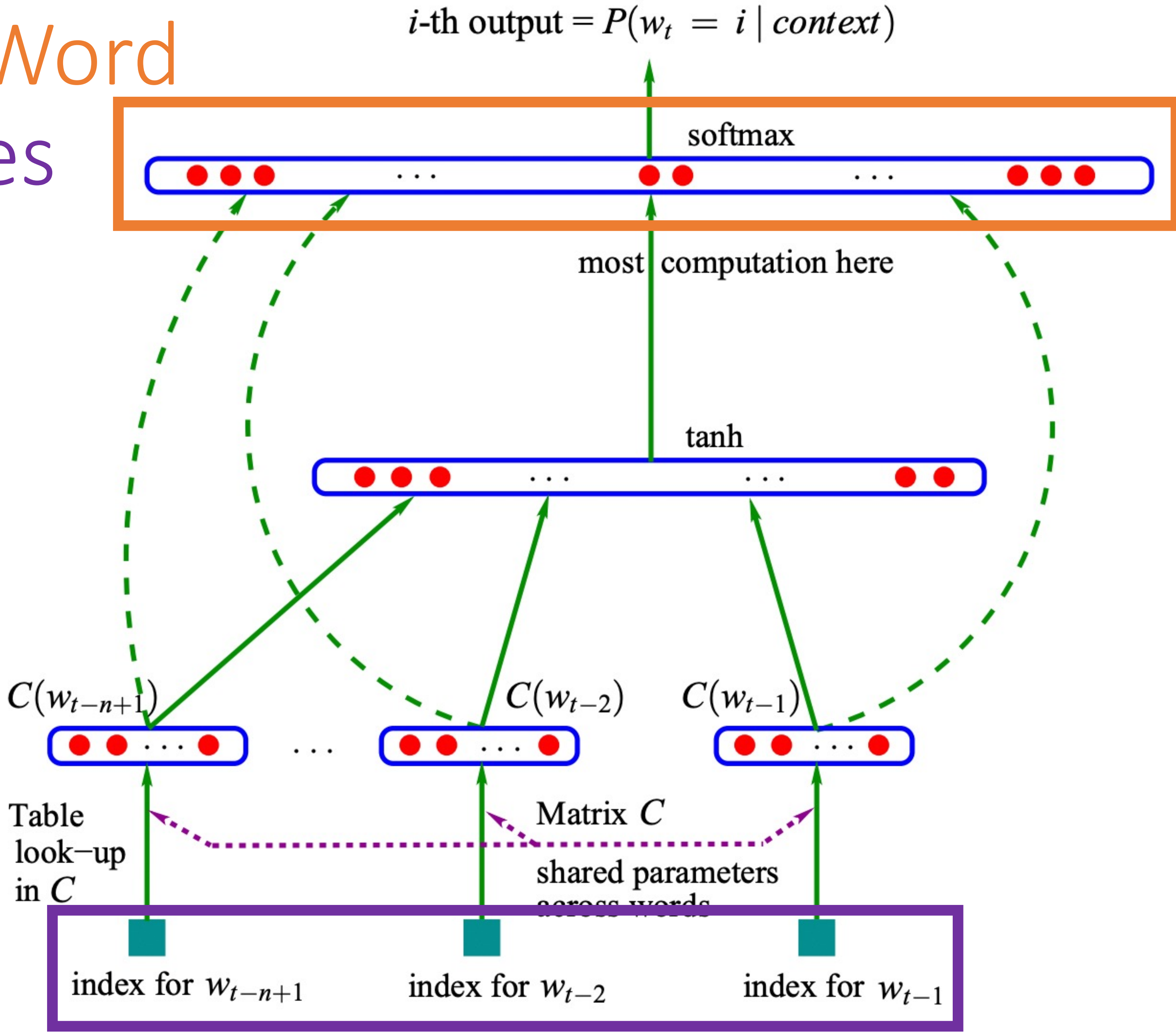
e.g.,

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?

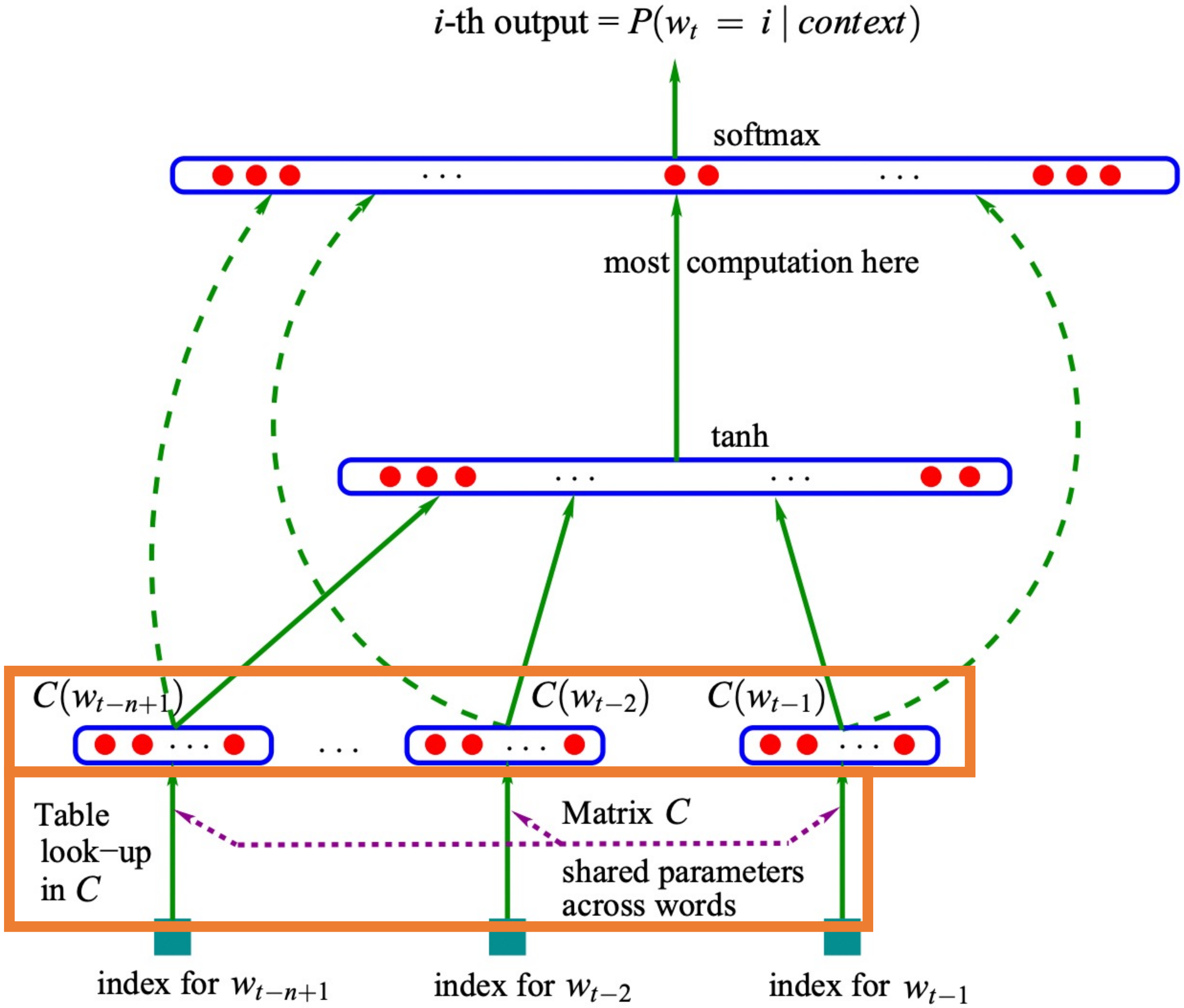


Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Architecture

Word embeddings:

Embedding matrix:



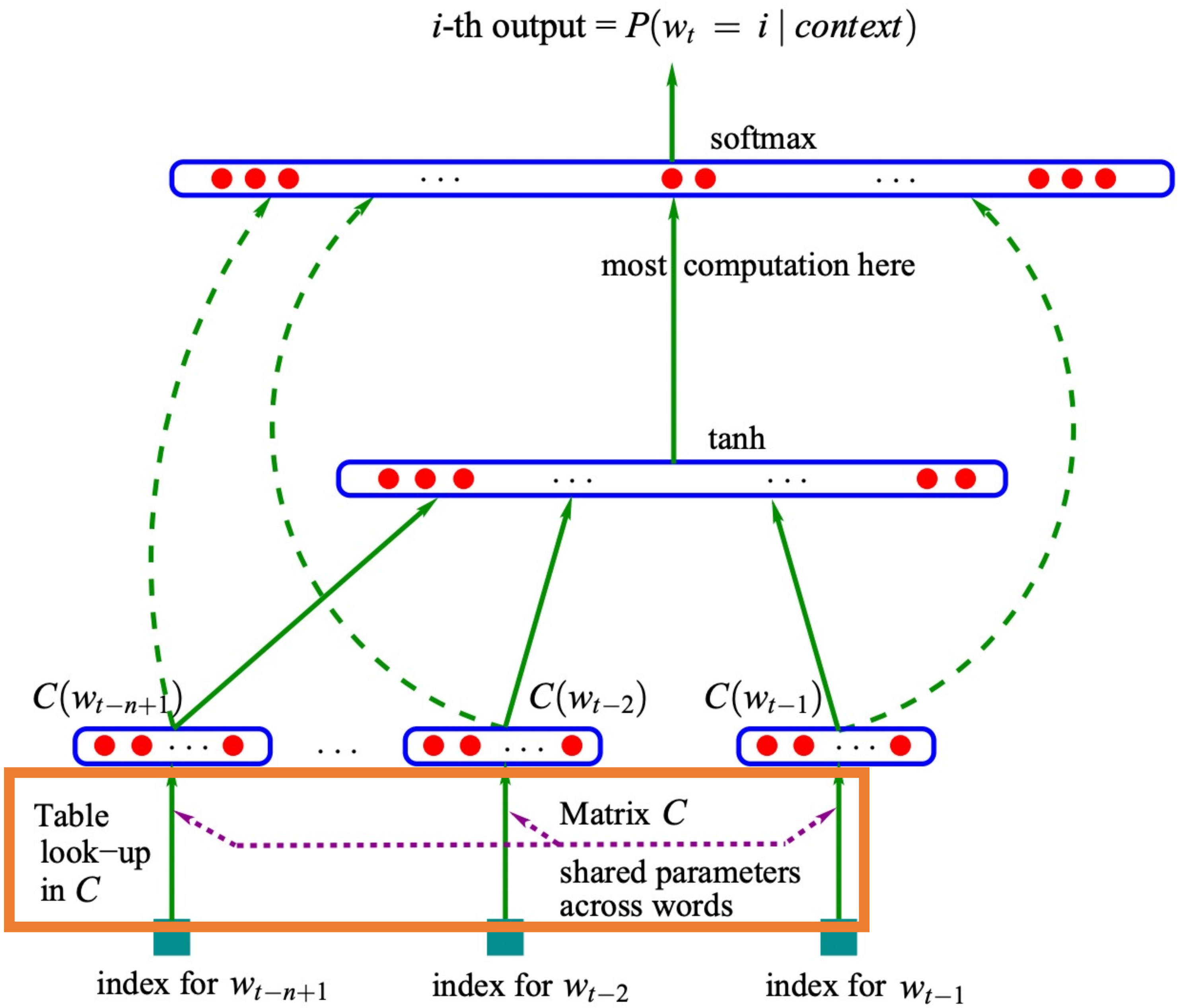
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

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)



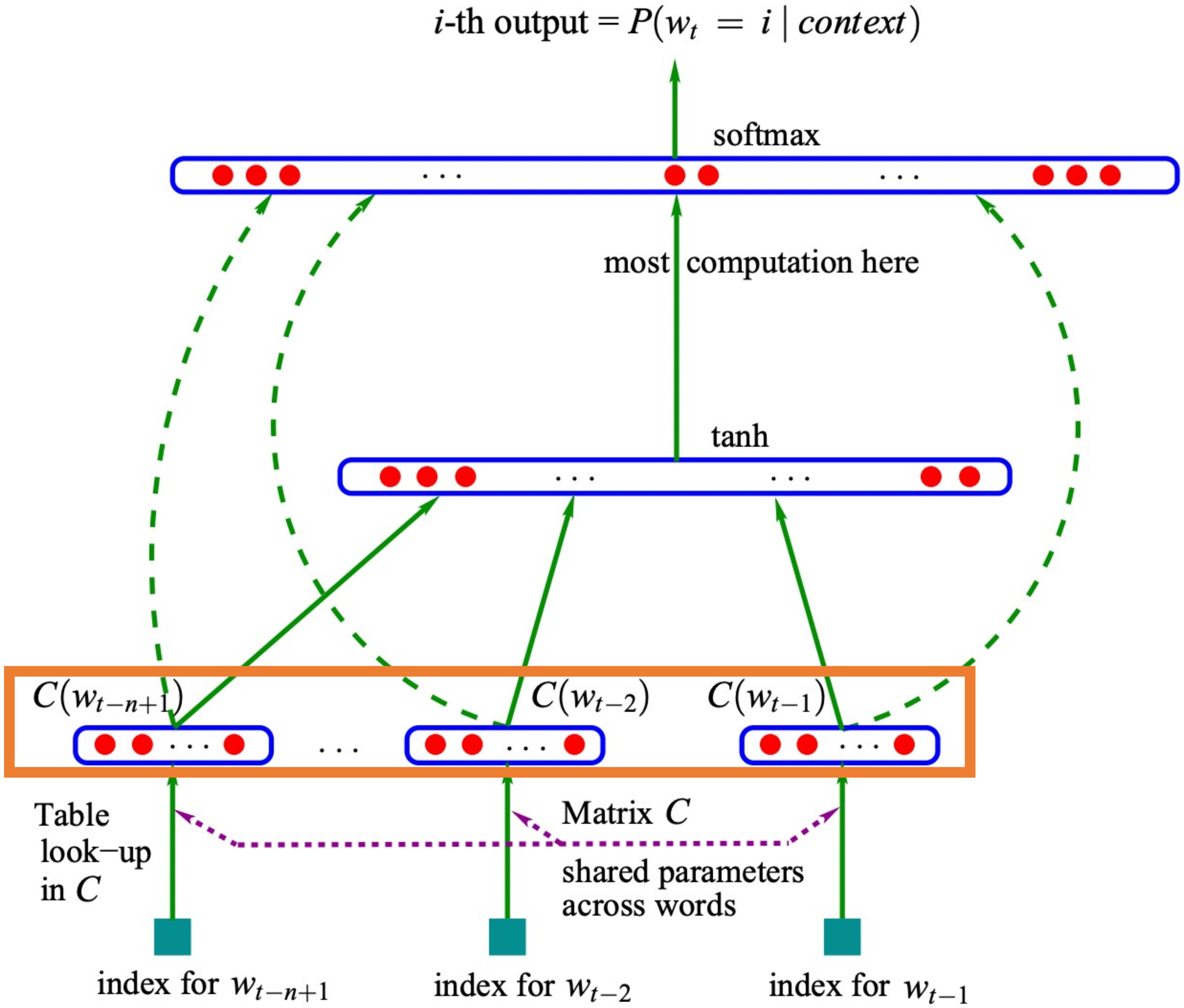
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

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?

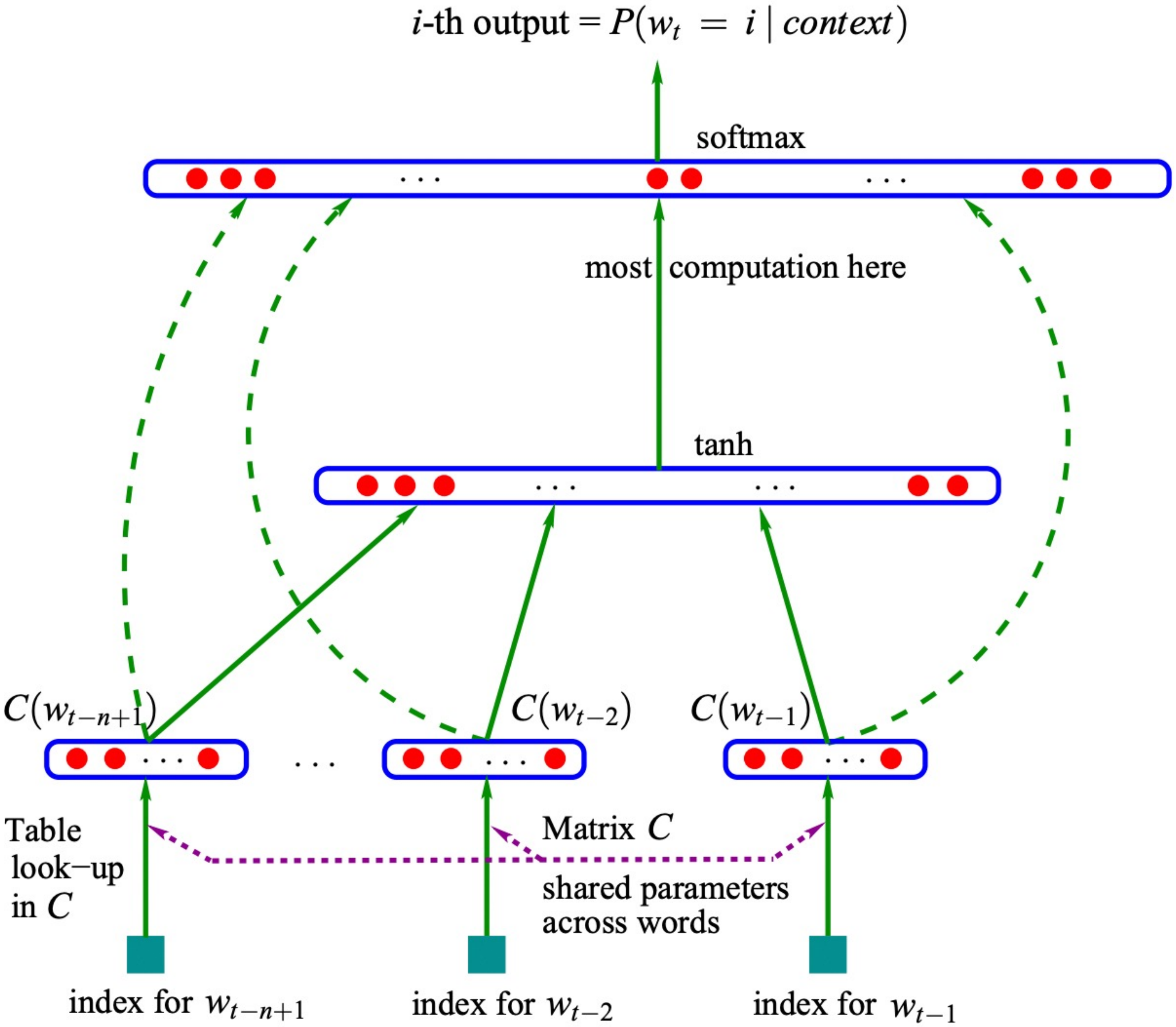
1 x 30



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

Architecture

Projection layer followed by a hidden layer with non-linearity



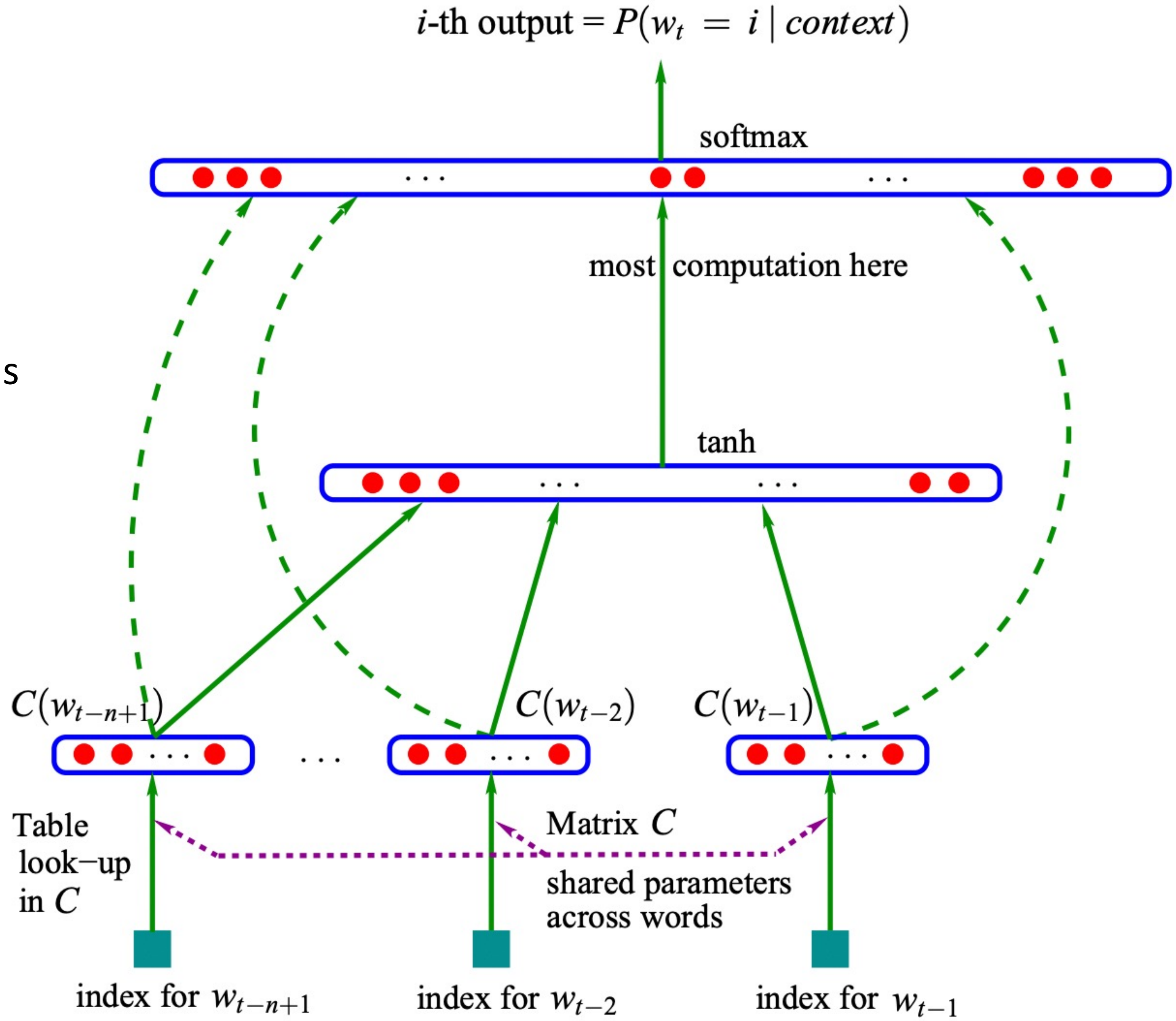
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

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



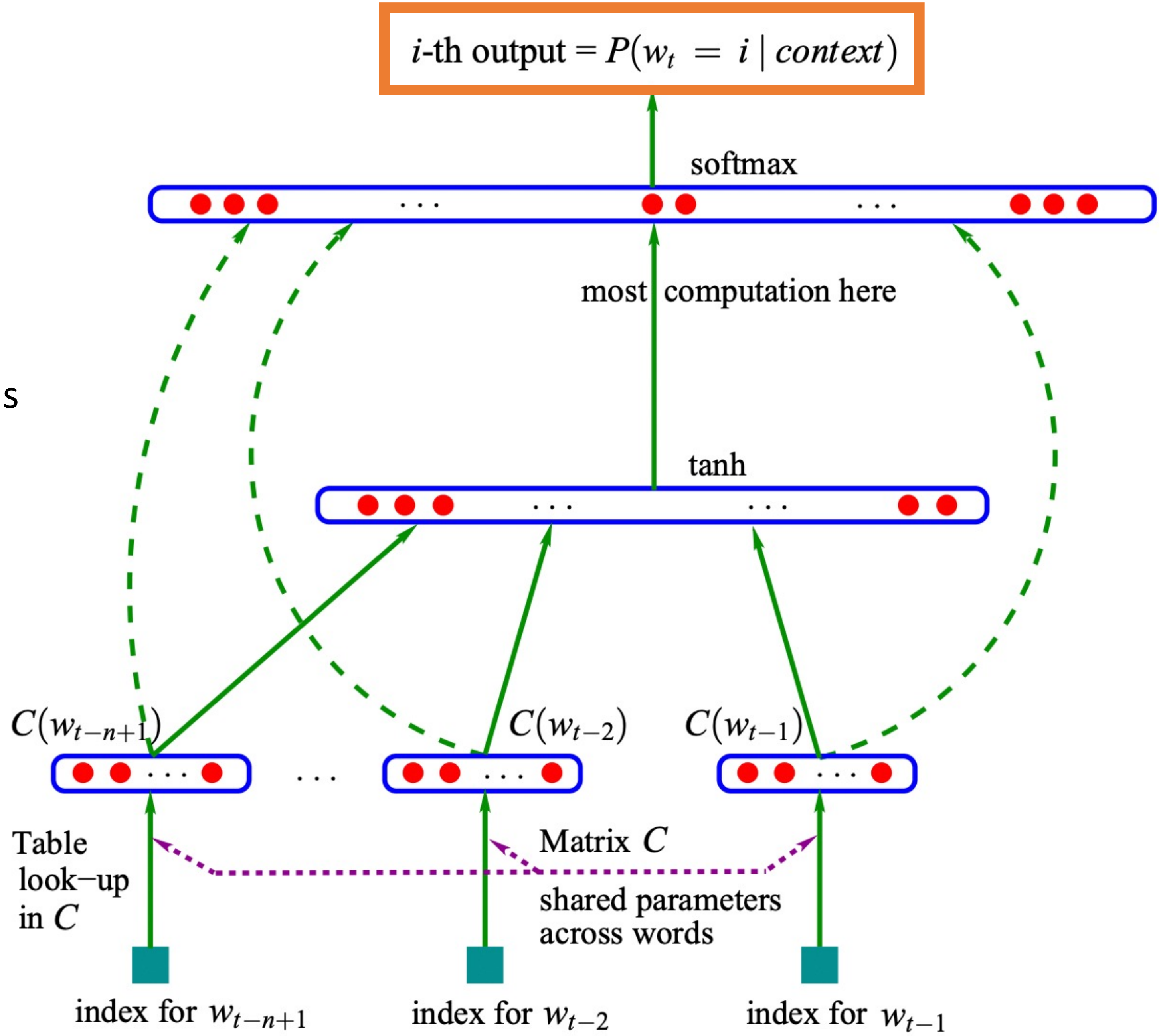
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

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



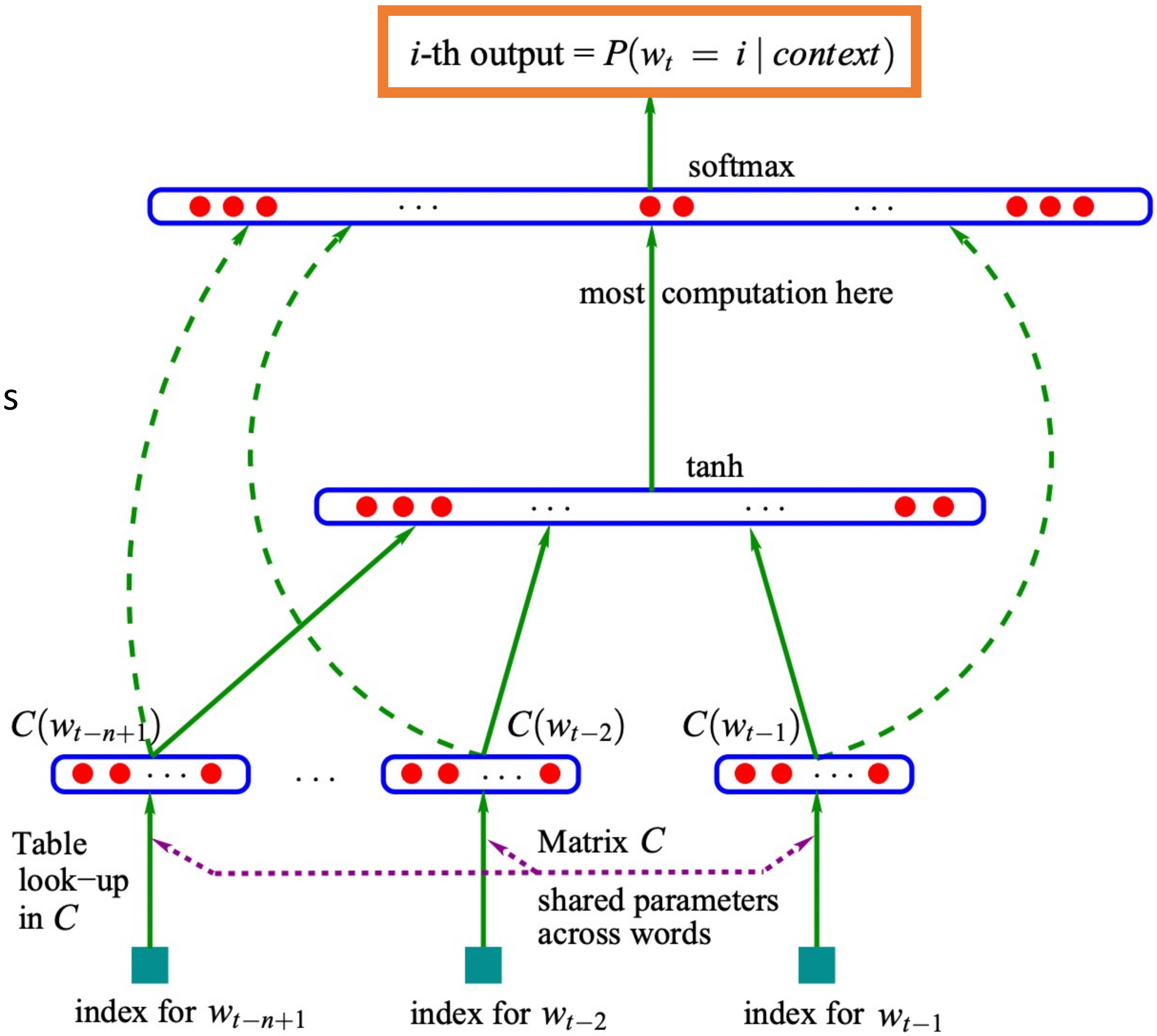
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

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



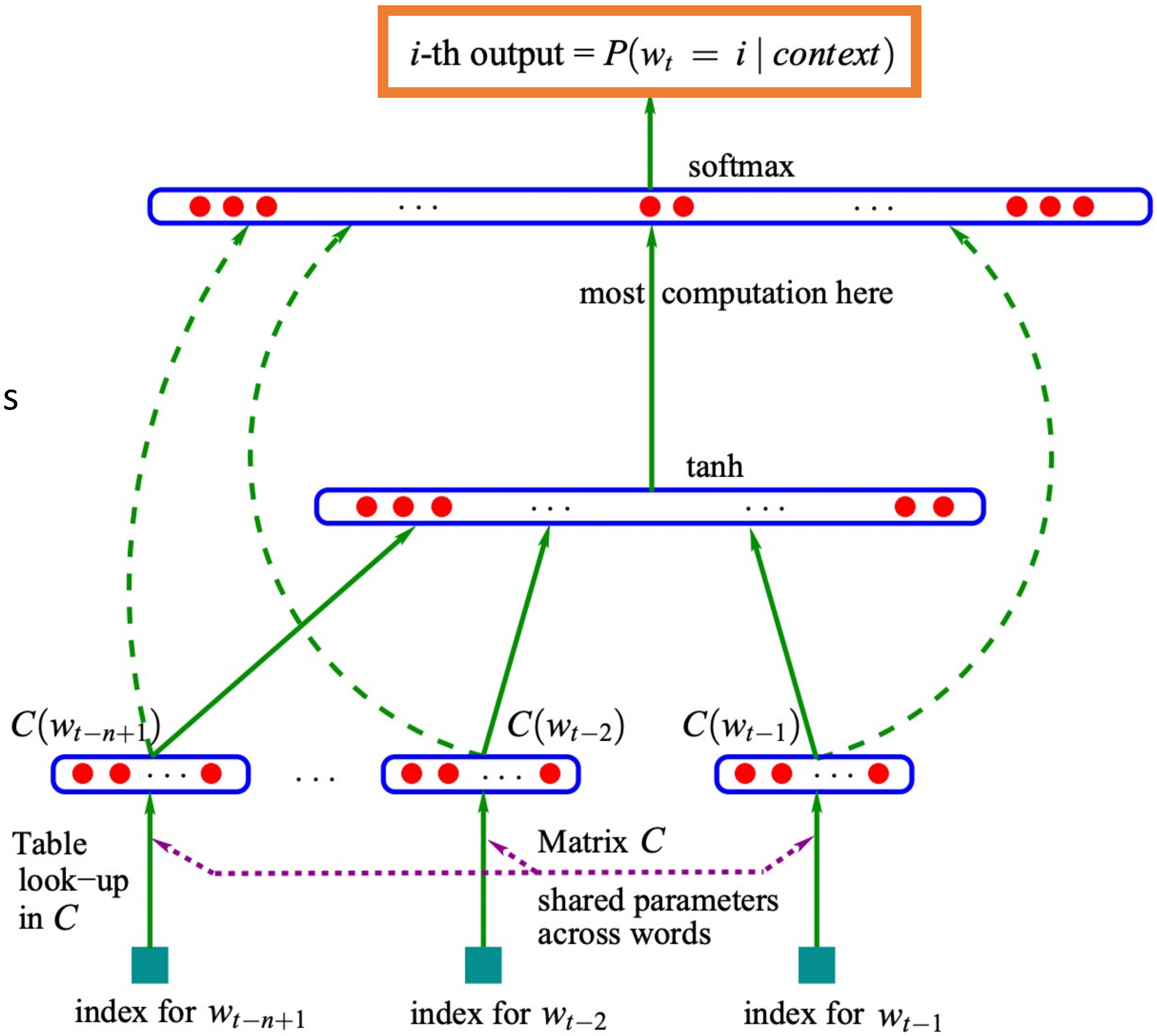
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

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



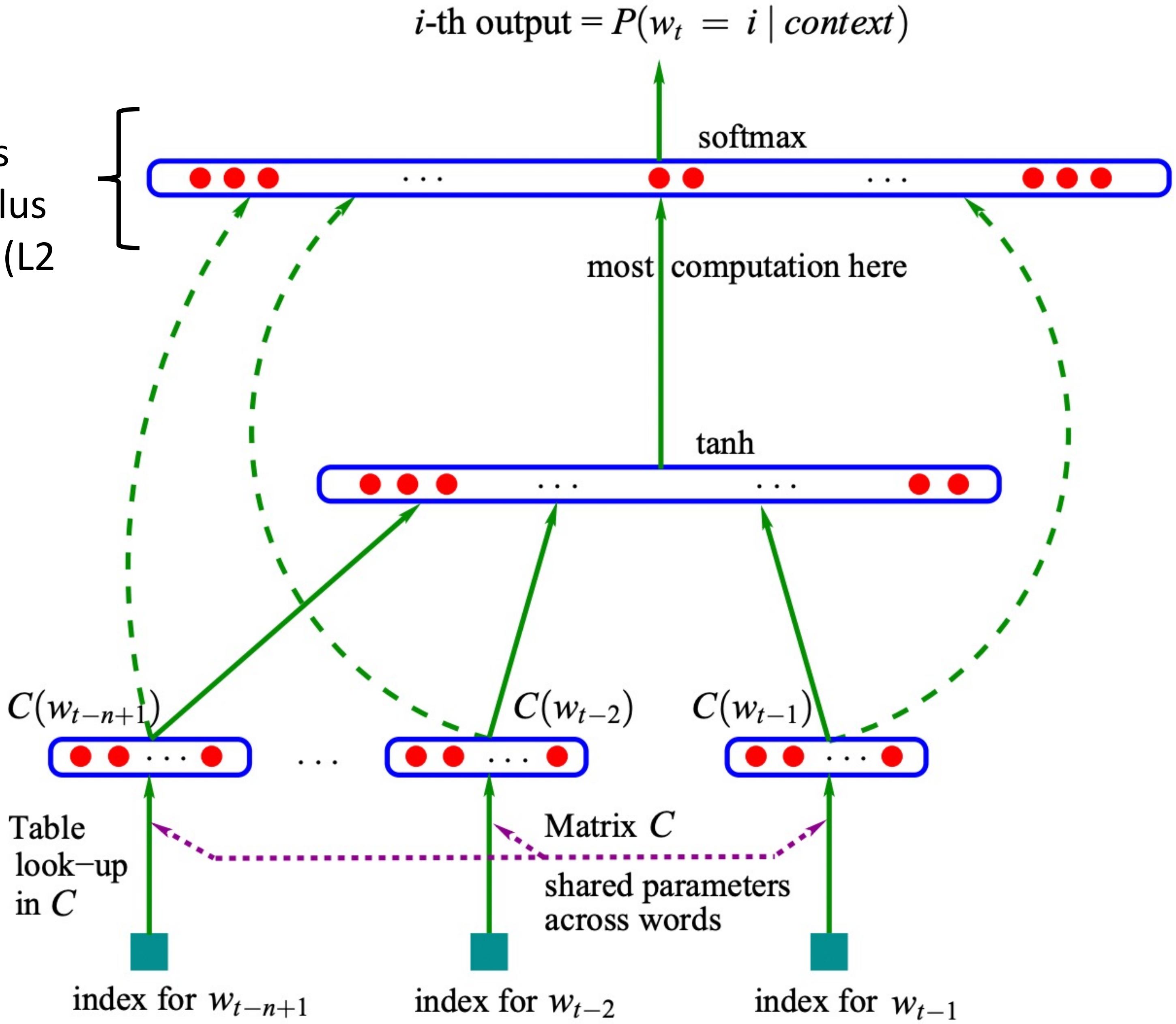
Bengio et al. A Neural Probabilistic Language Model. JMLR 2003.

Training

Cost function: minimize cross entropy loss plus regularization (L2 weight decay)

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 Are Learned that Support Predicting 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 ________

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 ___

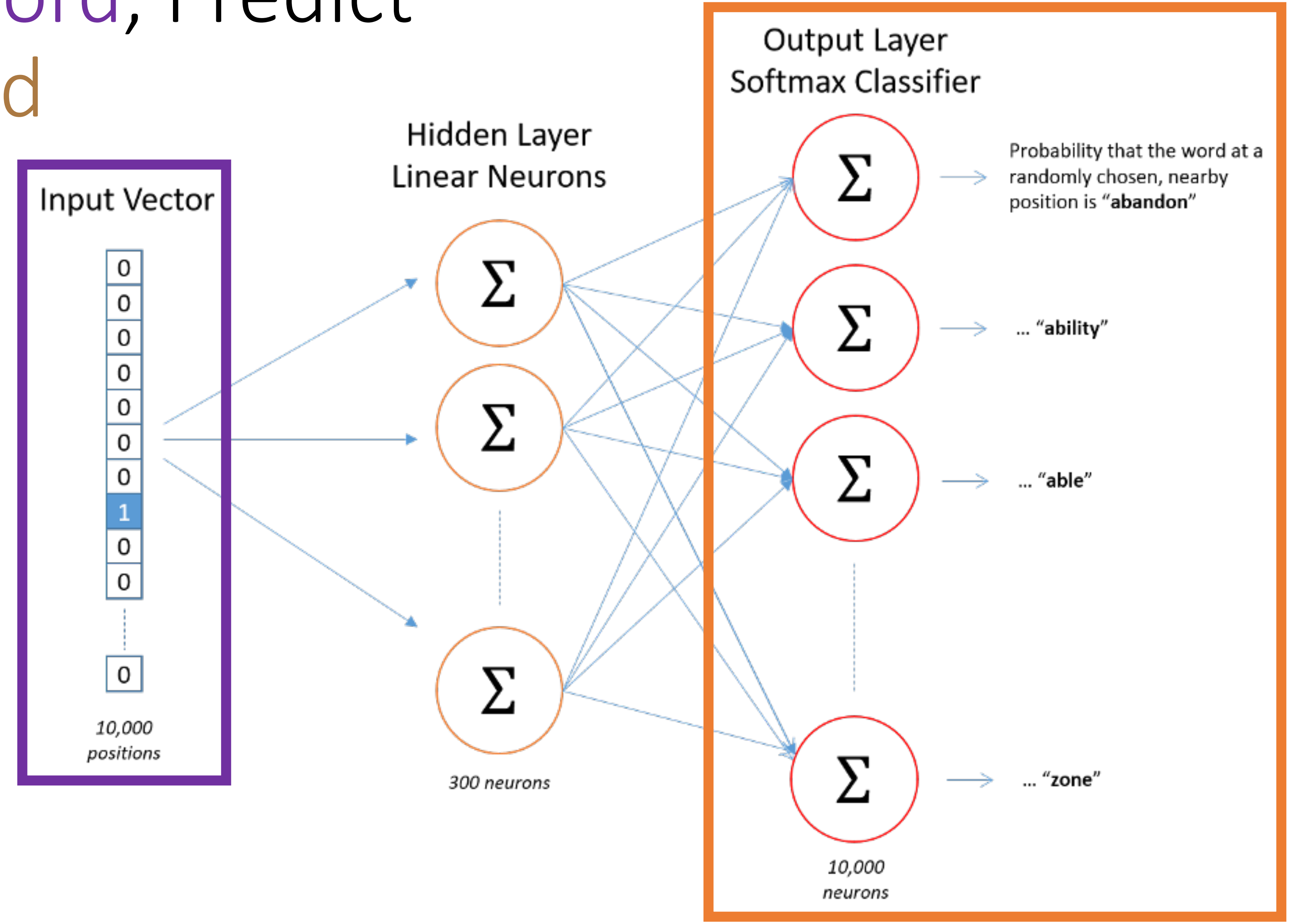
Mikolov et al. Efficient Estimation of Word Representations in Vector Space. arXiv 2013.

Task: Given Word, Predict a Nearby Word

e.g.,

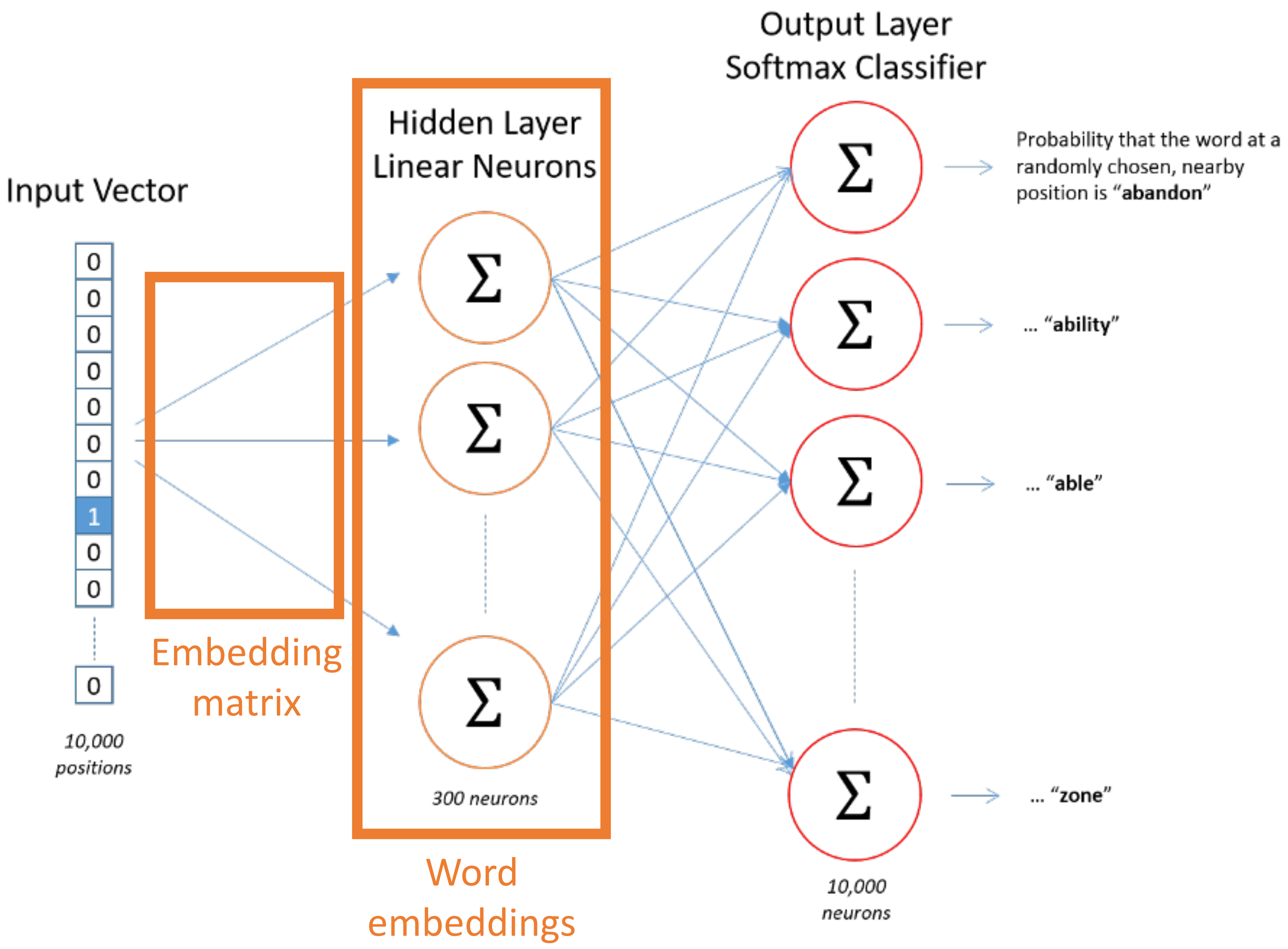
1. ___ ___ ___ ___ berimbau ___ ___ ___ ___
2. ___ berimbau ___

Task: Given Word, Predict a Nearby Word



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

Architecture



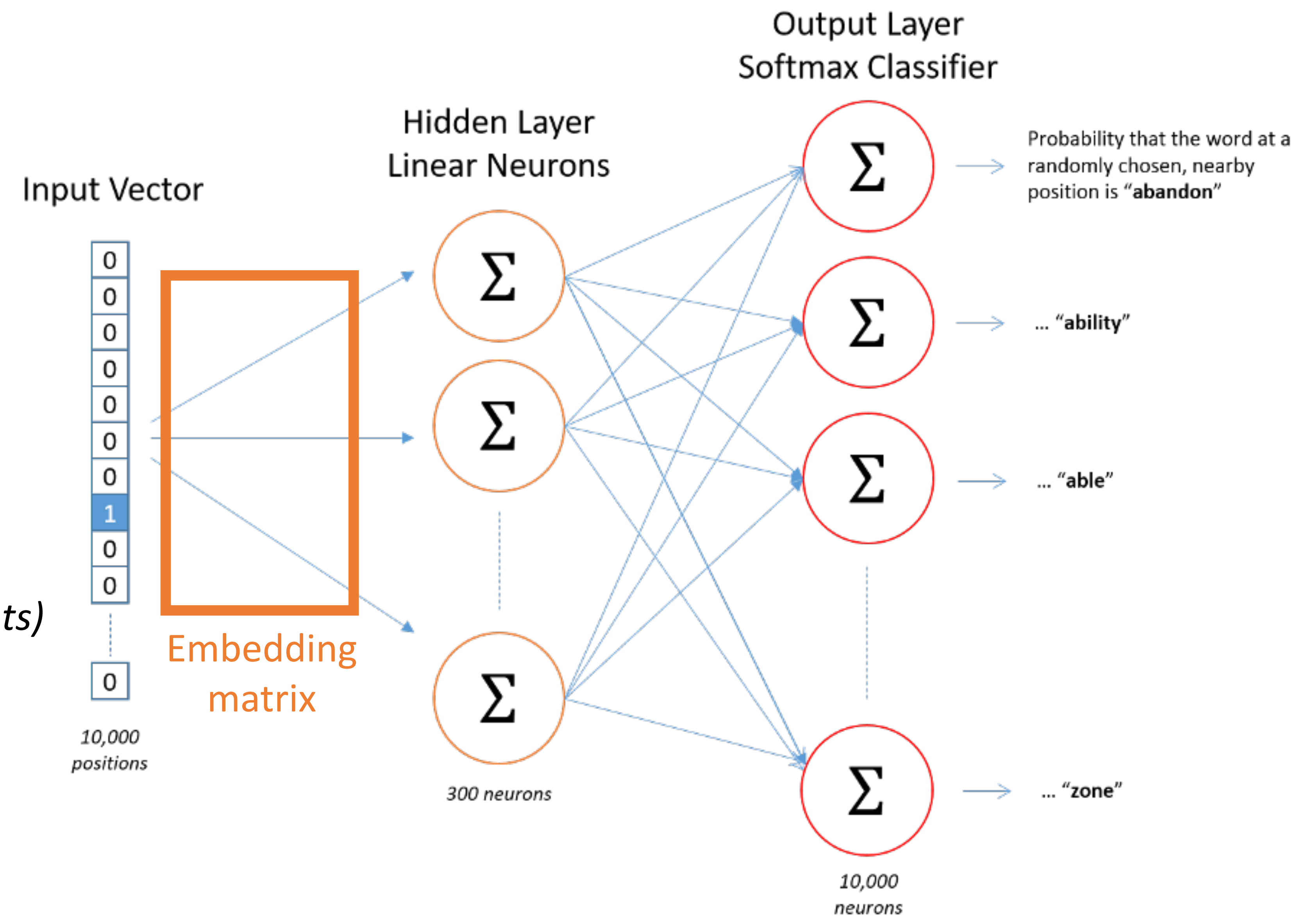
https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b

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)



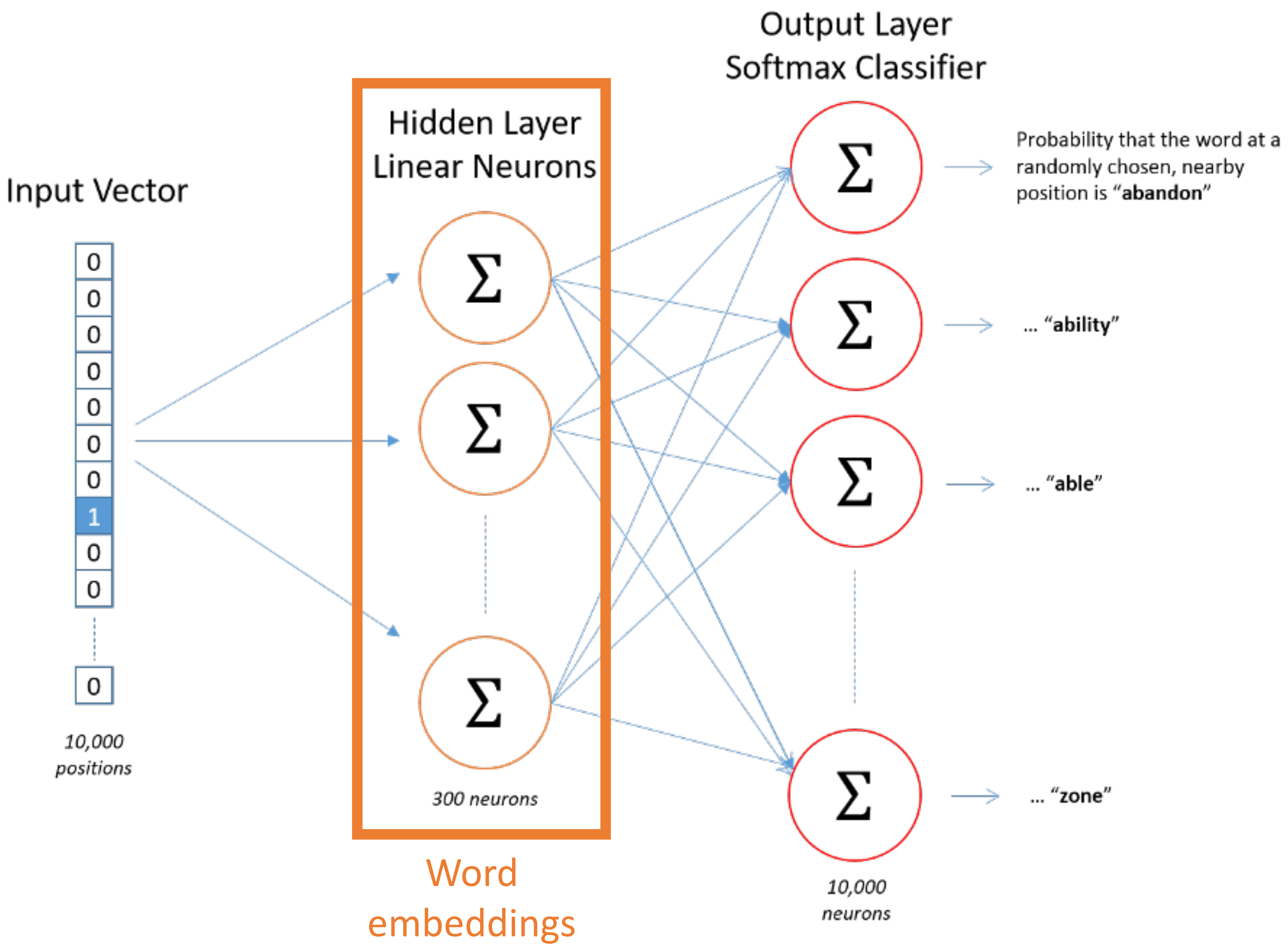
https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b

Architecture

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

What are the dimensions of each word embedding?

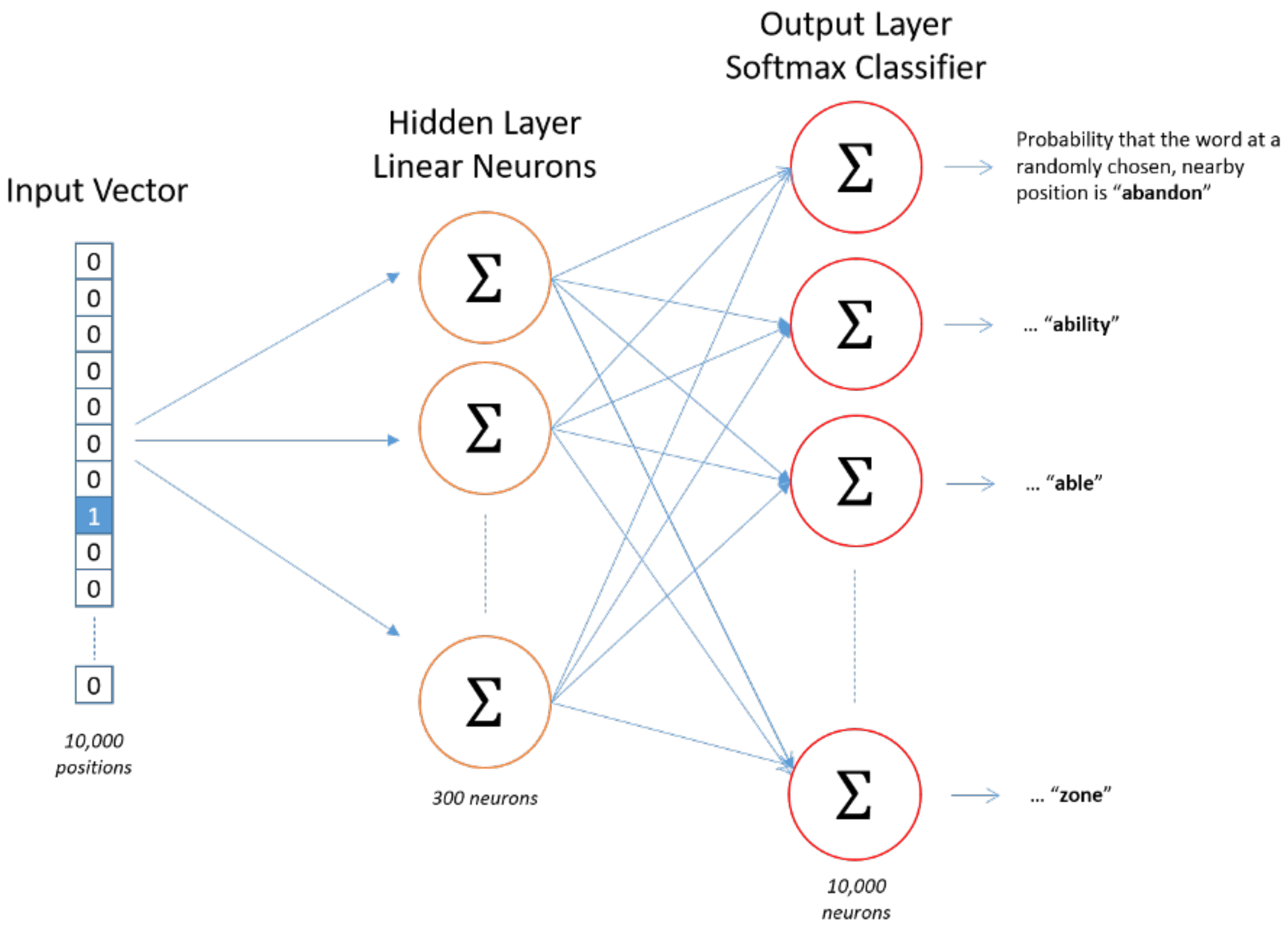
1 x 300



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

Architecture

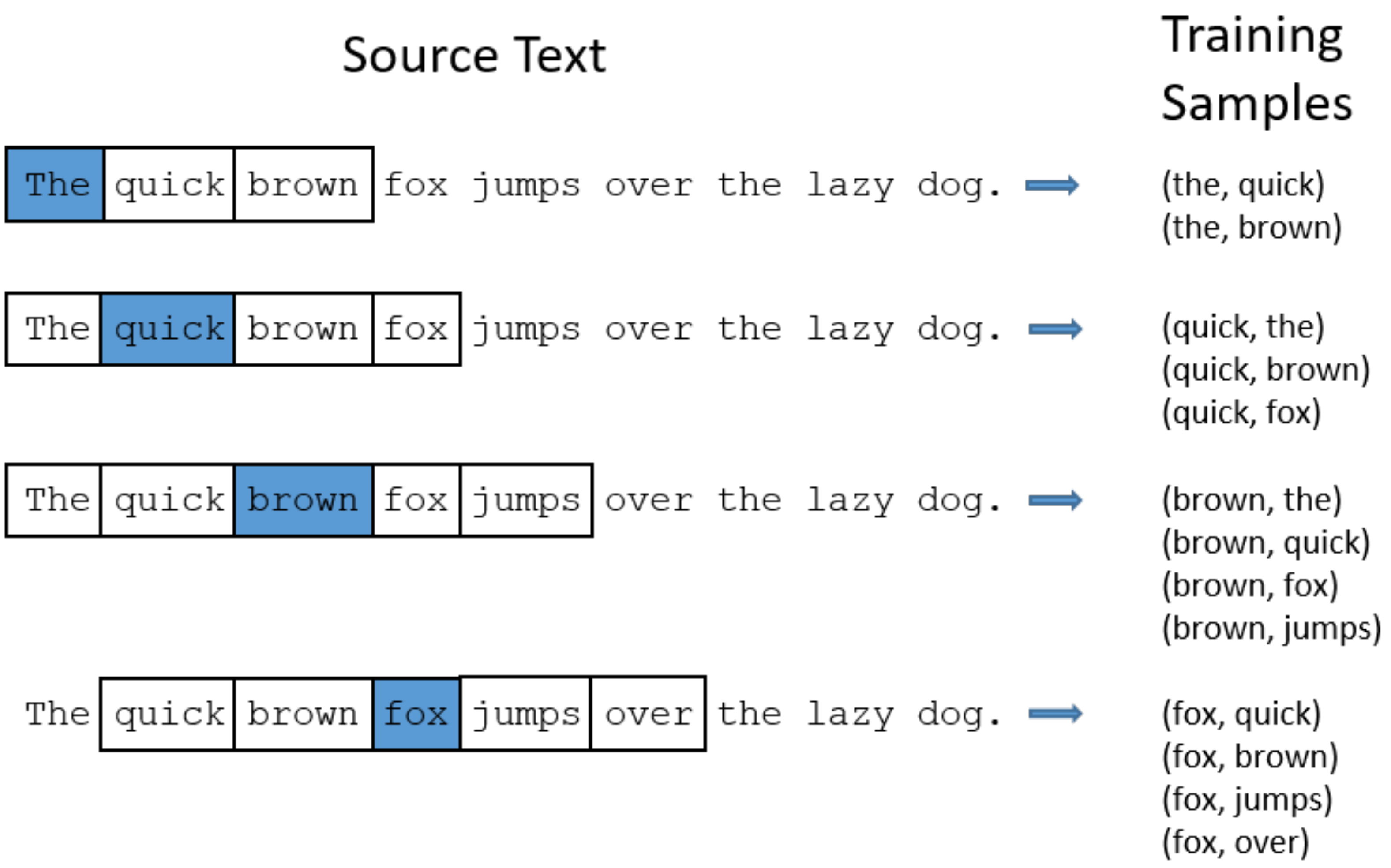
A shallower, simpler architecture than the Bengio approach (i.e., lacks a non-linear hidden layer)!



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

Training

Sliding window run on input data to sample neighbors of each target word (e.g., using window size of 2)

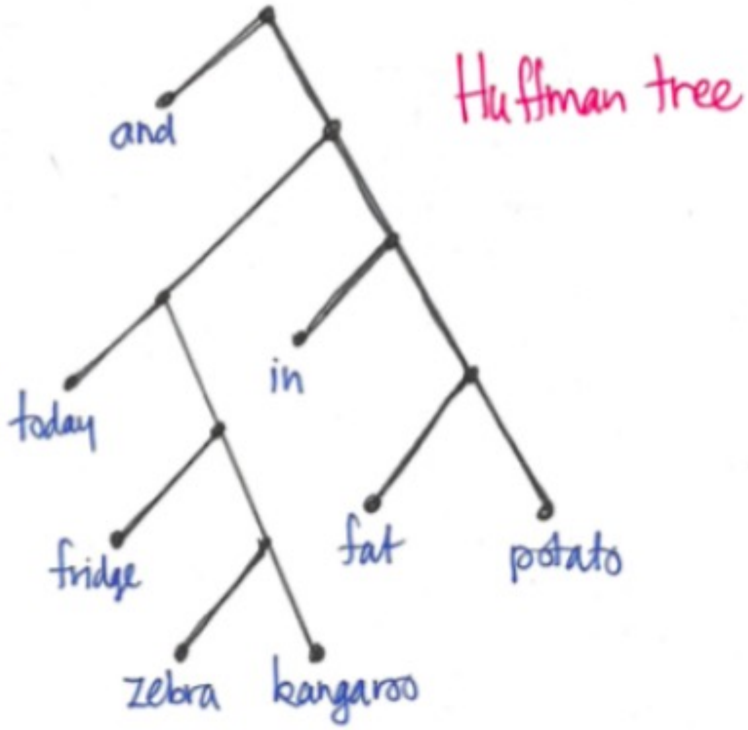


https://towardsdatascience.com/word2vec-skip-gram-model-part-1-intuition-78614e4d6e0b

Extra Tricks: More Efficient Representations

1. Change output layer to hierarchical softmax

word	count
fat	3
fridge	2
zebra	1
potato	3
and	14
in	7
today	4
kangaroo	2



2. Reformulate problem to perform negative sampling

Binary classification: predict for a given word if another word is nearby

- Positive examples: observed target and neighboring words
- Negative examples: randomly sampled other words

https://www.cs.princeton.edu/courses/archive/spring20/cos598C/lectures/lec2-word-embeddings.pdf

Mikolov et al. Distributed Representations of Words and Phrases and their Compositionality. Neurips 2013.

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 Word 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 word vectors under different contexts; e.g.,
  - Elmo for embeddings from language models (Peters et al., arXiv 2018)

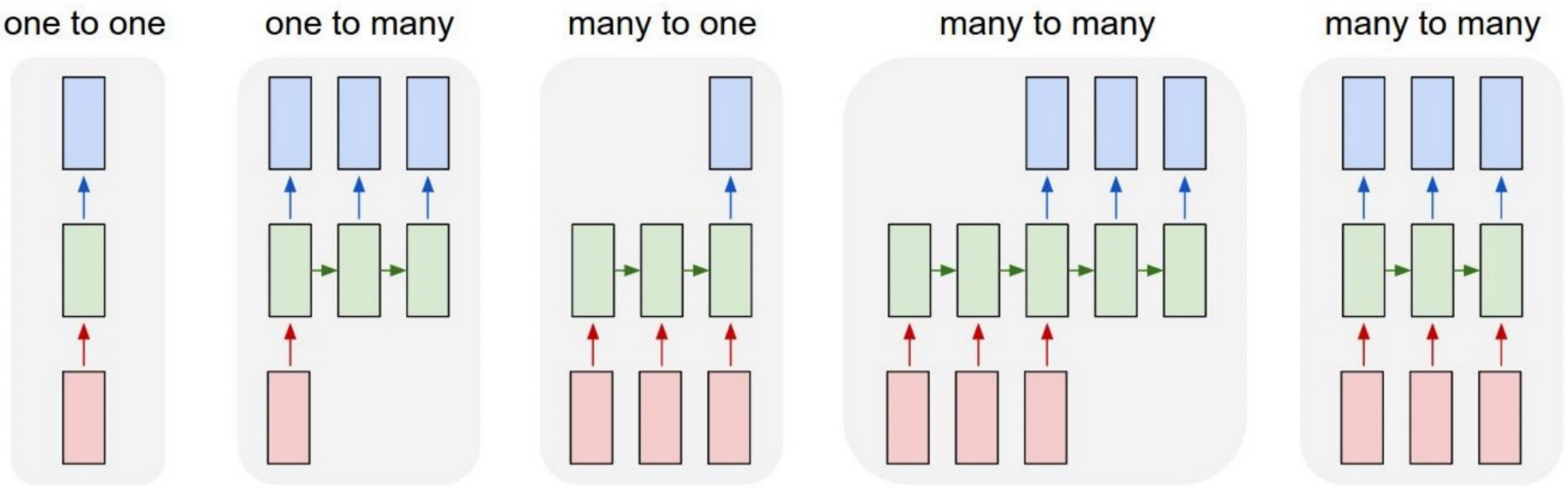
- Support multiple languages; e.g.,
  - Fast-text (Bojanowski et al., 2016)

Popular Word Embeddings

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

Recap of Big Picture

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



  - Implementation detail: may need to learn extra tokens such as “UNK” and “EOS” to represent out of vocabulary words and signify end of the string respectively
- Also, can fine-tune word embedding matrices for different applications

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

Word Embedding Limitations/Challenges

- Distinguish antonyms from synonyms
  - Antonyms are learned near each other in the embedding space since they are commonly used in similar contexts: “I hate math” vs “I love math” or “Take a right turn” vs “Take a left turn”

- 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 learned near each other in the embedding space since they are commonly used in similar contexts: “I hate math” vs “I love math” or “Take a right turn” vs “Take a left turn”

- 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 learned near each other in the embedding space since they are commonly used in similar contexts: “I hate math” vs “I love math” or “Take a right turn” vs “Take a left turn”

- Gender bias

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

Bolukbasi et al. Neurips 2016.

Today’s Topics

- Introduction to natural language processing
- Text representation
- Neural word embeddings
- Programming tutorial

Today’s Topics

- Introduction to natural language processing
- Text representation
- Neural word embeddings
- Programming tutorial


The End