Applied Machine Learning for Business Analytics

Lecture 8: Frontiers in NLP

Lecturer: Zhao Rui

Logistics

- All codes from lecture 6 can be run over colab:
  - Check this git repo: https://github.com/rz0718/BT5153_2022
- The topics for the last four lectures have been finalized!!
- Appreciate if you keeps video on!

Fri 03/11	Frontiers in NLP	LINK	N.A.
Fri 03/18	Model Evaluation in Machine Learning	TBU	N.A.
Fri 03/25	Causal Inference for Decision Making	TBU	Kaggle Competition Due
Fri 04/01	Model Deployment in Machine Learning	TBU	Kaggle Report Due
Fri 04/08	Why ML Projects Fail in Business	TBU	N.A.

Agenda

1. Representation Learning in NLP
2. Word Embeddings
3. Neural Networks for NLP
4. Introduction to BERT

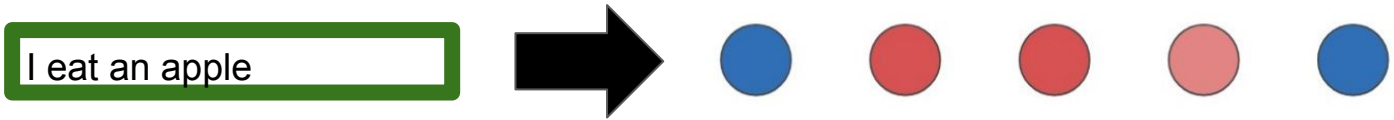
1. Representation Learning

Representation Learning

- We need to develop systems that read and understand text the way a person does, by forming a representation of the text, and other context information that humans create to understand a piece of text.

Representation Learning

- We need to develop systems that read and understand text the way a person does, by forming a representation of the text, and other context information that humans create to understand a piece of text.



Text

Numerical Representation

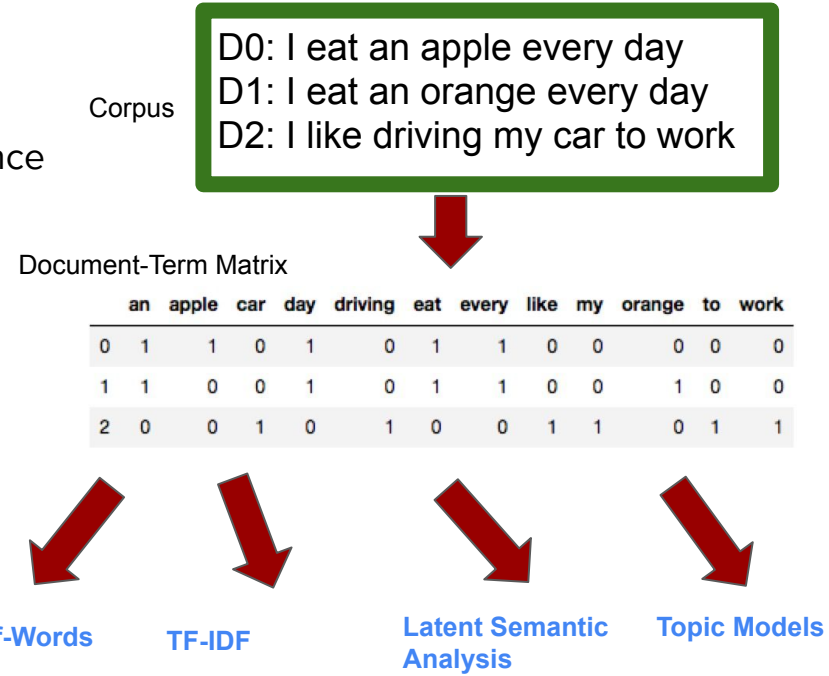
The learned representation should capture high-level semantic and syntactic information.

History of NLP

- Now, neural nlp models are able to achieve state-of-arts results in all tasks.
- Before neural nlp:
  - Symbolic NLP: rule-based system (derived from linguistic)
  - Statistical NLP: data-driven and use statistical methods

Statistical NLP

- Starting from Document-Term Matrix
  - It contains the co-occurrence information
  - Bag-of-Words: n-gram as features
  - TF-IDF: frequency of words to measure importance
  - Matrix Decomposition:
    - SVD->Latent Semantic Analysis
    - Probabilistic model-> Topic Model



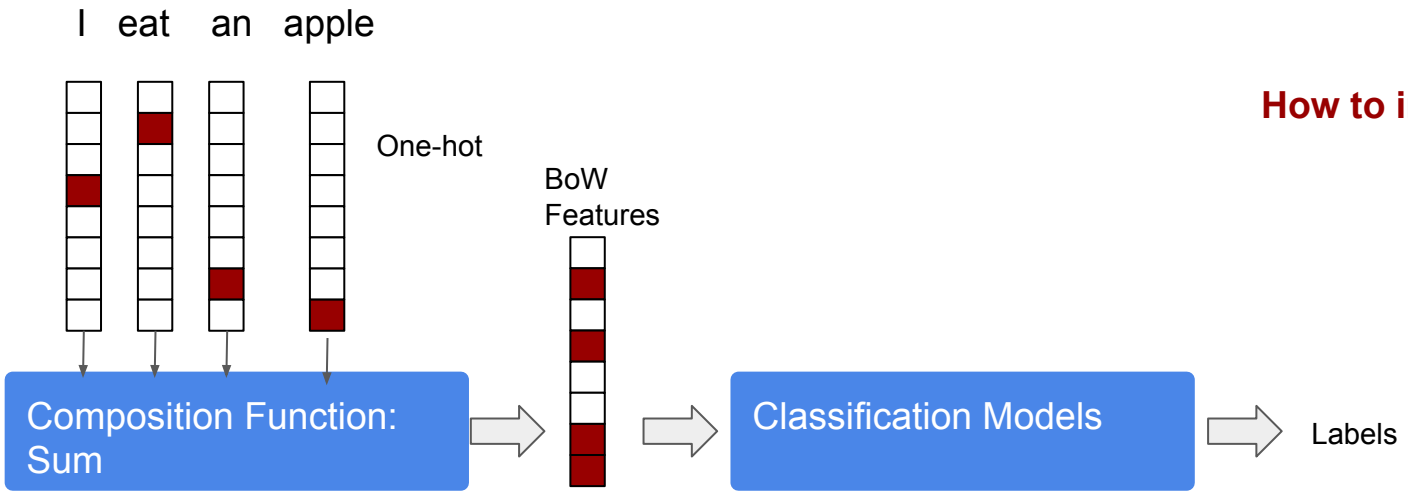
Limitations of Document-Term Matrix

- Too strong assumption: all words are independent of each other
  - | *orange - peach* | < | *orange - car* |
- Can not capture the order information in the sequence
- High dimensionality due to large size of vocabulary

	an	apple	car	day	driving	eat	every	like	my	orange	to	work
0	1	1	0	1	0	1	1	0	0	0	0	0
1	1	0	0	1	0	1	1	0	0	1	0	0
2	0	0	1	0	1	0	0	1	1	0	1	1

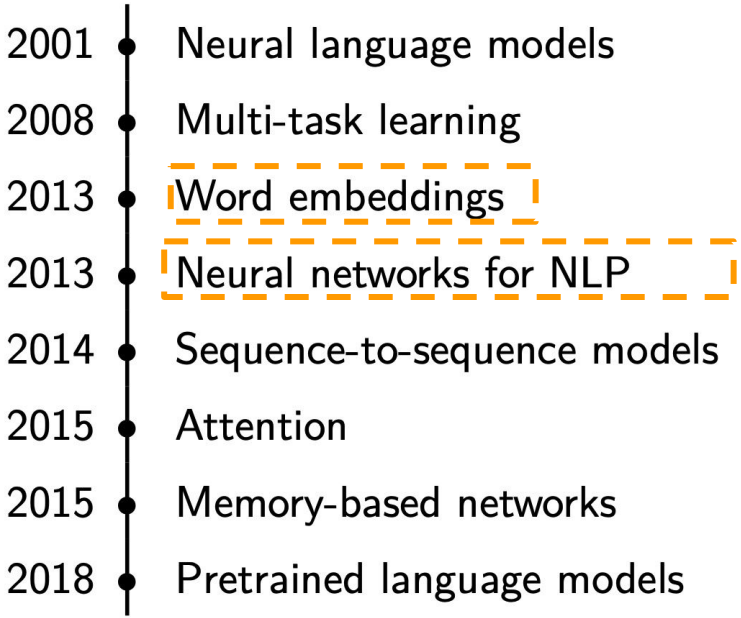
A New Perspective on BoW

- Each word in vocab is represented in one-hot embedding
- Sum one-hot vectors of the words in a sentence
- The final vector is the representation for the given sentence and then fed into a classifier.



How to improve?

Neural NLP



https://www.kamperh.com/slides/ruder+kamper_indaba2018_talk.pdf

2. Word Embeddings

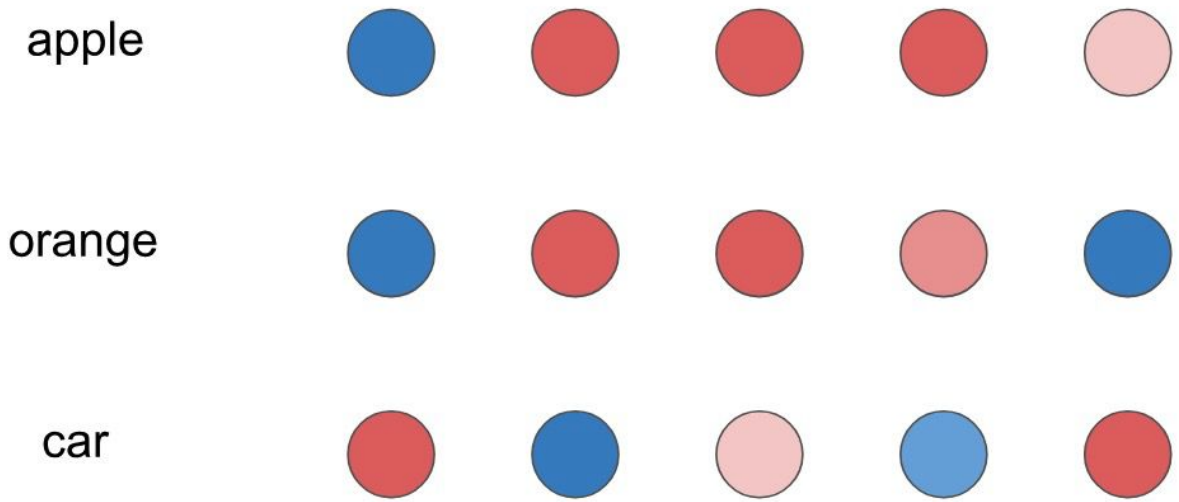
Word Representation

- How to represent words in a vector space

apple	[0 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 0 0 ... 0 0 0 0 0 0]
orange	[0 0 0 0 0 0 0 0 0 0 0 0 0 **1** 0 0 0 0 ... 0 0 0 0 0 0]
car	[0 0 0 0 0 0 0 **1** 0 0 0 0 0 0 0 0 0 ... 0 0 0 0 0 0]

Distributed Representation

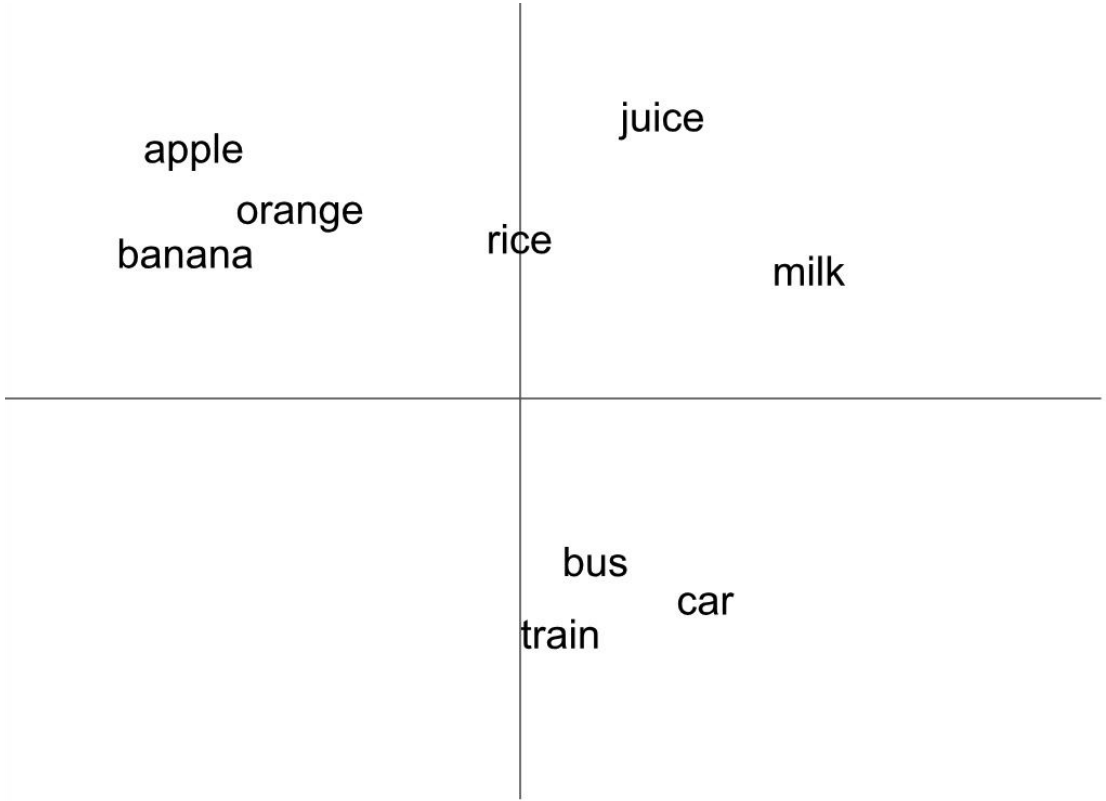
- Words should be encoded into a low-dimensional and dense vector



Word Vectors

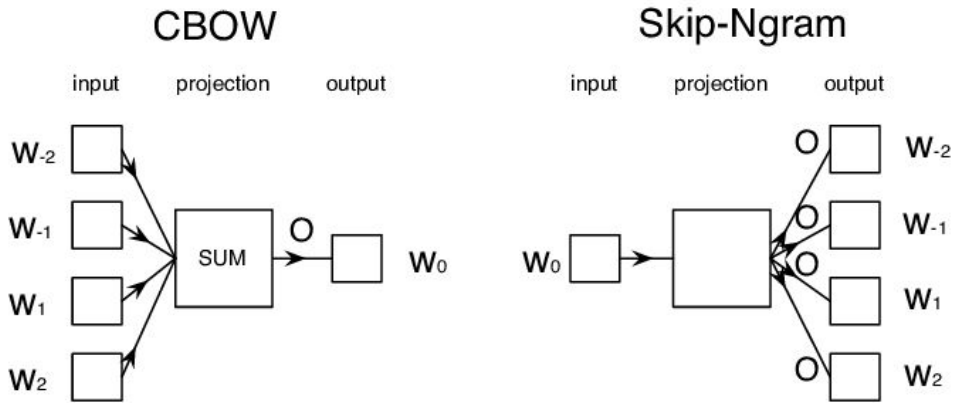
Project word vectors in a two-dimensional space. And visualize them!

Similar words are close to each other.

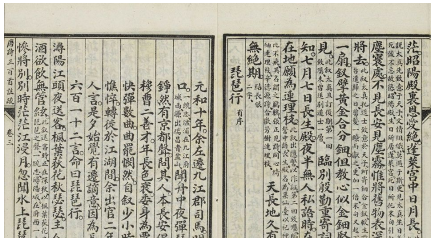


Word2Vec

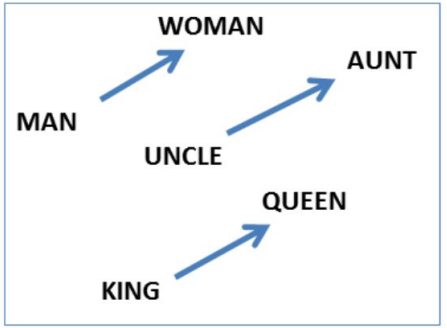
- A method of computing vector representation of words developed by Google.
- Open-source version of Word2Vec hosted by Google (in C)
- Train a simple neural network with a single hidden layer to perform word prediction tasks.
- Two structures proposed Continuous Bag of Words (CBoW) vs Skip-Gram



Word2Vec as BlackBox



input, output



Corpus **Word2Vec Tool** **Word Embeddings**

A Good Visualization for Word2 Vec

https://ronxin.github.io/wevi/

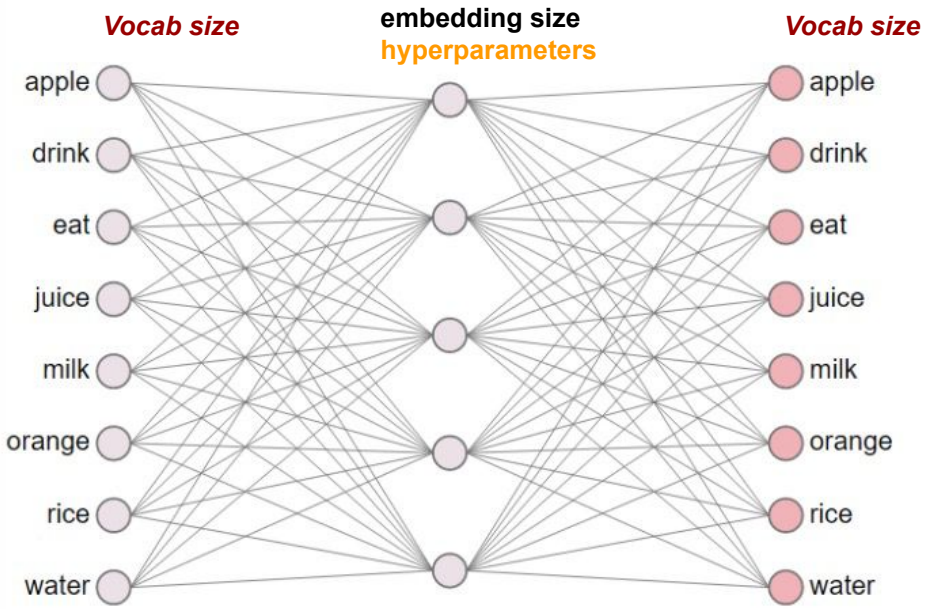
Target

- Given a training corpus, we prepare a list of N (input_word, output_word).
- Objective Function: Maximize probability of all the output words given the corresponding input words.

$$\mathbf{J}(\theta) = \prod_{i=1}^{N} p(w^i_{output} | w^i_{input}, \theta)$$

Neural network parameters that will be optimized

Model Architecture



Structure Highlights:

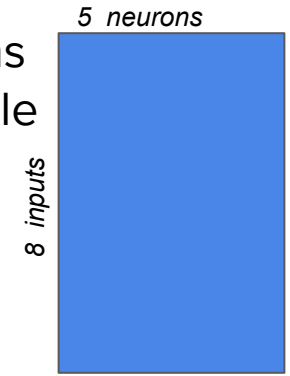
- input layer
  - one-hot vector
- hidden layer
  - linear (identity)
- output layer
  - softmax

From Xin Rong 2016

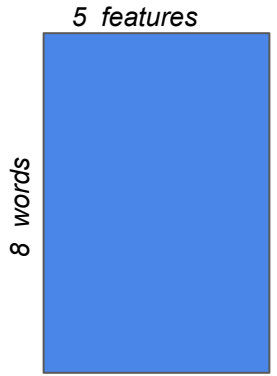
Hidden Layer

- **Linear-activation** function here
- **5** neurons are the word vec. dimensions
- This layer is operating as a ‘lookup’ table
- Input word matrix denoted as **IVec**

Hidden Layer Weights Matrix



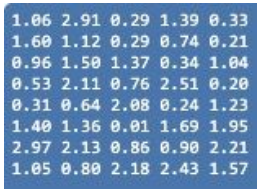
Word Vector Look Up Table



One-hot vector

[0,0,1,0,0,0,0, 0]

Index of eat



Word vector for “eat”

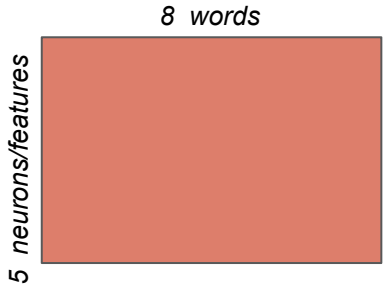
0.96, 1.5, 1.37, 0.34, 1.04

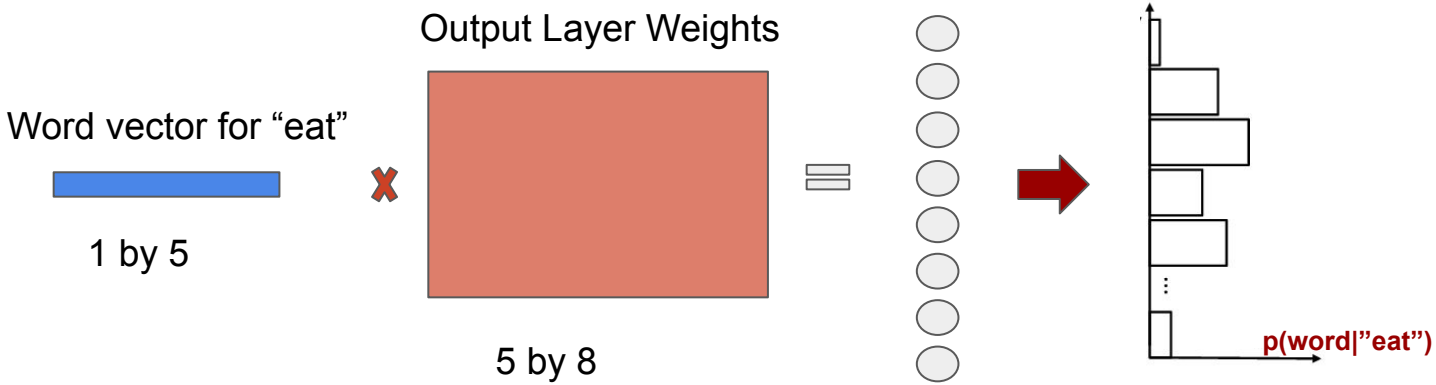
This is a **projection/look up** process: given the index of the word, we take the ith row in the word vector matrix out

Output Layer

- Softmax Classifier
- Output word matrix denoted as **OVec**

Output Layer Weights Matrix
A.K.A Output word vectors

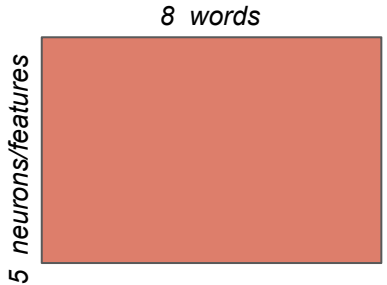




Output Layer

- Softmax Classifier
- Output word matrix denoted as **OVec**

Output Layer Weights Matrix
A.K.A Output word vectors



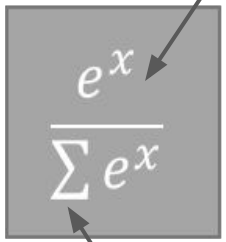
Output weights for “apple”

e^(IVec[eat]*OVec[apple])

Word vector for “eat”

5 features

×



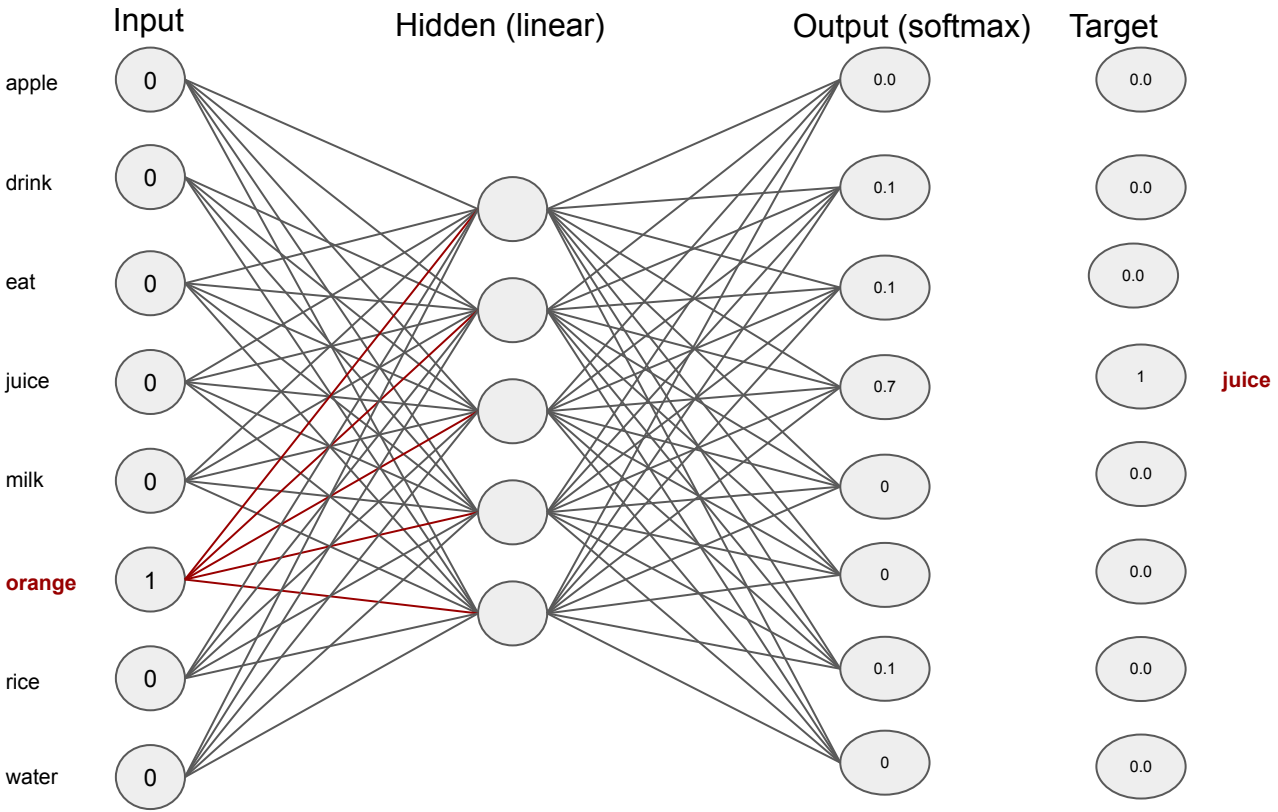
=

Probability that if you randomly pick a word nearby “eat”, that it is “apple”

p(apple|eat)

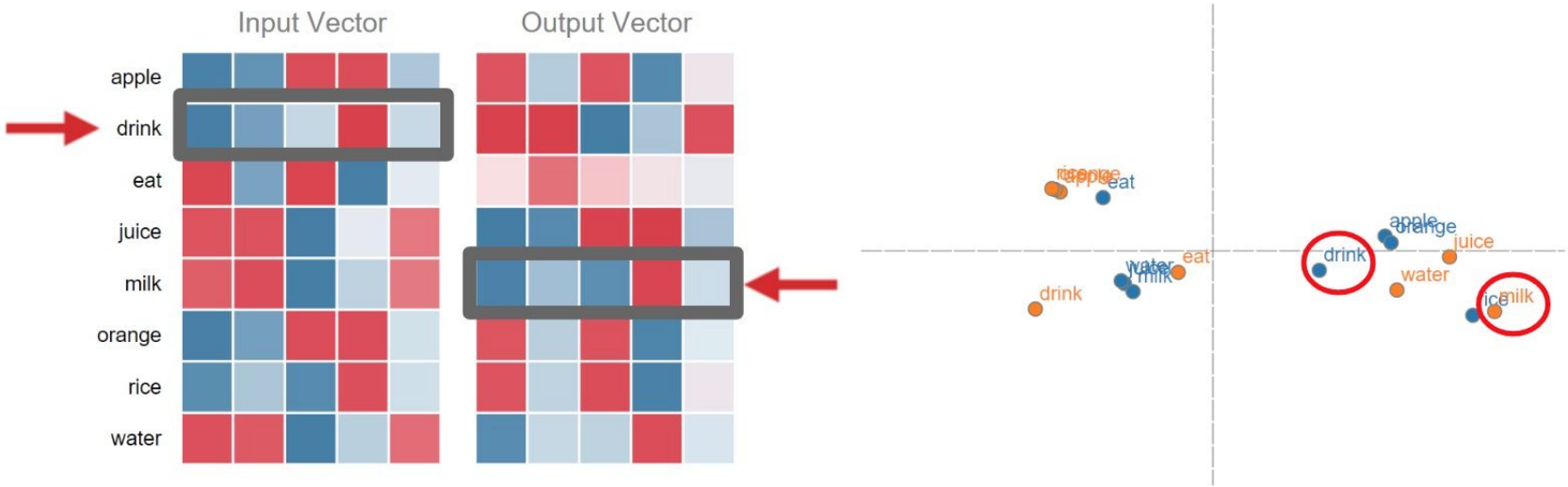
e^(IVec[eat]*OVec[apple]) + e^(IVec[eat]*OVec[juice]) + e^(IVec[eat]*OVec[drink])+e^(IVec[eat]*OVec[other vocab words)

Word2Vec



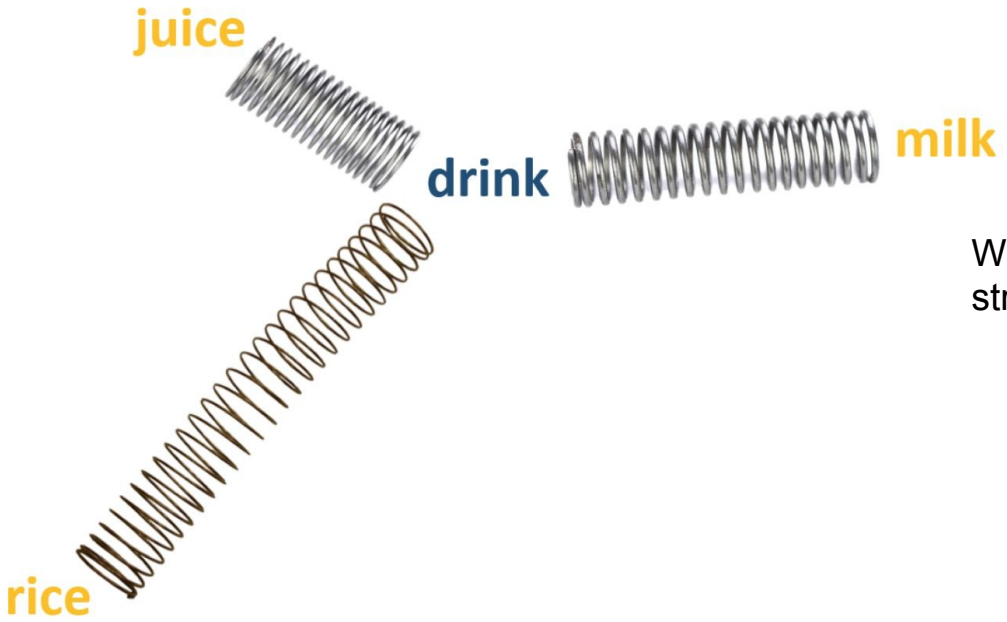
Then, we can compute the loss and call gradient descent to update model parameters.

Updating Word Vectors



From Xin Rong 2016

A force-directed graph



What decides the strength of the string?

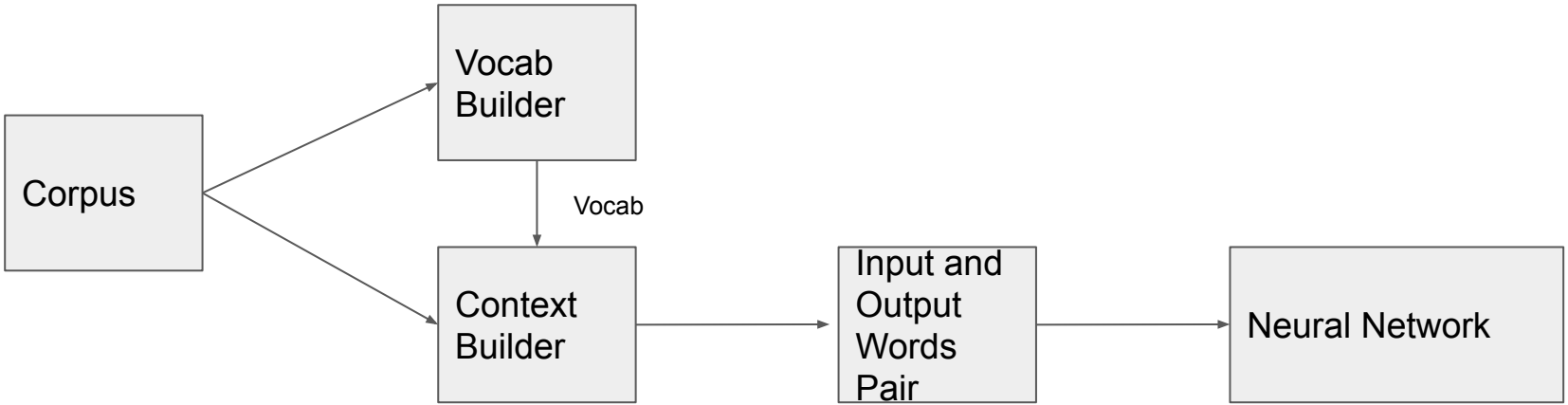
Idea behind Word2Vec

- Feature vector assigned to a word will be adjusted if it can not be used for accurate prediction of that word's context.
- Each word's context in the corpus is the teacher sending error signals back to modify the feature vector.
- It means that words with similar **context** will be assigned similar **vectors**!

"**You shall know a word by the company it keeps**" - by Firth (1957)

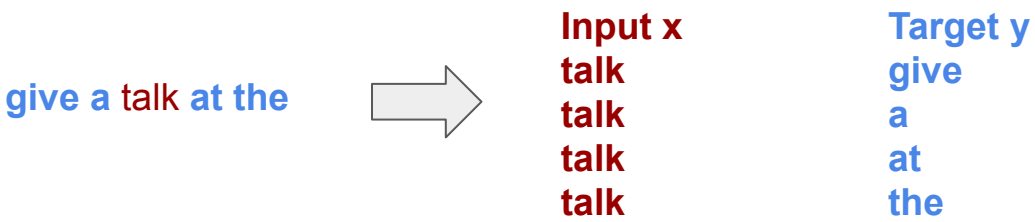
Input and Output Words

- How to select them from corpus
- Skip-gram and CBoW differ here



Skip-Gram

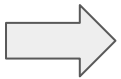
- Task Definition: given a specific word, predict its nearby word (probability output)
- Model input: source word, Model output: nearby word
- Input is one word, output is one word
- The output can be interpreted as prob. scores, which are regarded as how likely it is that each vocabulary word can be nearby your input word.



CBoW

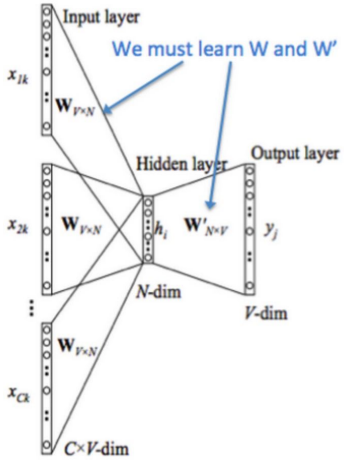
- Task Definition: given context, predict its target word
- Model input: context (several words), Model output: center word
- Input is several words, output is one word
- Core Trick: **average** these context vectors for prob. score computing

give a **talk** at the



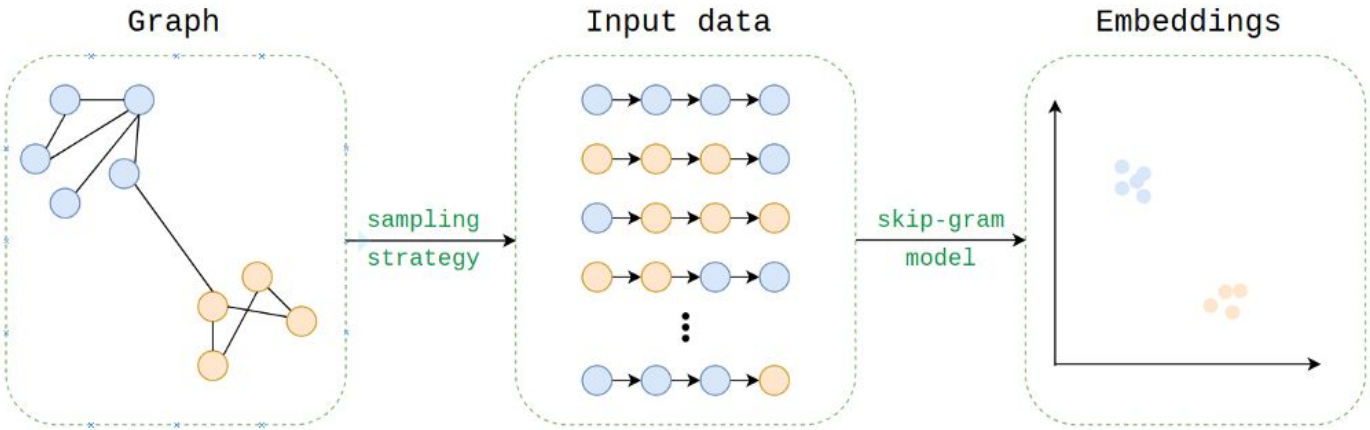
Input x
(give,a,at,the)

Target y
talk



Embedding for Graph Data

- Embeddings can be extended beyond NLP domain
- Embeddings can be learned for any nodes in a graph
- Nodes can be items, web pages and so on in user clicked stream data
- Embeddings can be learned for any group of discrete and co-occurring states.

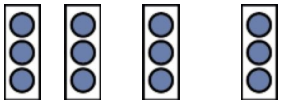


3. Neural Networks for NLP

Sequence of Words

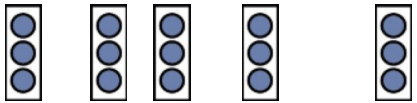
- Each sentence or document can be regarded as a sequence of vectors.
- The shape of matrix depends on the length of sequence. However, the majority of ML systems need fixed-length feature vectors.
- One simple solution: average the sequence of vectors, just like bag-of-words (abandon order information).

I hate this movie



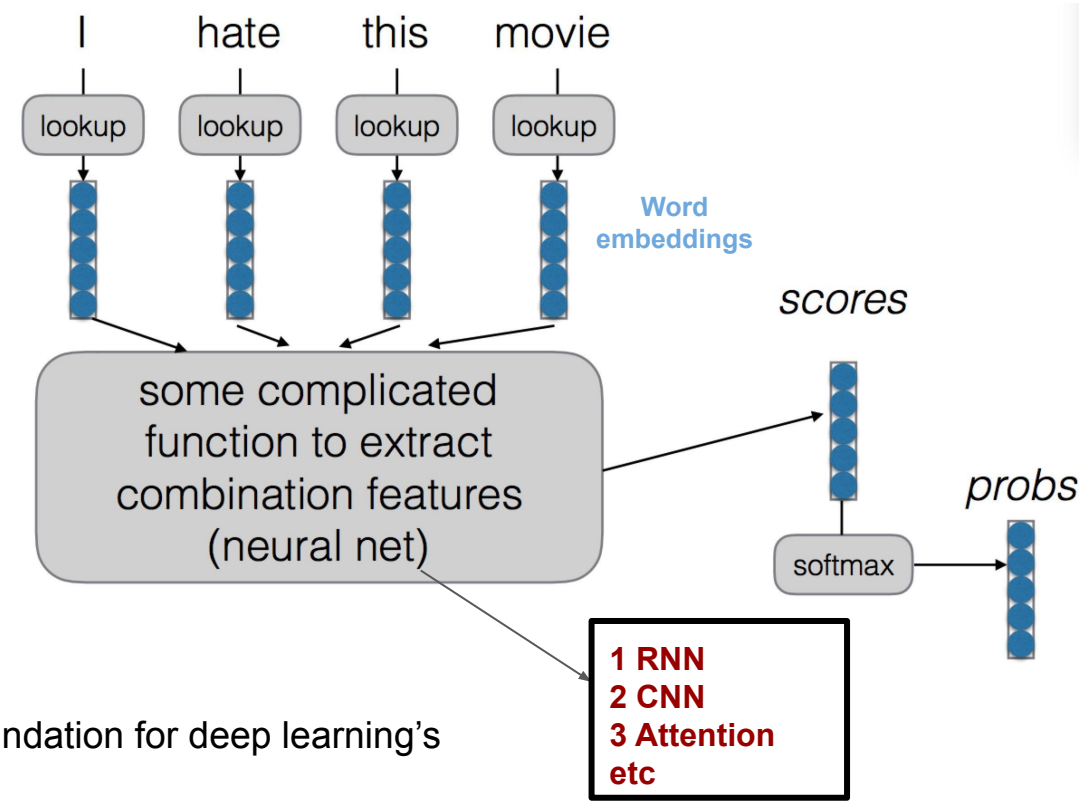
4 by d

This is my favorite movie.



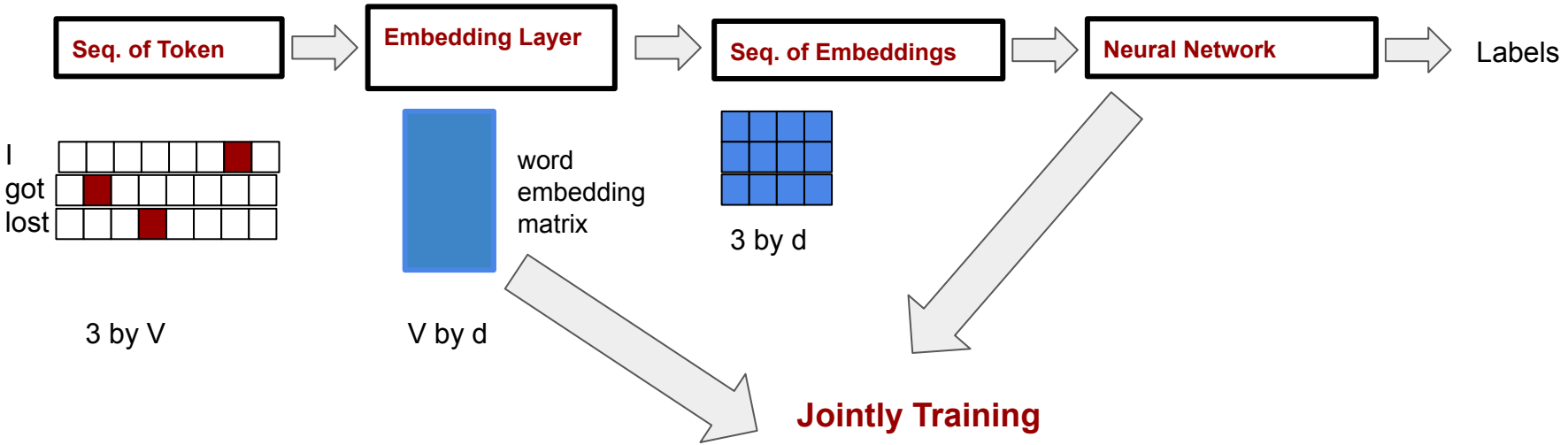
5 by d

Complex Semantic



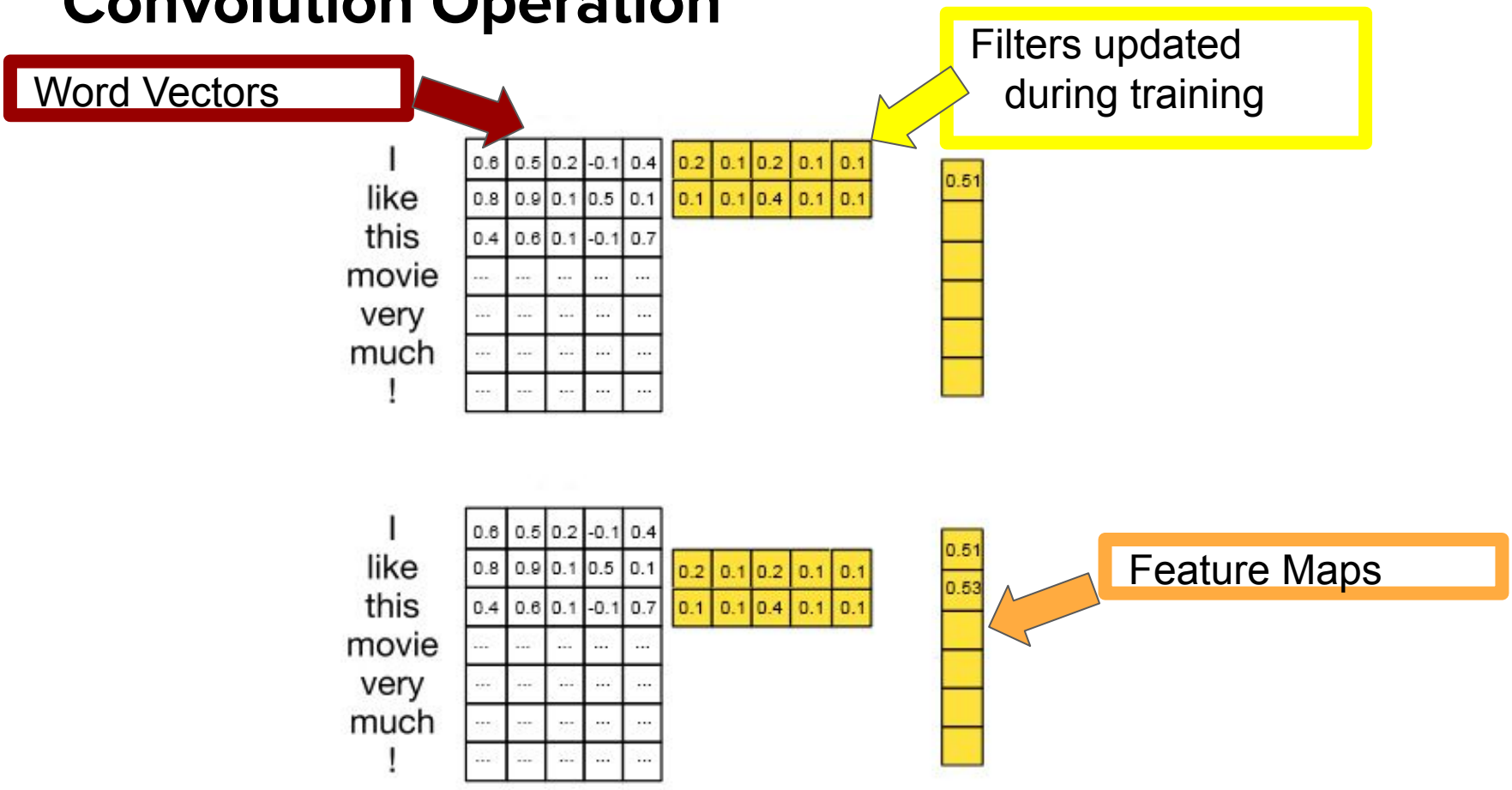
Word Embeddings is the foundation for deep learning's applications on NLP

Neural Networks for NLP

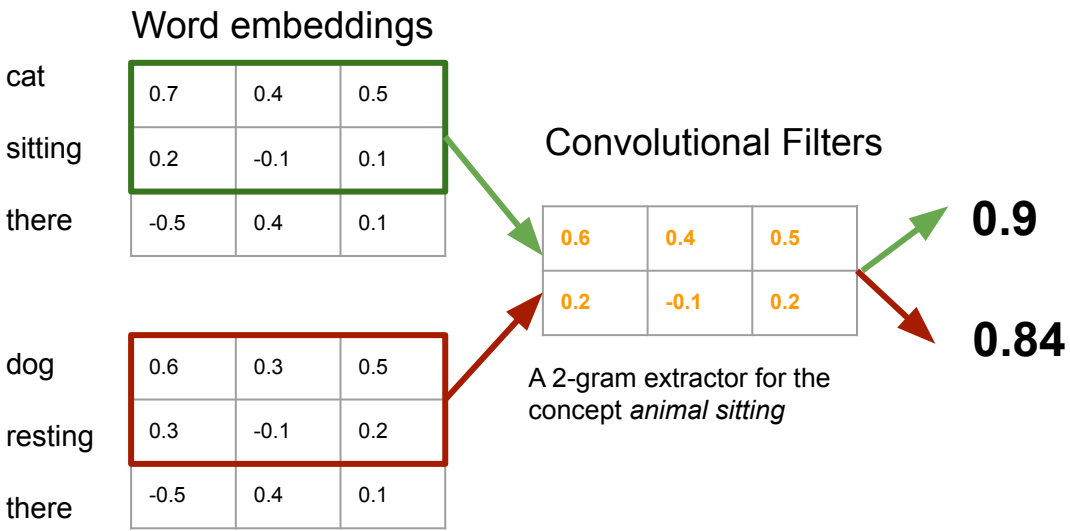


- Learn from Scratch: Random initialize the word embedding matrix and update the matrix and neural network parameters in the specific task
- Pre-train: Got pre-trained word embeddings as the embedding layer and only update neural network parameters in the specific task
- **Pre-train then fine tune**

Convolution Operation



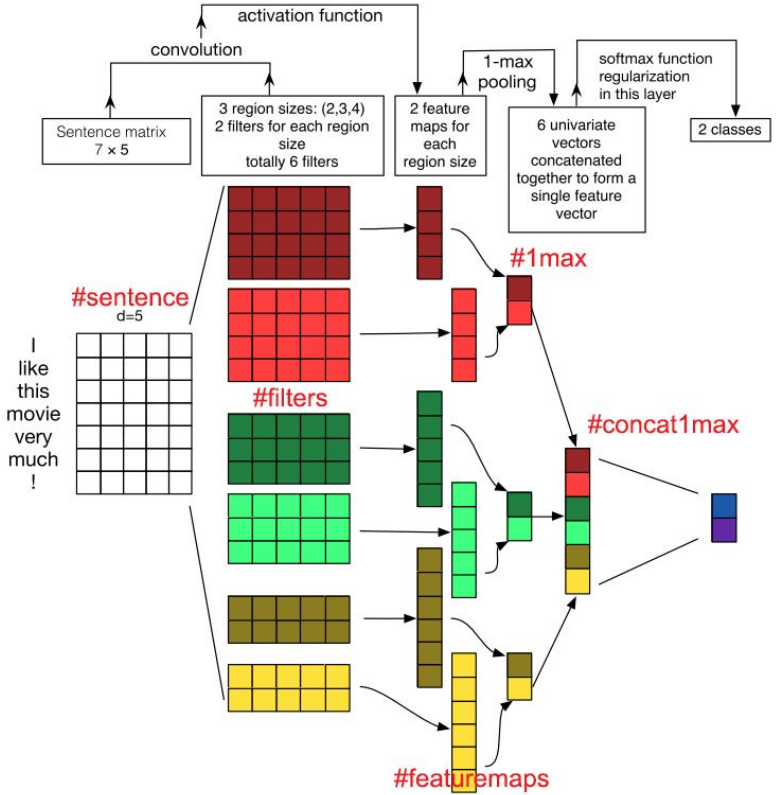
Toy Example



- This convolution provides high activations for 2-grams with certain meaning
- Can be extended to 3-grams, 4-grams, etc.
- Can have various filters, need to track many n-grams.
- They are called 1D since we only slice the windows only in one direction

Why is it better than BoW?

CNN Framework



From Zhang 2015

How to build word embedding layer

- Learn from Scratch: Random initialize the word embedding matrix and update the matrix and neural network parameters in the specific task
- Pre-train: Got pre-trained word embeddings as the embedding layer and only update neural network parameters in the specific task
- Pre-train then fine tune

Pre-train then Fine-tune

Text without annotation

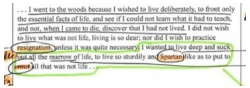
A model that can “read” text

Pre-train

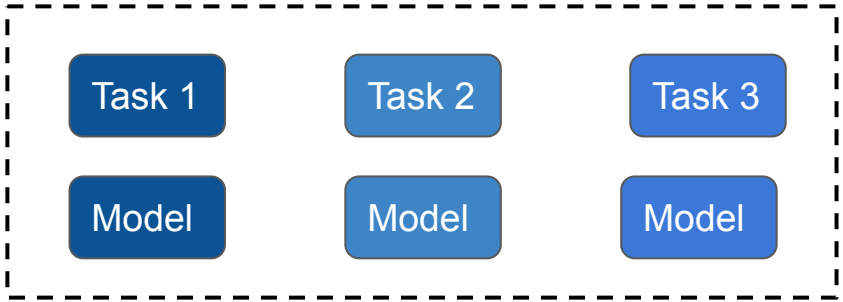
Model

In this lecture, the model is only referred to word embedding matrix. However, in state-of-arts NLP techniques, the model will also include the following neural network.

Fine-tune



Task-specific data with labels

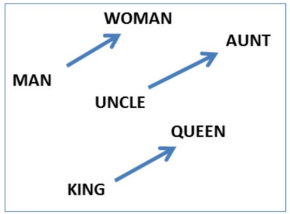


Is Word2Vec good enough?

- Can not capture different senses of words (context independent)
  - Solution: Take the word order into account
- Can not address Out-of-Vocabulary words
  - Solution: Use characters or subwords

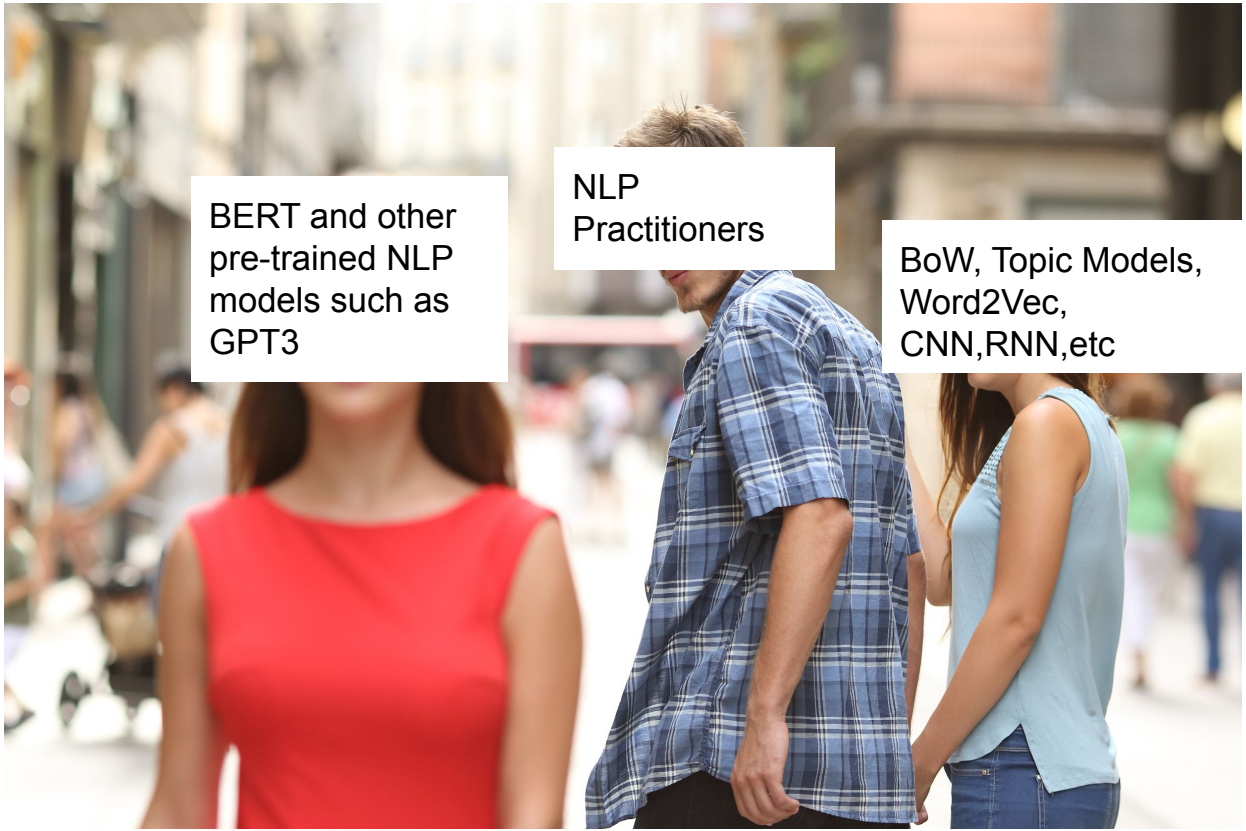
Corpus

Word2Vec Tool

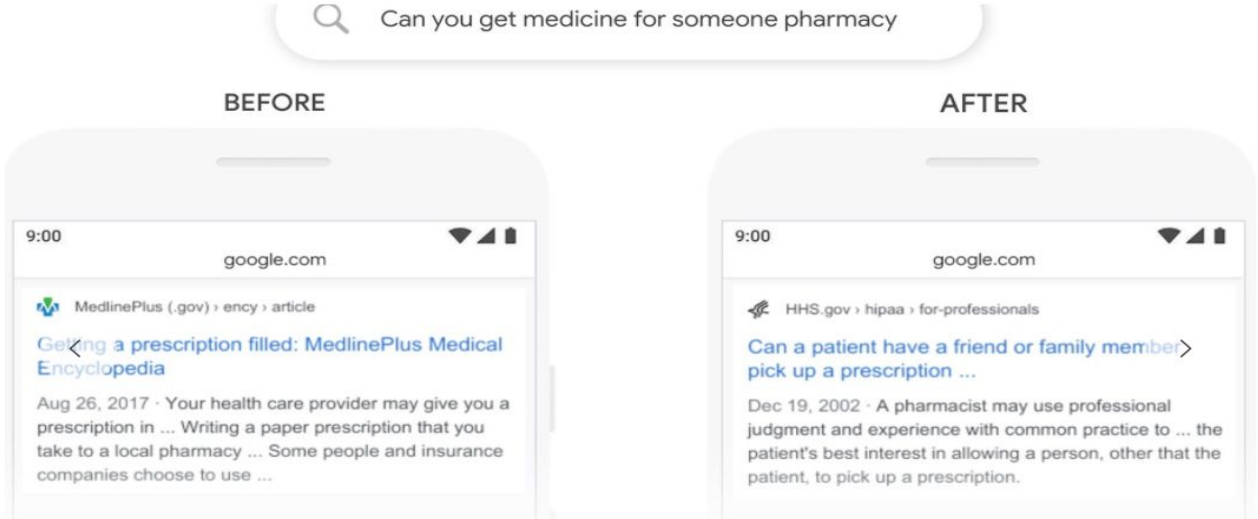


Word Embeddings used for downstream tasks

4. Introduction to BERT


BERT and other pre-trained NLP models such as GPT3
NLP Practitioners
BoW, Topic Models, Word2Vec, CNN,RNN,etc

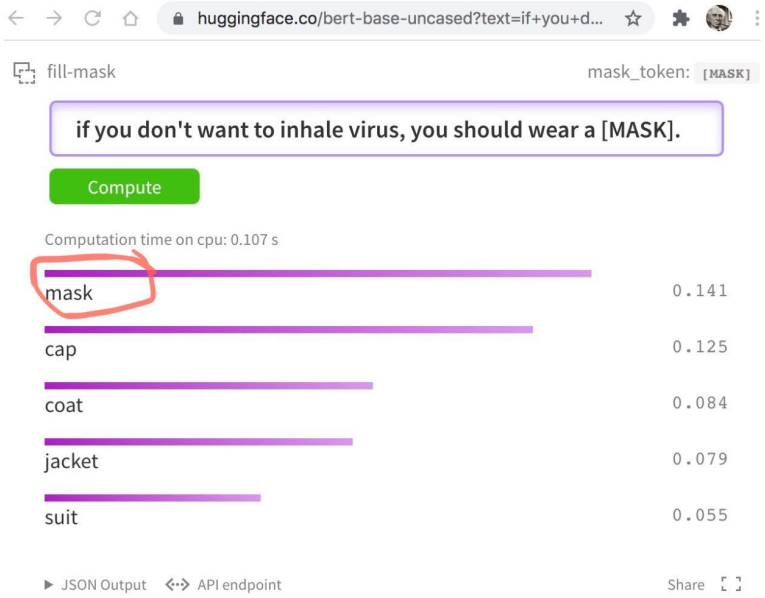
BERT in Google Search



With the latest advancements from our research team in the science of language understanding--made possible by machine learning--we're making a significant improvement to how we understand queries, representing the biggest leap forward in the past five years, and one of the biggest leaps forward in the history of Search.

https://blog.google/products/search/search-language-understanding-bert/

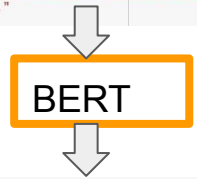
BERT vs Somebody



Try:
https://huggingface.co/bert-base-uncased?text=if+you+don%27t+want+to+inhale+virus+is%2C+you+should+wear+a+%5BMASK%5D

Extraction-based QA using BERT

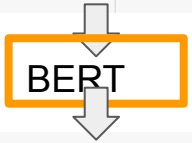
```
question = "What can we learn in NUS?"
answer_text = "The National University of Singapore (NUS) is the national research university of Singapore. \
               Founded in 1905 as the Straits Settlements and Federated Malay States Government Medical School, NUS is the oldest higher education institution in Singapore. \
               It is consistently ranked within the top 20 universities in the world and is considered to be the best university in the Asia-Pacific. \
               NUS is a comprehensive research university, \
               offering a wide range of disciplines, including the sciences, medicine and dentistry, design and environment, law, arts and social sciences, engineering, business, computing and music \
               at both the undergraduate and postgraduate levels."
```




```
print('Answer: "' + answer + '"')
```

```
Answer: "sciences , medicine and dentistry , design and environment , law , arts and social sciences , engineering , business , computing and music"
```

```
question = "What does NUS mean?"
answer_text = "The National University of Singapore (NUS) is the national research university of Singapore. \
               Founded in 1905 as the Straits Settlements and Federated Malay States Government Medical School, NUS is the oldest higher education institution in Singapore. \
               It is consistently ranked within the top 20 universities in the world and is considered to be the best university in the Asia-Pacific. \
               NUS is a comprehensive research university, \
               offering a wide range of disciplines, including the sciences, medicine and dentistry, design and environment, law, arts and social sciences, engineering, business, computing and music \
               at both the undergraduate and postgraduate levels."
```

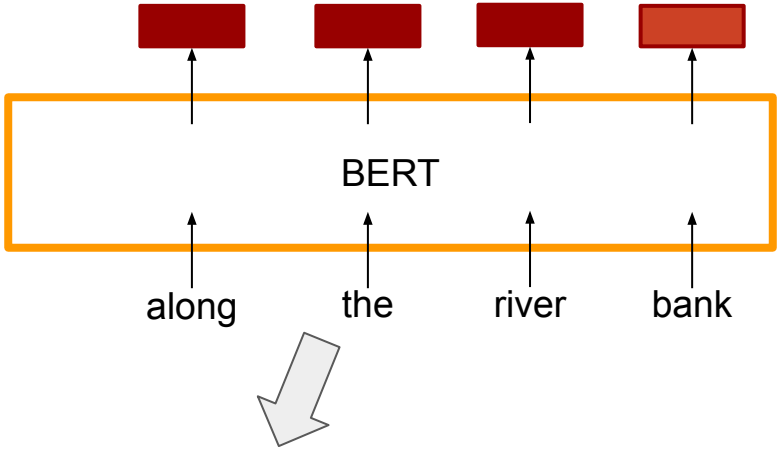



```
print('Answer: "' + answer + '"')
```

```
Answer: "national university of singapore"
```

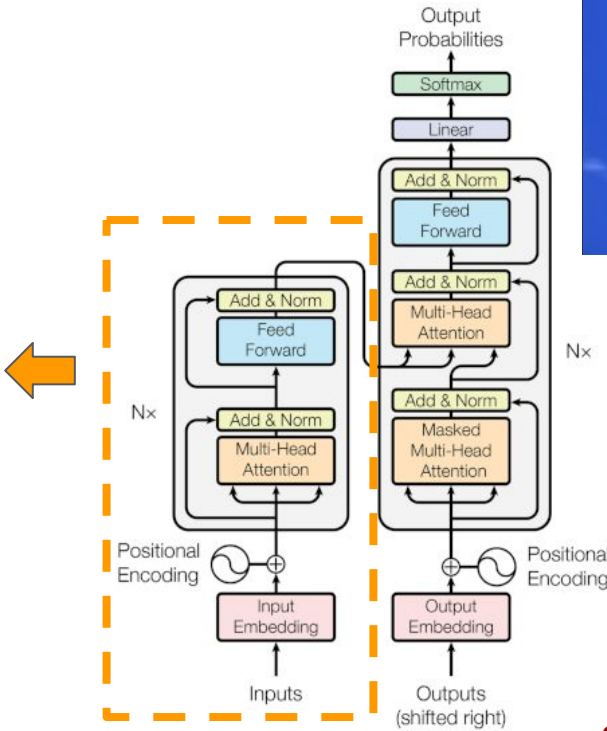
What is BERT

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (**BERT**)
- BERT: Encoder of Transformer,



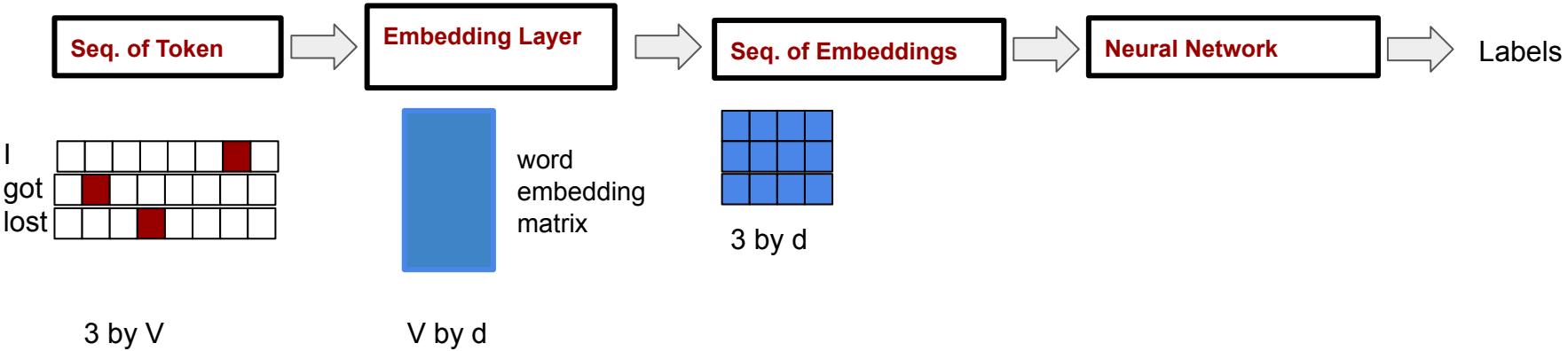
Given a sequence of words, generate a sequence of vectors and then can be used for various NLP tasks

Transformer



Solve Seq2Seq Task

Neural Networks for NLP



Each word has a fixed vector

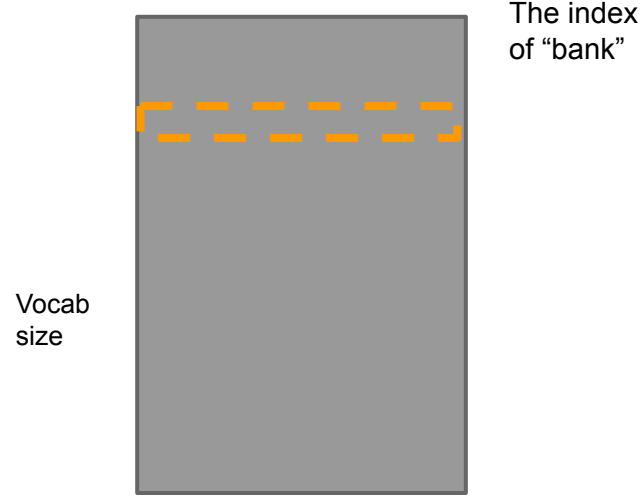
Can not address multi-sense problem!

Multiple Senses of Words

- It is safest to deposit your money in the bank.
- All the animals lined up along the river bank.
- Today, blood banks collect blood.

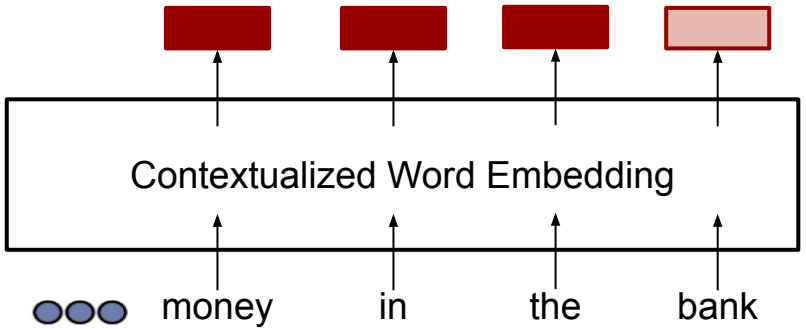
The third sense of not?

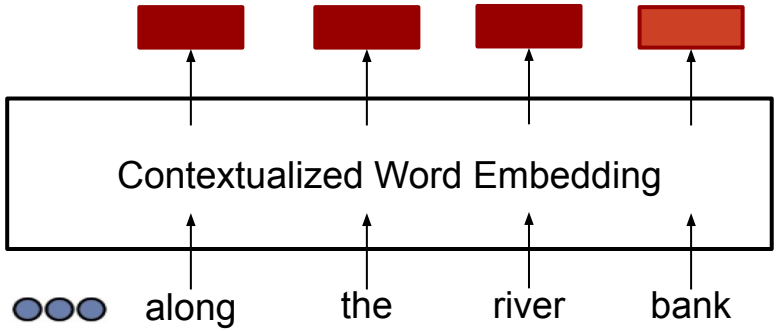
Word2Vec, Fasttext,Glove and other word embedding models

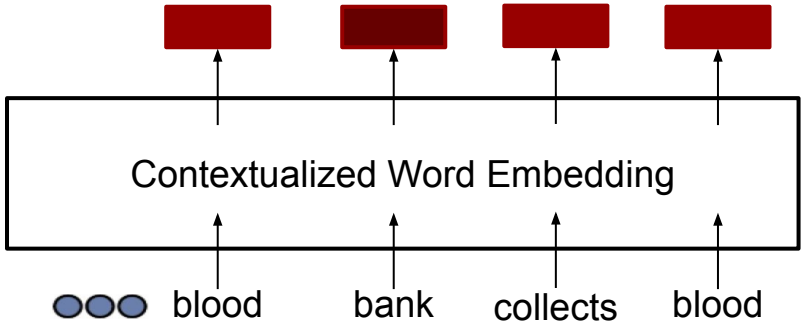


Contextualized Word Embeddings

Encoded Embeddings







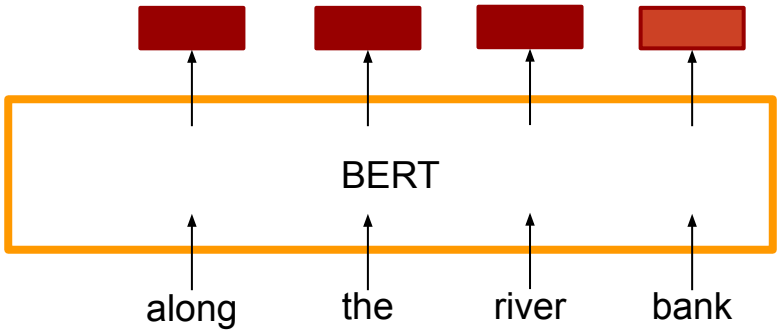
Different Contexts,
Different Encoded Embeddings for bank.

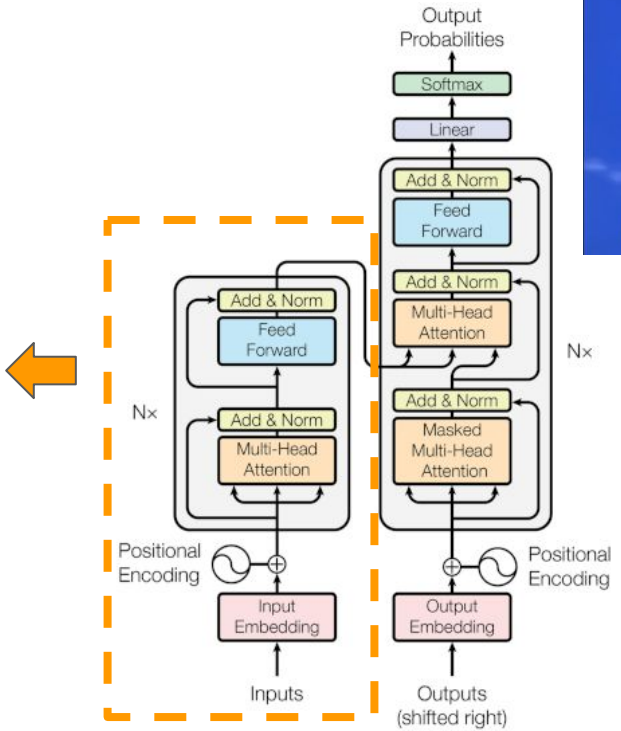
Embeddings generated from BERT

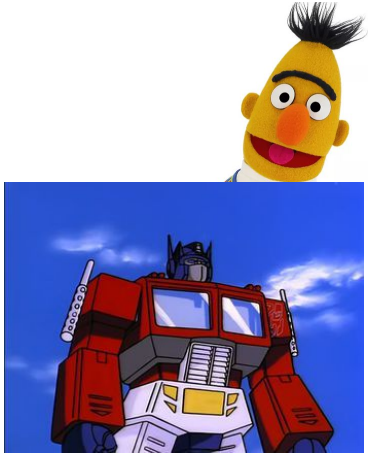
Cos-similarities among vectors of “apple” in different context

BERT

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (**BERT**)
- BERT: Encoder of Transformer,





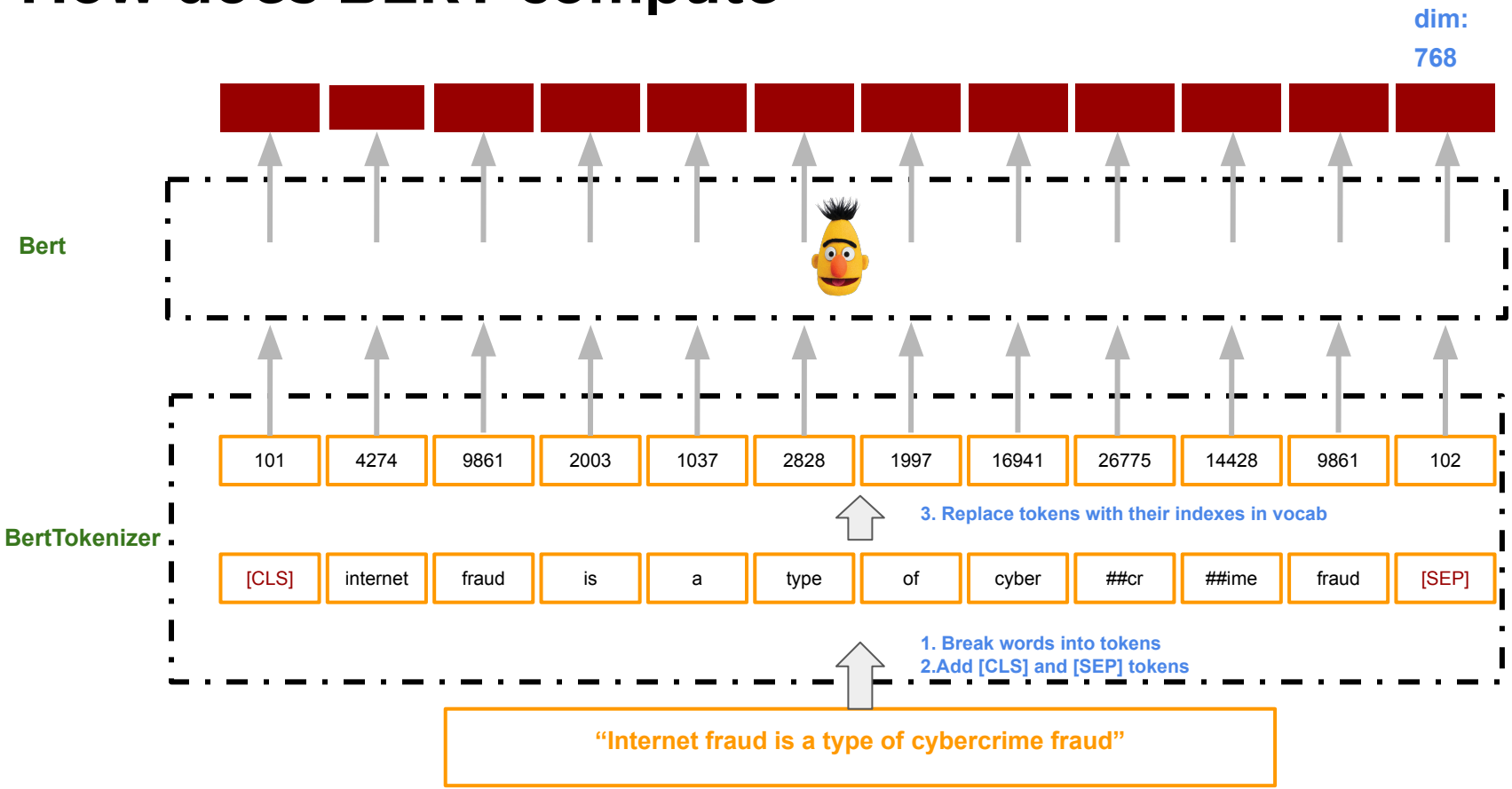


Details behind Transformers

1. https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/
2. http://jalammar.github.io/illustrated-transformer/

Visualizing Ichiban!!!

How does BERT compute



How to use BERT

Text without annotation

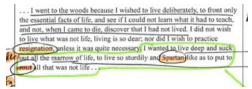
Pre-train

A model that can “read” text

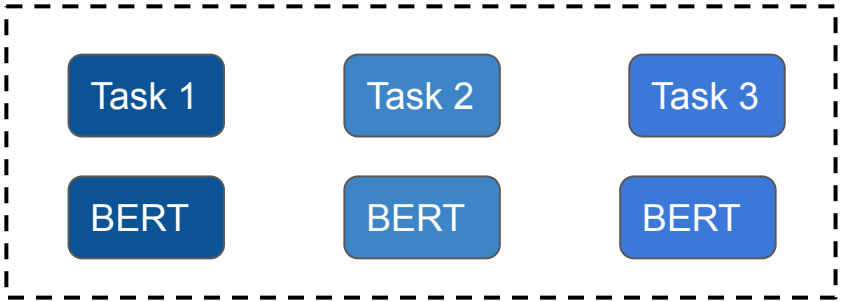
BERT

Learned from a large amount of **text without annotation**

Fine-tune

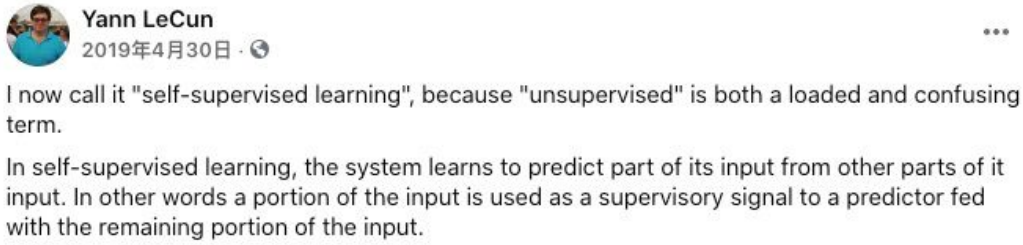


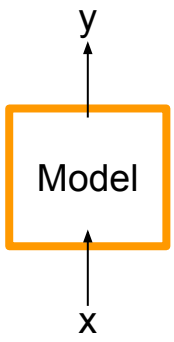
Task-specific data with labels



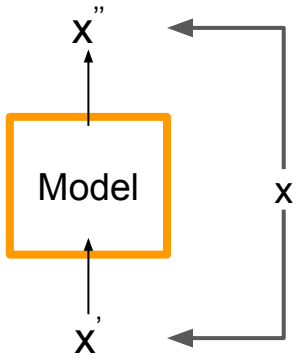
How to Pre-Train

The answer is **self-supervised learning**.





Supervised



Generated by Rules

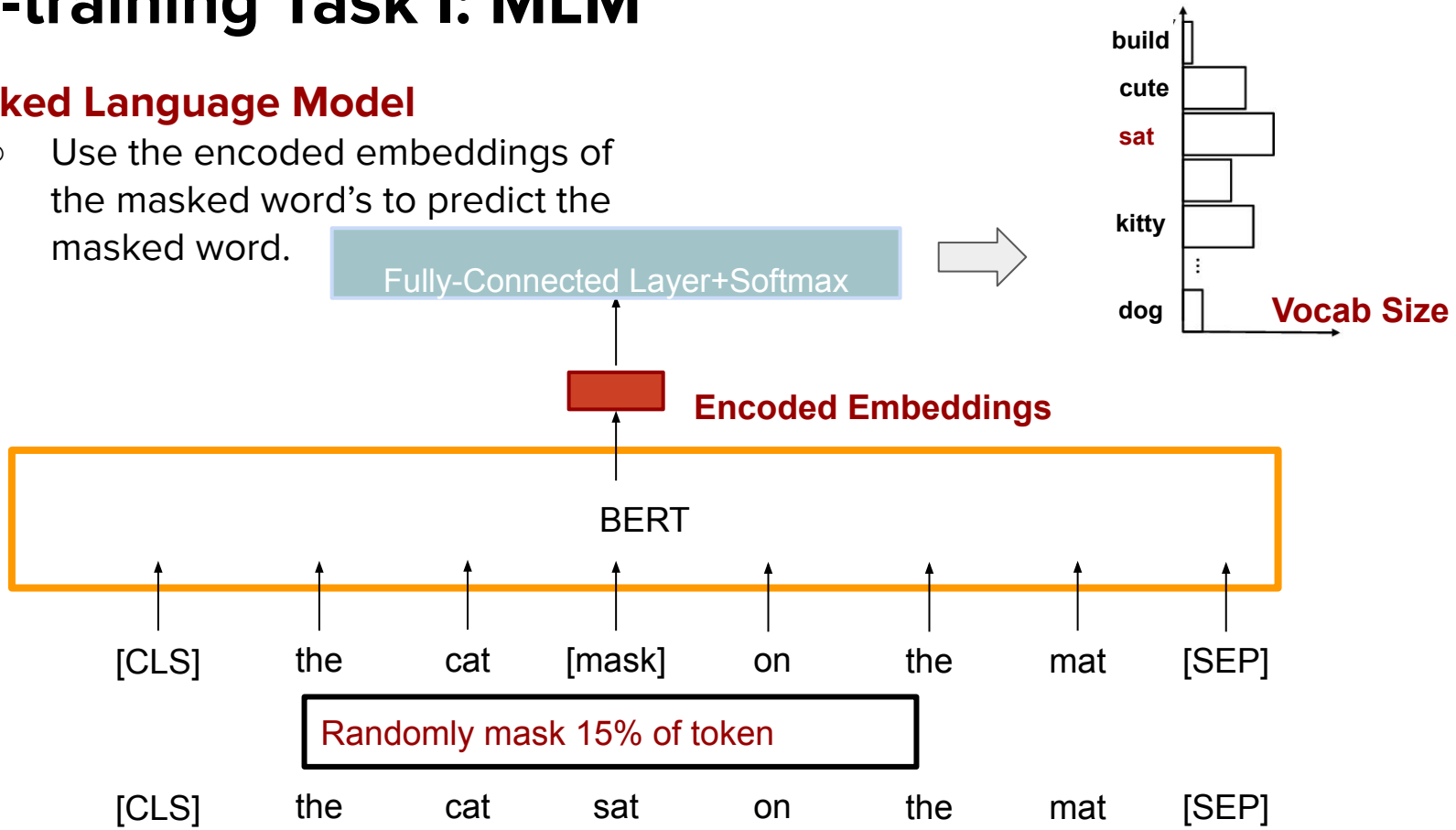
x

Automatically generate some kind of supervisory tasks

Self-Supervised

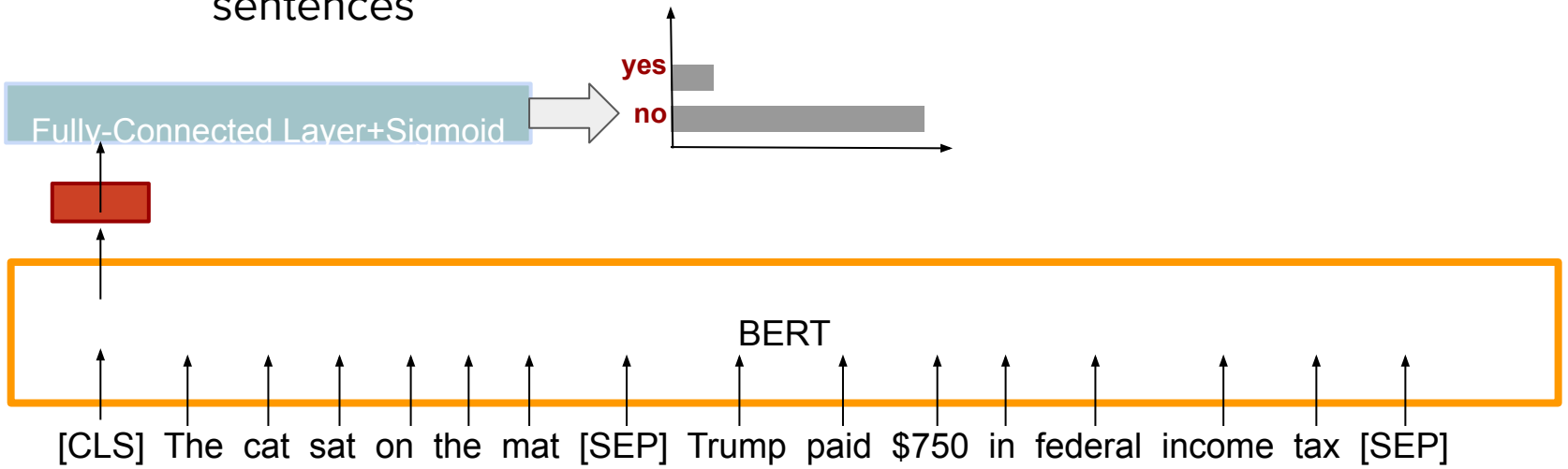
Pre-training Task I: MLM

- **Masked Language Model**
  - Use the encoded embeddings of the masked word's to predict the masked word.

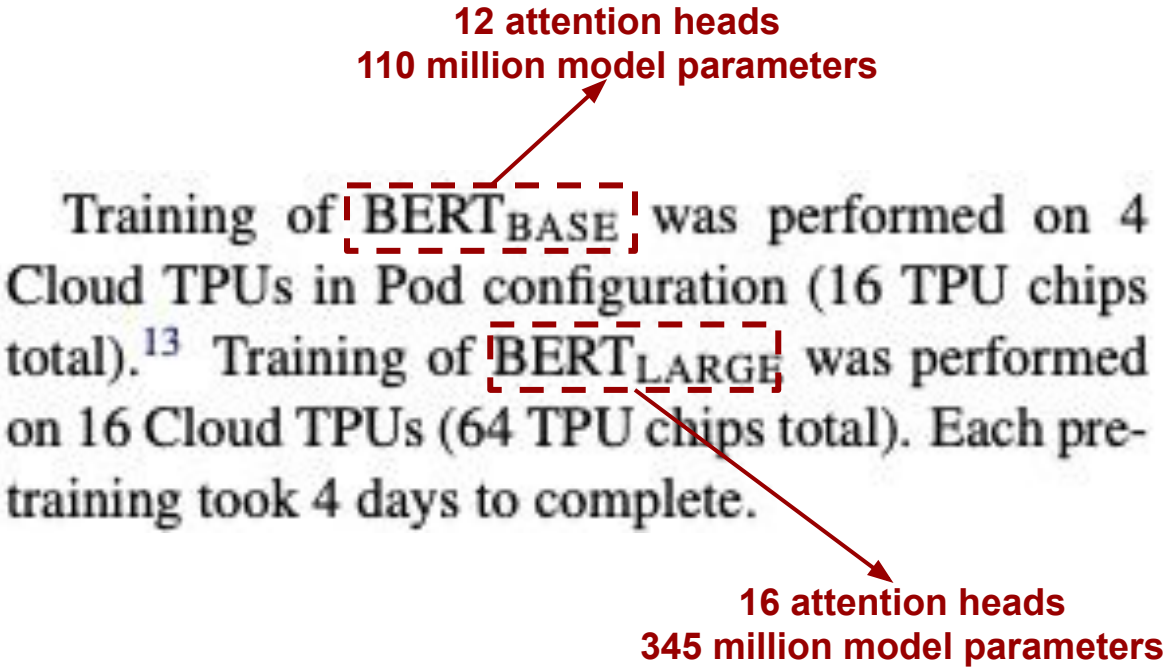


Pre-training Task II: NSP

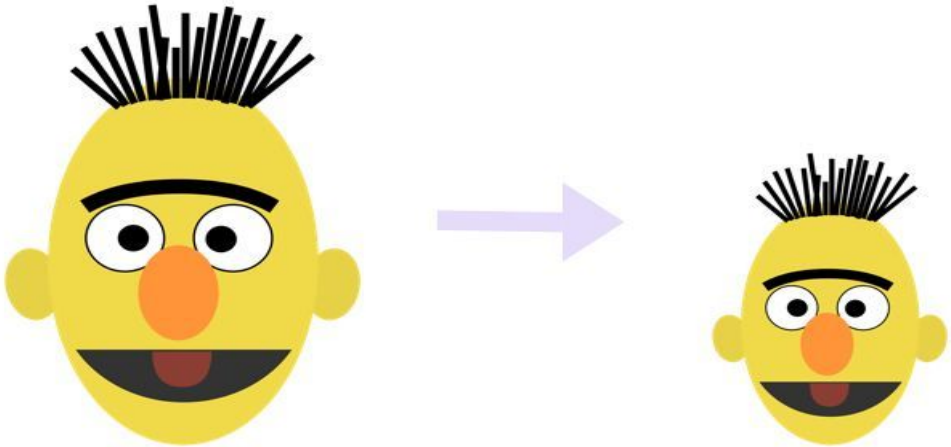
- **Next sentence prediction**
  - Given two sentences A and B, is B likely to be the sentence followed by A?
  - Make bert good at handling relationships between multiple sentences



Huge Model Size



Smaller Model



Published as a conference paper at ICLR 2020

ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS

Zhenzhong Lan[1] Mingda Chen[2*] Sebastian Goodman[1] Kevin Gimpel[2]
Pivush Sharma[1] Radu Soricut[1]

DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter

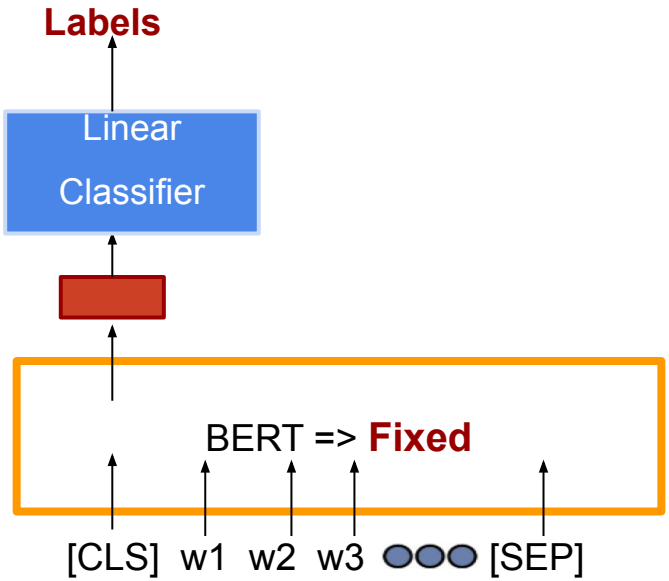
Victor SANH, Lysandre DEBUT, Julien CHAUMOND, Thomas WOLF
Hugging Face
{victor,lysandre,julien,thomas}@huggingface.co

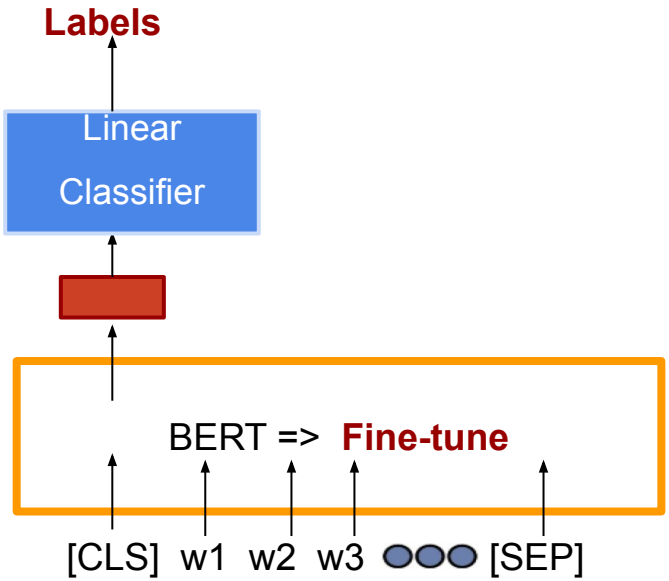
Abstract

Good summary:
http://mitchgordon.me/machine/learning/2019/11/18/all-the-ways-to-compress-BERT.html

BERT Usage I

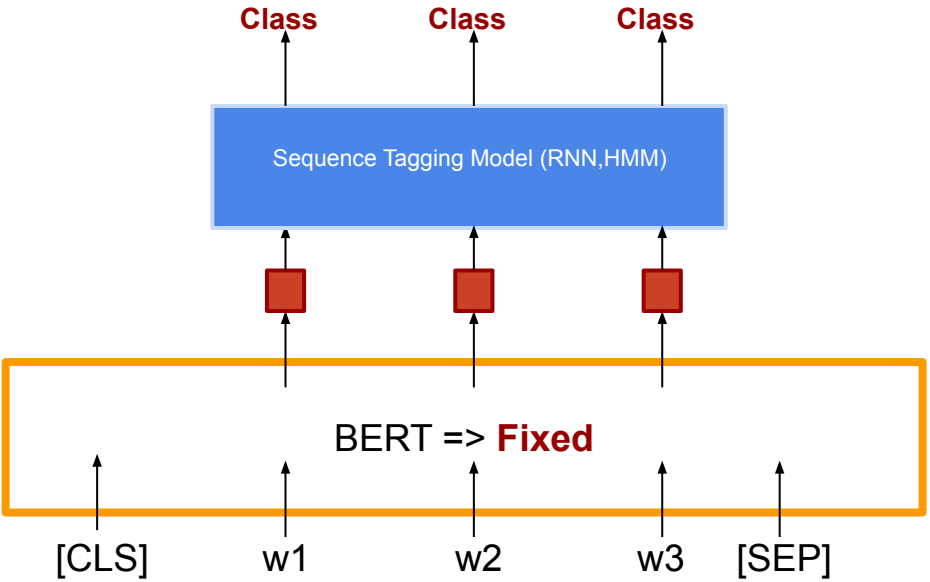
- **Input: Single Sentence Output: Class**
  - Sentiment Analysis
  - Document Classification

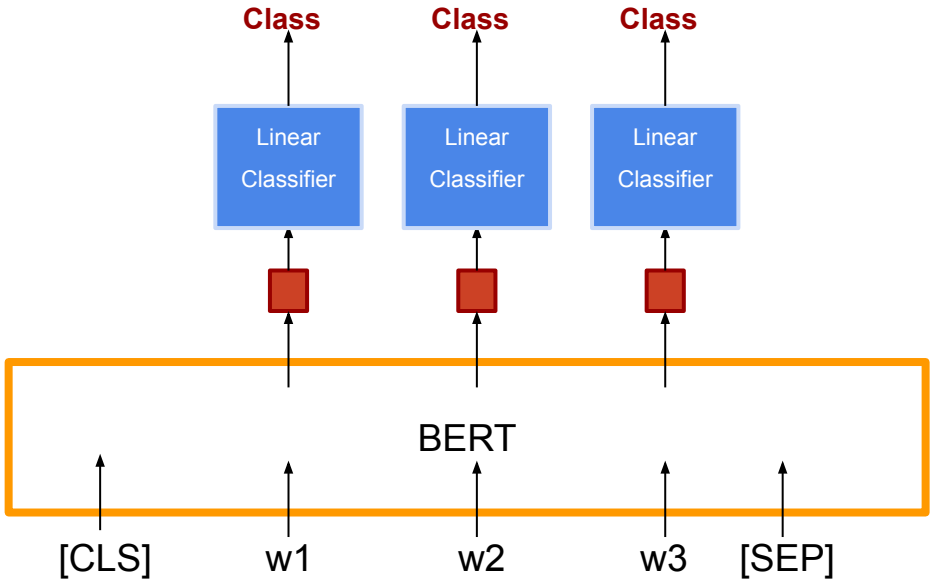




BERT Usage II

- **Input: Single Sentence Output: Class**
  - NER, POS Tagging

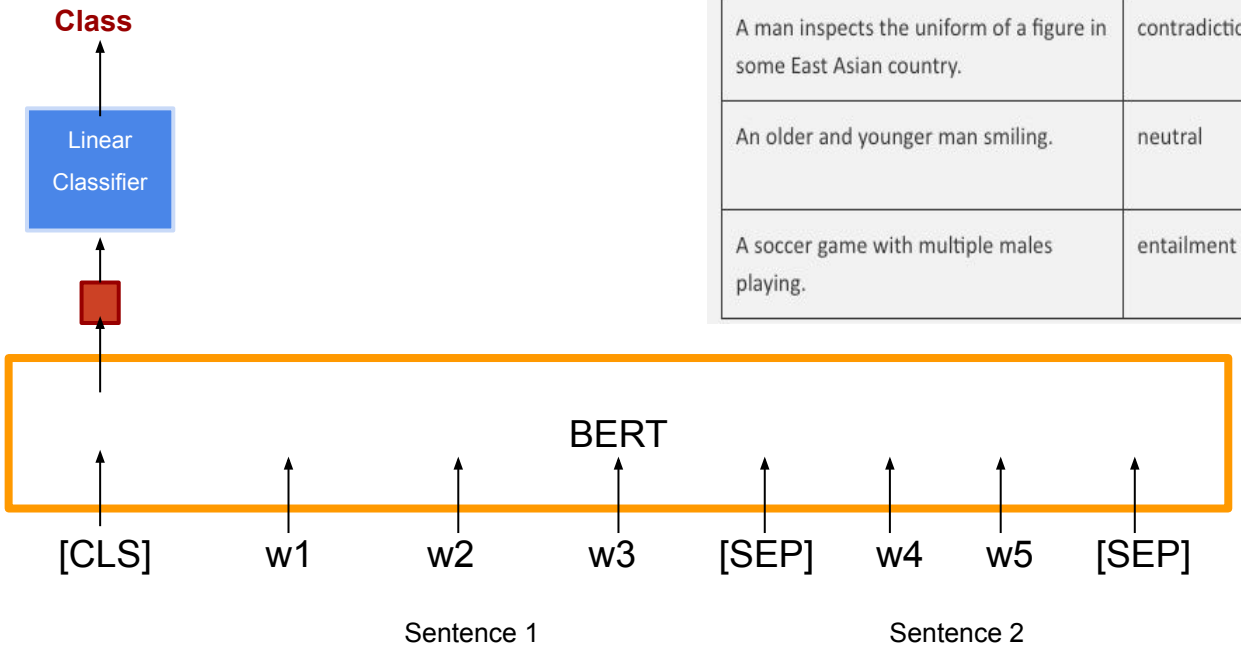




BERT Usage III

- **Input: Two Sentences Output: Class**
  - Natural Language Inference

Premise	Label	Hypothesis
A man inspects the uniform of a figure in some East Asian country.	contradiction	The man is sleeping.
An older and younger man smiling.	neutral	Two men are smiling and laughing at the cats playing on the floor.
A soccer game with multiple males playing.	entailment	Some men are playing a sport.



BERT Usage IV

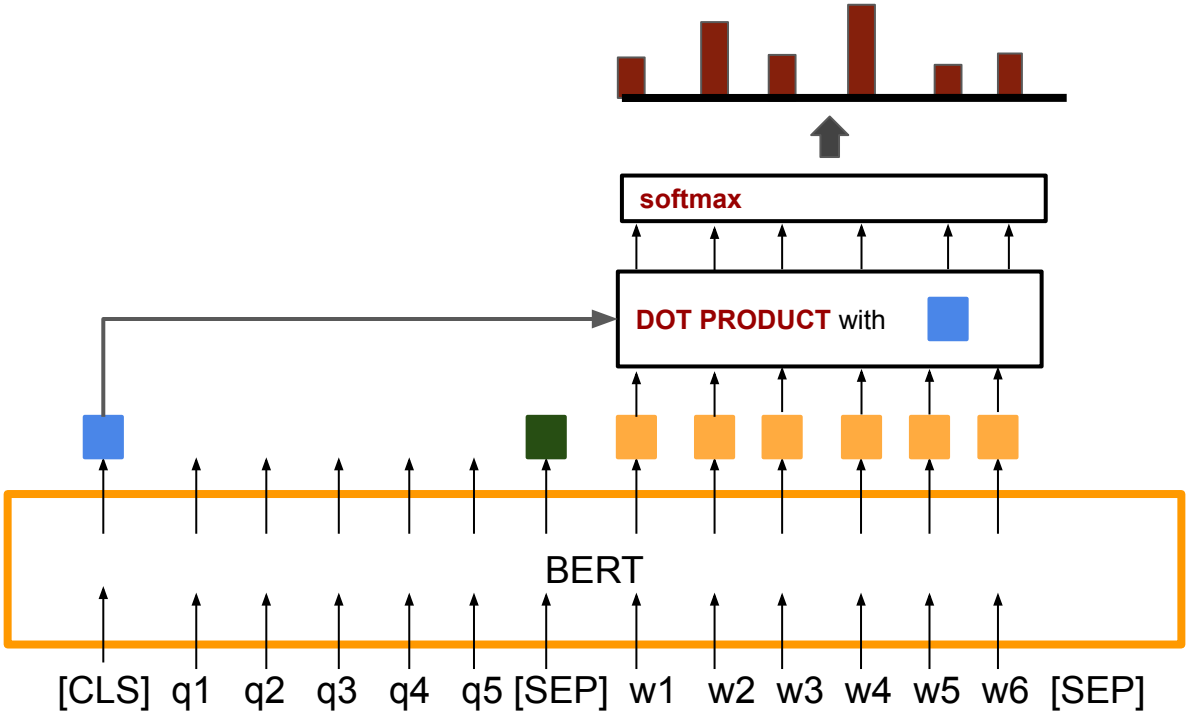
- Extraction-based Question Answering (SQuAD):
  - Input: two “sentences” (Question and Reference Text)
  - Output: start and end positions in Reference (Answer)

Question: How many parameters does BERT-large have?

Reference Text: BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

BERT Usage IV

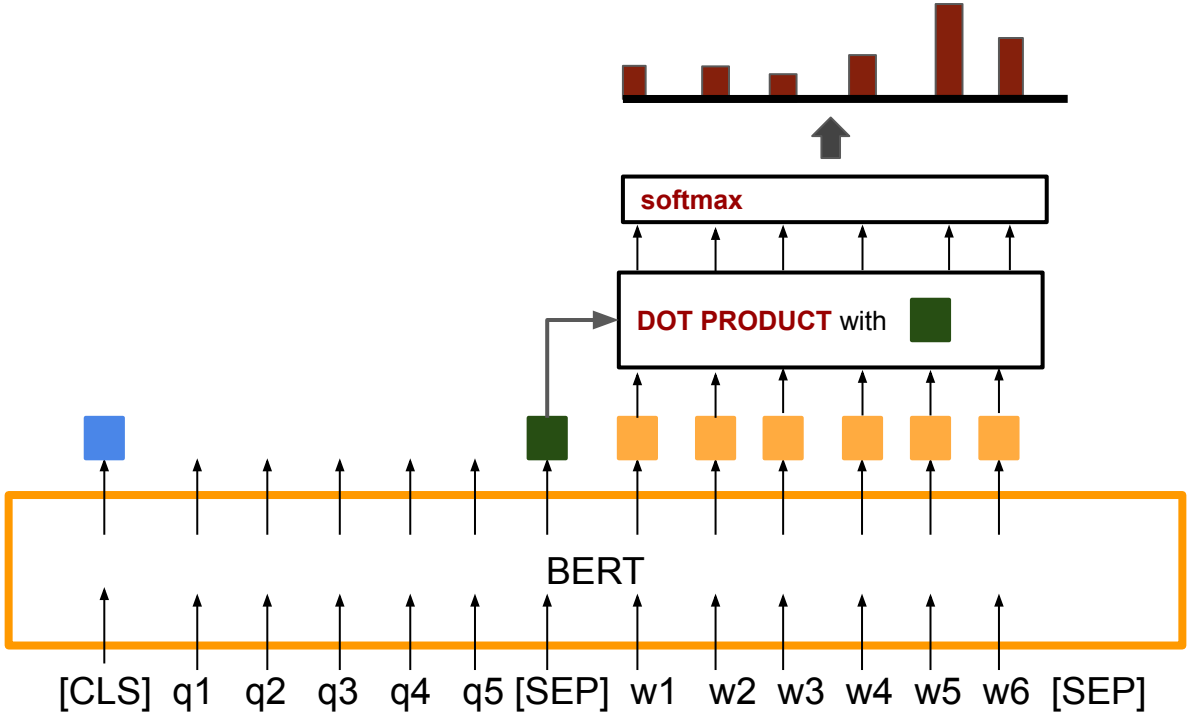
- Extraction-based Question Answering (SQuAD):
  - Question {q1,q2,q3,q4,q5} Reference Text{w1,w2,w3,w4,w5,w6}



The starting position for answer in reference is 4

BERT Usage IV

- Extraction-based Question Answering (SQuAD):
  - Question {q1,q2,q3,q4,q5} Reference Text{w1,w2,w3,w4,w5,w6}



The starting position for answer in reference is 4

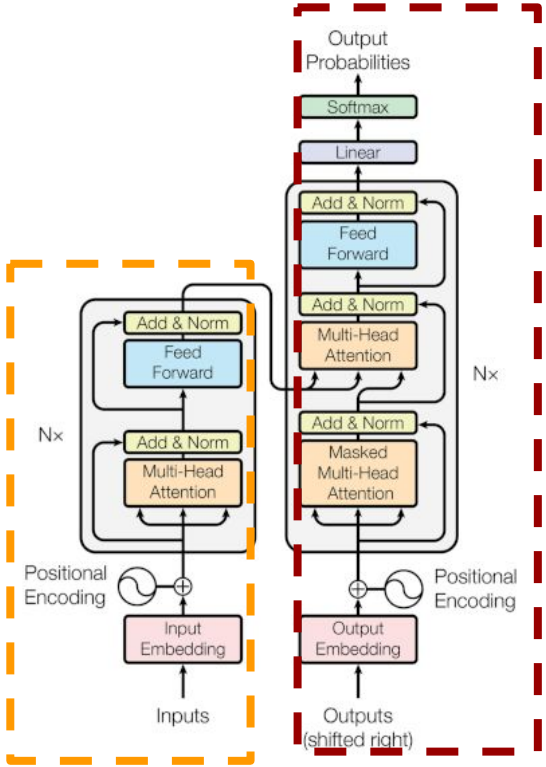
The ending position for answer in reference is 5

The answer is w4w5

BERT, GPT and Transformers

BERT

Encoder
Maksed Language Model



Transformer

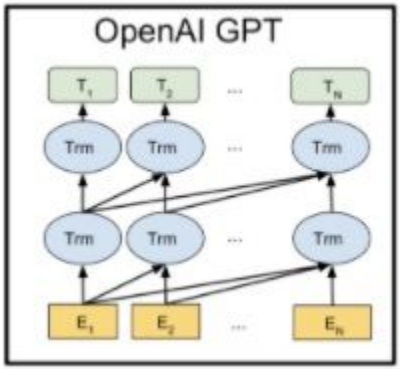
GPT2
GPT3

OpenAI

GPT-3, an autoregressive language model with 175 billion parameters

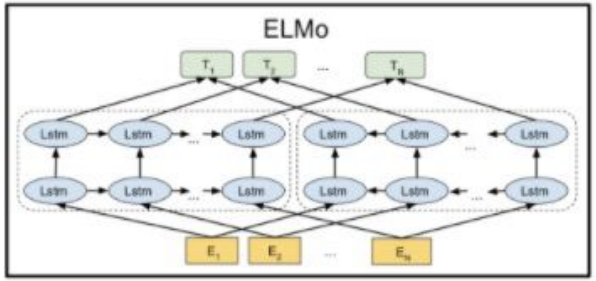
Decoder
Language Model

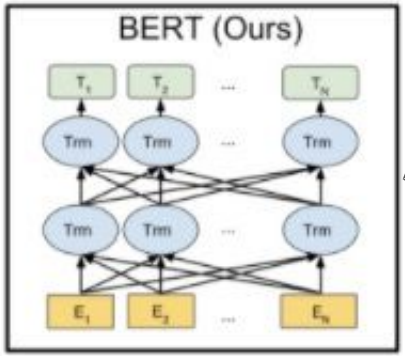
BERT and other Pre-trained Models



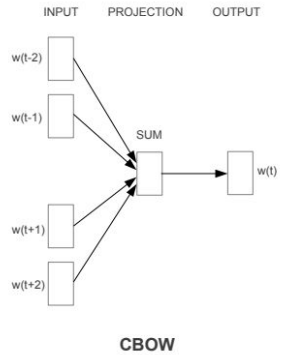
From
single-directional to
Bi-directional

From RNNs to
transformers





From FCNs to
transformers



https://lilianweng.github.io/lil-log/2019/01/31/generalized-language-models.html