VECTOR INSTITUTE

Bias in AI:
Week #2: Introduction to Modern NLP

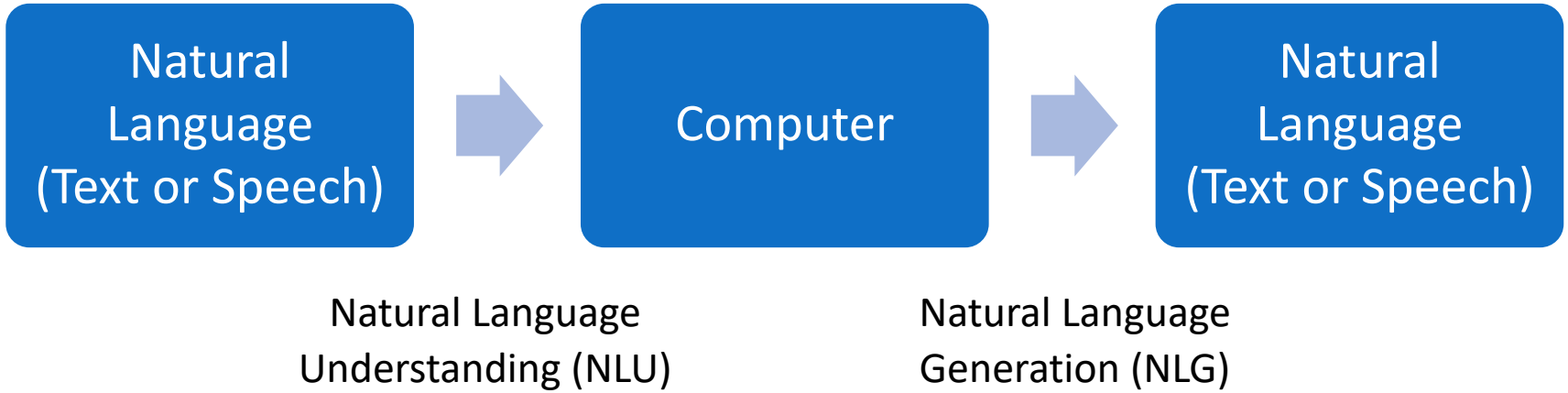
Instructor:
Sayyed Nezhadi

Winter 2023

Definition

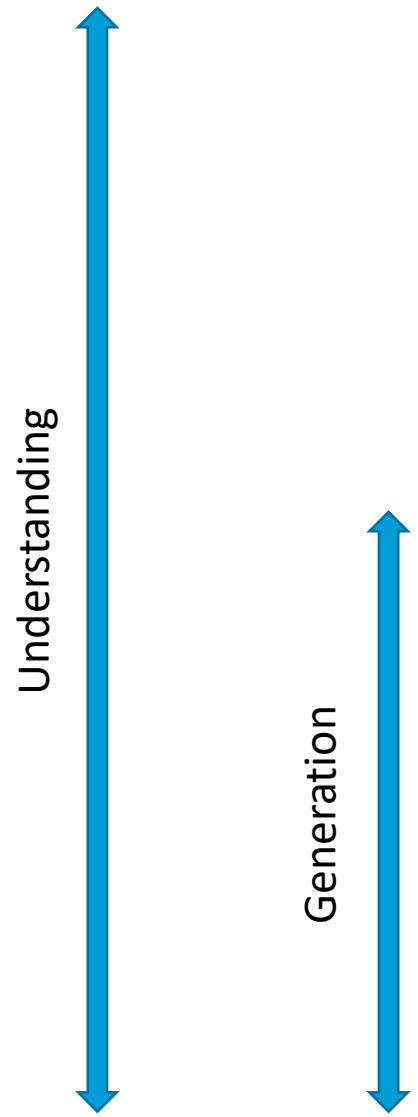
- **Natural Language Processing (NLP)**, is a branch of artificial intelligence that deals with the interaction between computers and humans using the natural language.

- **The main goals of NLP:** To understand and generate natural language.



Sample NLP Applications

- Text Classification (e.g., Spam Filtering)
- Sentiment Analysis / Market Intelligence
- Information Retrieval / Document Parsing
- Search Engines (e.g., Google, Bing)
- Content Recommendation
- Machine Translation (e.g. Google Translate)
- Chatbots (e.g. IBM Watson-powered chatbot)
- Personal Assistants (Siri, Google Home, Alexa)
- Text to Speech (Voice Generation)
- Speech to Text (Voice Transcription)
- Text Summarization
- Question and Answering



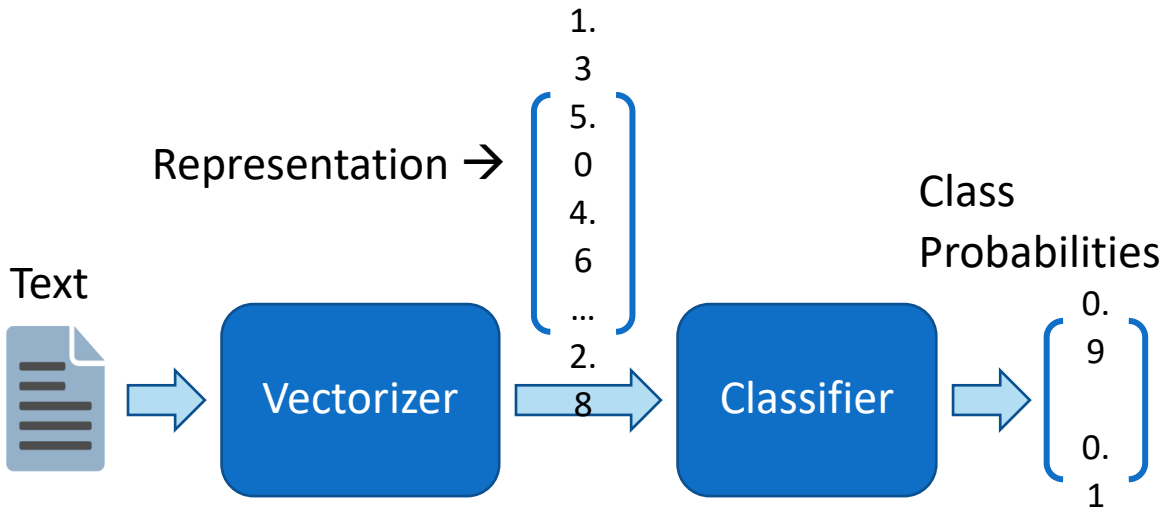
NLP is Hard (Ambiguity in Language)

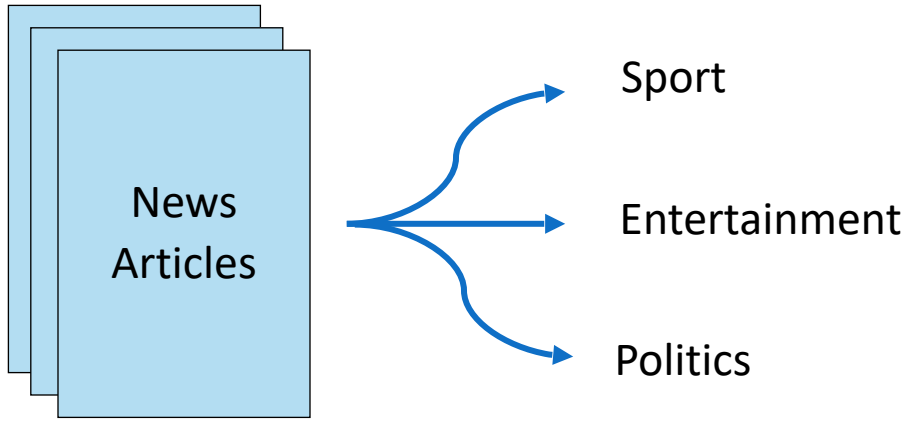
- Lexical Ambiguity:
  - I saw a bat.
  - The fisherman went to the bank.
- Syntactic Ambiguity:
  - The chicken is ready to eat.
  - I ate a salad with dressing from Italy.
  - I threw a brick through the window. It broke.
- Semantic Ambiguity:
  - John and Mary are married." (To each other? or separately?)
  - Can you lift this bag? (Can? Or Will?)
- Misspellings
- Acronyms, slangs, …

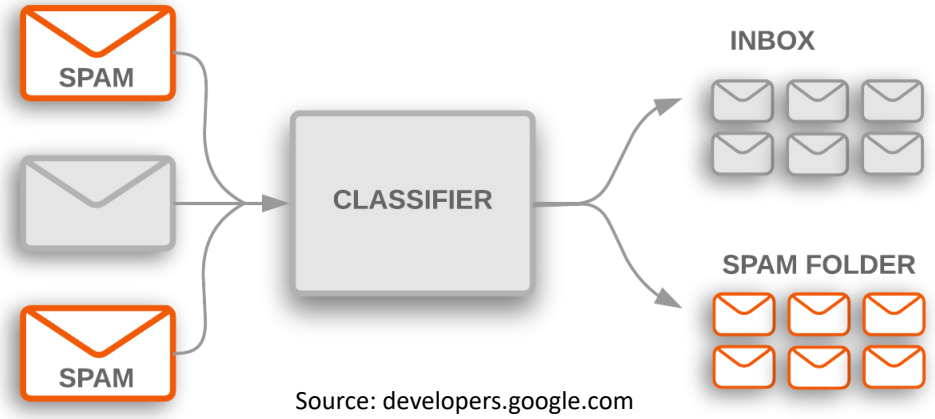
Text Classification

To use a classifier, we need to represent the text by a vector (Vectorization):

- How to represent a word?
- How to represent a text? (variable number of words)







Source: developers.google.com

Word Representation - Discrete

In traditional NLP, words are represented as discrete entities (one-hot vectors):

$$\text{Word} = [x_1, x_2, \ldots, x_D],\ x_i \in [0, 1]$$

D= number of words in vocabulary (e.g., 30,000)

Problems:

- Very large and sparse vectors (not suitable for deep networks)
- Similarities (meaning and word family) are not captured:

House = [0, 0, …, 0, 1, …]

Home = [0, 0, …, 1, 0, …]

* The above vectors are orthogonal

Word Representation – by Context

The idea:

- Represent any word with a dense vector (a.k.a., word embedding), so that the similar words have similar vectors.
- They are also called, “word embedding”

How?

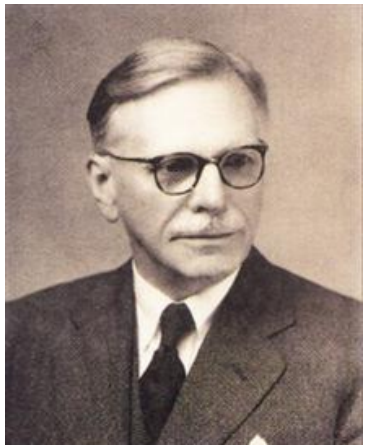
- Use different context of a word to generate a representation (a very successful idea in modern NLP)
- In each text, a word context is the set of words that appear close to the word.

house = $\begin{bmatrix} 0.23 \\ 0.57 \\ -0.45 \\ 0.77 \\ 0.17 \\ -0.99 \\ 0.35 \\ 0.01 \end{bmatrix}$

Word Representation – by Context

You shall know a word by the company it keeps

(Firth, J. R. 1957:11)



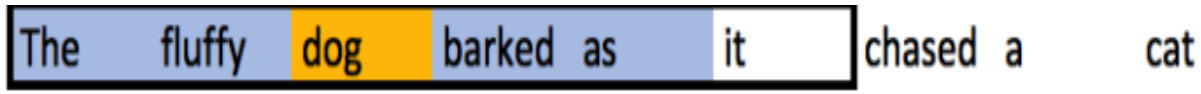
…government debt problems turning into	***banking***	*crises as happened in 2009…*
…saying that Europe needs unified	***banking***	*regulation to replace the hodgepodge…*
…India has just given its	***banking***	*system a shot in the arm…*

These context words will represent ***banking***

Source: cs224n slides Stanford

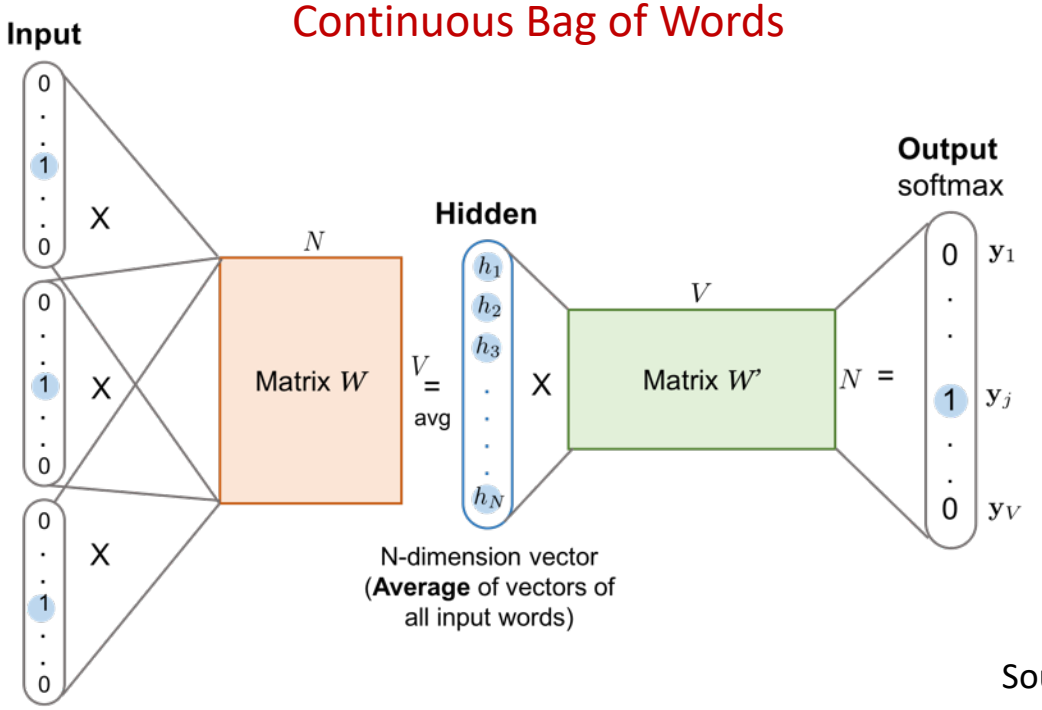
Example: word2vec (Mikolov et al. 2013)

- Using a sliding window to choose a word and its context

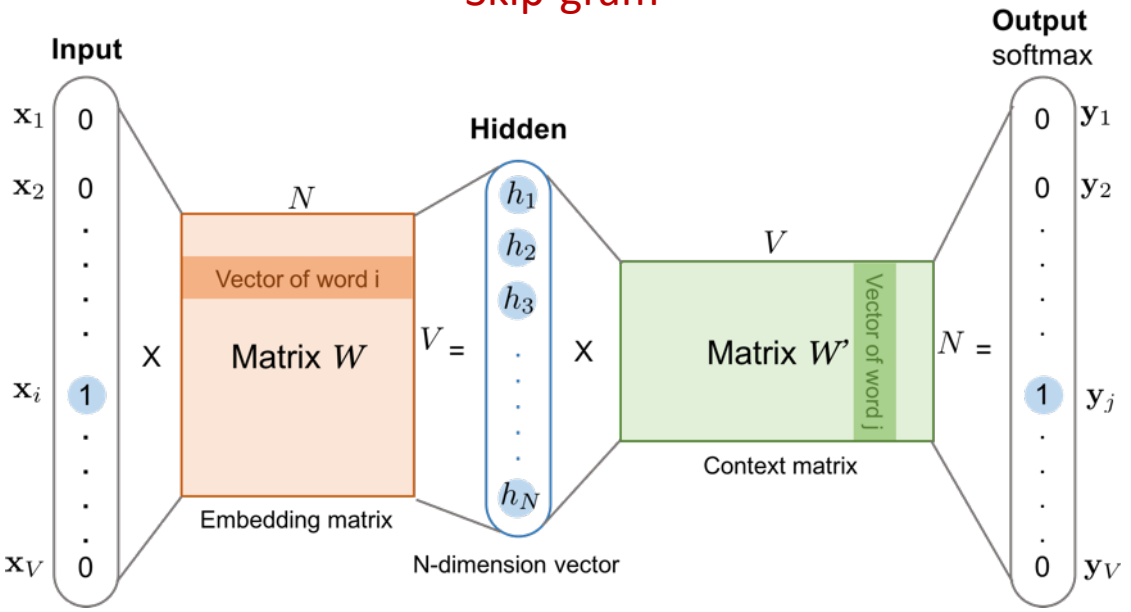


- Two variants:

Continuous Bag of Words



Skip-gram



Source: Lil'Log

Other Methods

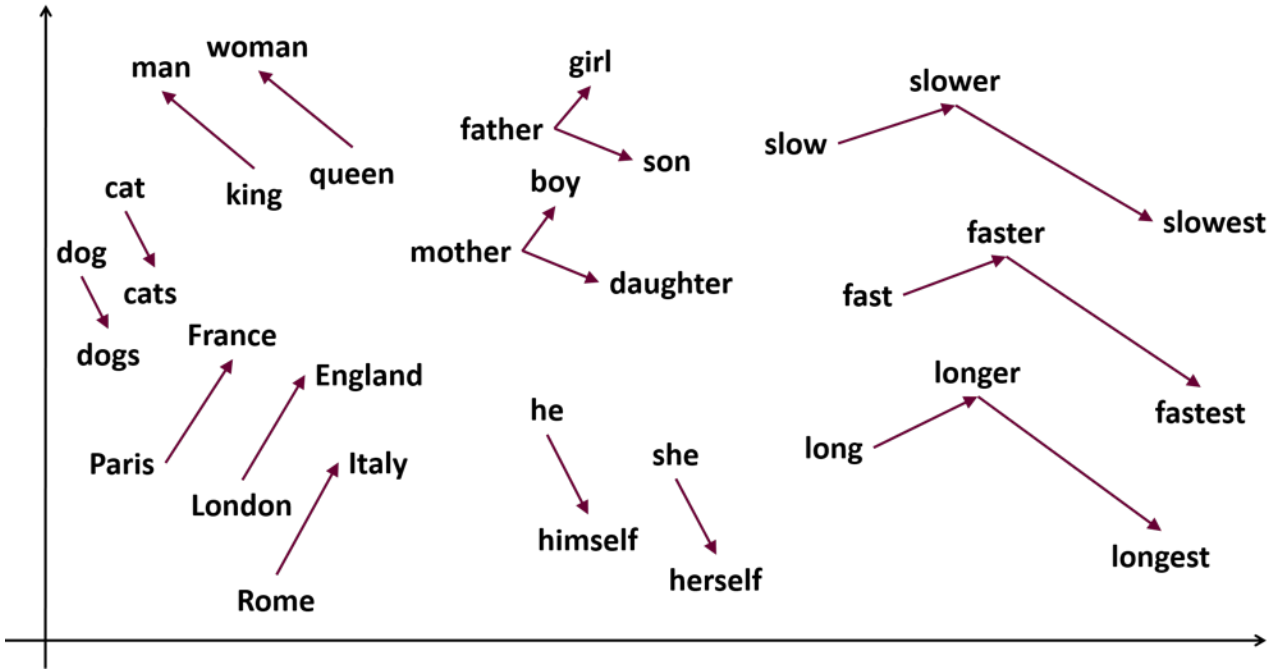
- GloVe: Global Vectors for Word Representation (Pennington et al. 2014)
- fastText: Enriching Word Vectors with Subword Information (Facebook Research 2016)

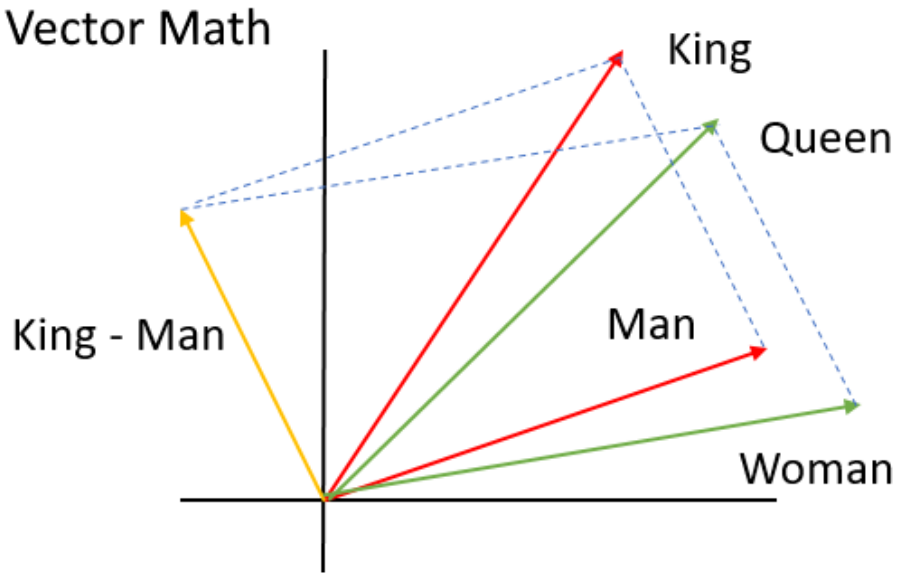
Word Embedding Properties

- Capturing similarities
- Dense vector
  (e.g., 300 dimensions)

- Surprising property: meaningful arithmetic operations

$$Closest\Big(V(king) - V(man) + V(woman)\Big) = V(queen)$$





Bias in Word Embedding

$$\overrightarrow{man} - \overrightarrow{woman} \approx \overrightarrow{computer\ programmer} - \overrightarrow{homemaker}$$

$$\overrightarrow{man} - \overrightarrow{woman} \approx \overrightarrow{doctor} - \overrightarrow{nurse}$$

Source: Bolukbasi et al. 2016

How to Represent a Text

- A couple of ideas:
  - Average the word embedding vectors (remember arithmetic operations are meaningful).
  - Concatenate the word embedding vectors (Challenge: the representation vector will be very long. Also, the number of words is not the same in each text)
- Other methods like "Doc2Vec" (Mikolov et al. 2014)
- But, what about the order of the words? The following sentences are different:

"The dog bit the baby" ≠ "The baby bit the dog"

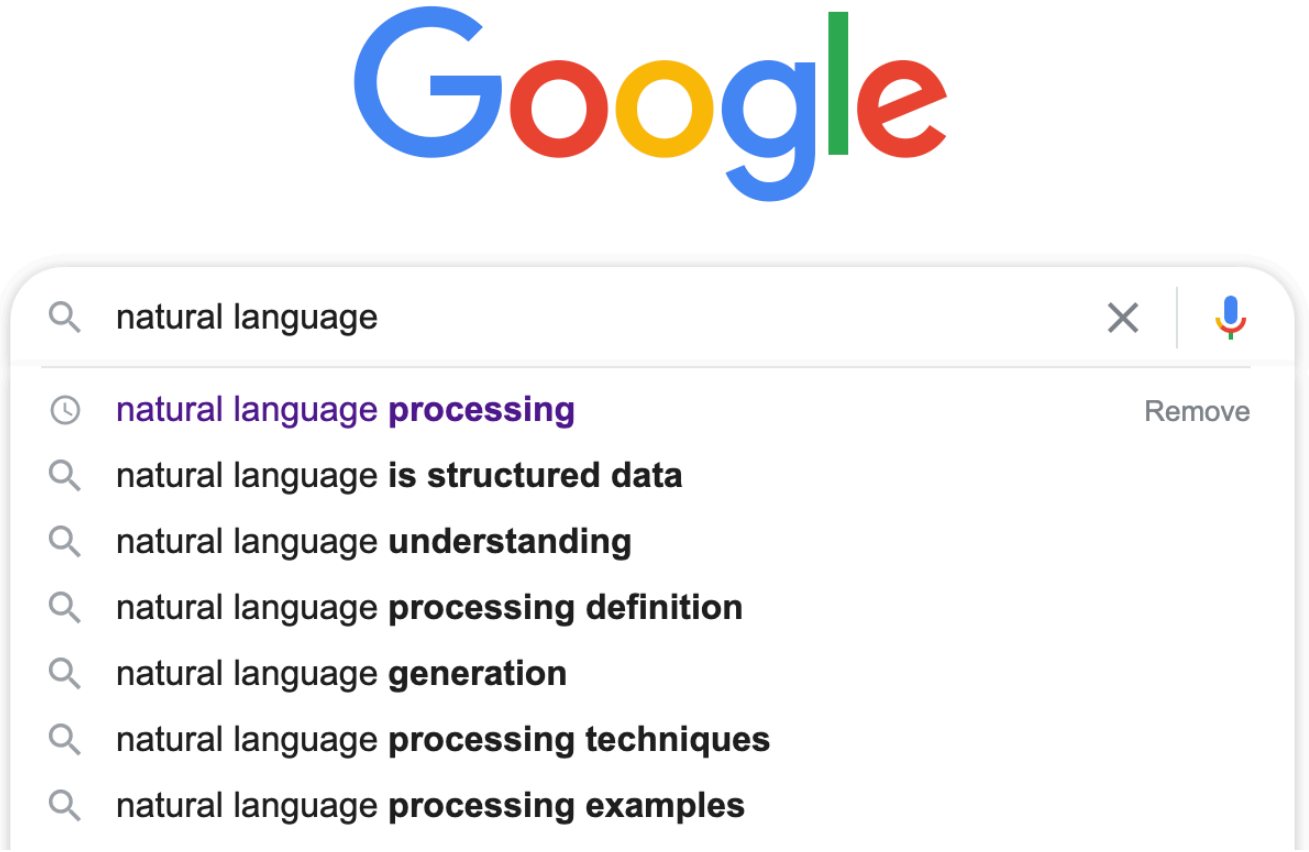
** We need a sequential model **

Language Model

- Consider a “Word Prediction” or “Auto Suggestion” use case.
- Predicting the next word is called Language Modeling.

- A probability distribution over sequences of words, is called Language Model.

$$P(w_{t+1} \mid w_t, w_{t-1}, \ldots, w_1)$$



Example Language Models

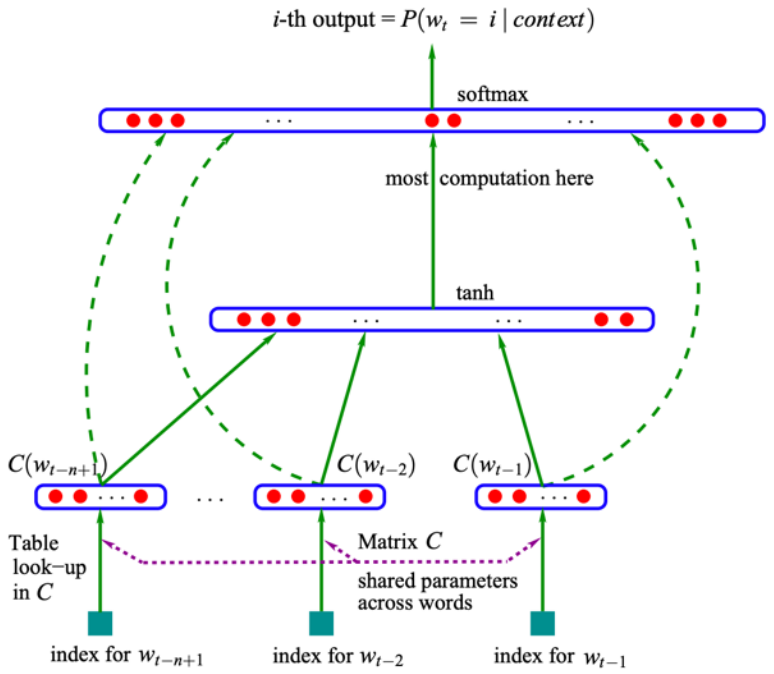
- N-gram Language Model (pre-deep learning era): build and store a probability table for occurrence of words based on previous (n-1) words
  - Sparsity problem
  - Storage problem
  - Fixed window (small)

Bigram example (Dan Jurafsky)

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

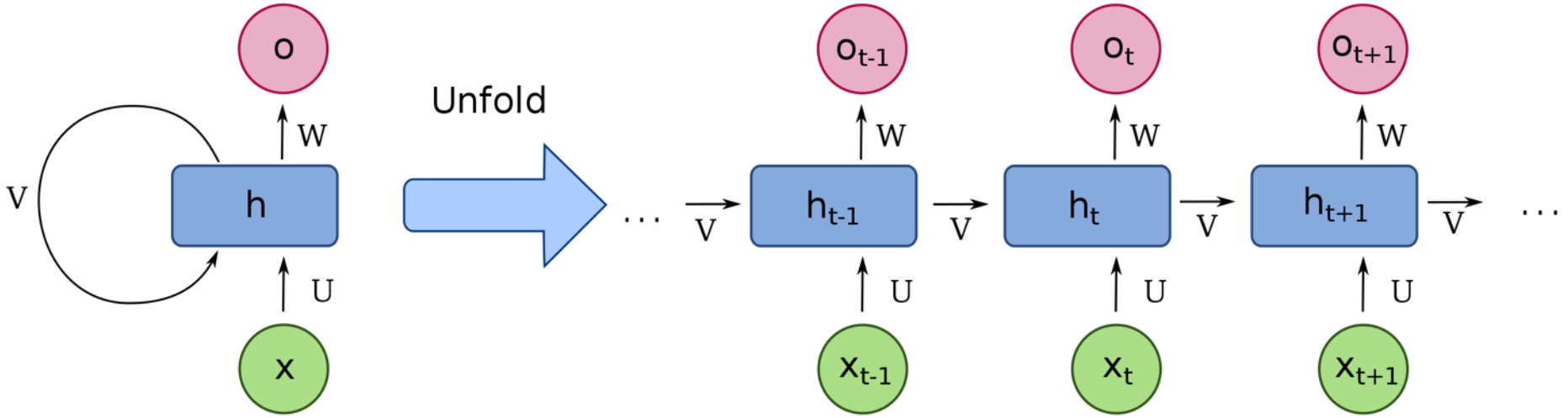
- Fixed-window Neural Model (Bengio, et al. 2000-2003):
  - Fixed window problem

** Need a model for any length **



Recurrent Neural Networks (RNN)

- A class of neural networks that are suited to process time-series and other sequential data.
- It stores the information from the previous experiences to be used for future prediction.
- Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRUs) are two common RNN architectures.



By fdeloche - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=60109157

RNN Language Model

- The input sequence can be of any size
- In theory, it can hold info from a long time in the past
- Model size doesn't depend on the input size
- The weights are shared (symmetry)

output distribution

$$\hat{\boldsymbol{y}}^{(t)} = \text{softmax}\left(\boldsymbol{U}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2\right) \in \mathbb{R}^{|V|}$$

hidden states

$$\boldsymbol{h}^{(t)} = \sigma\left(\boldsymbol{W}_h\boldsymbol{h}^{(t-1)} + \boldsymbol{W}_e\boldsymbol{e}^{(t)} + \boldsymbol{b}_1\right)$$

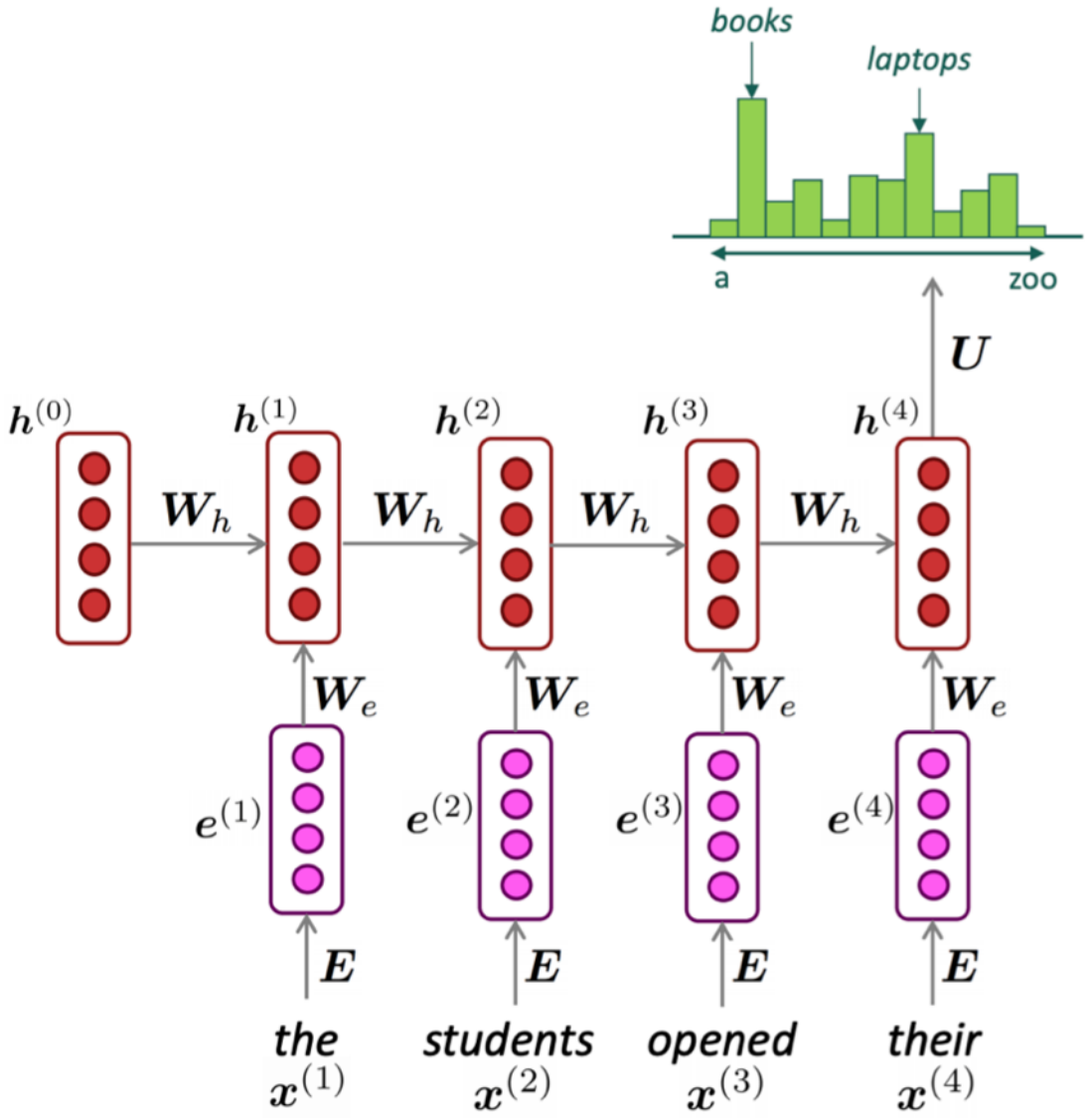
$\boldsymbol{h}^{(0)}$ is the initial hidden state

word embeddings

$$\boldsymbol{e}^{(t)} = \boldsymbol{E}\boldsymbol{x}^{(t)}$$

words / one-hot vectors

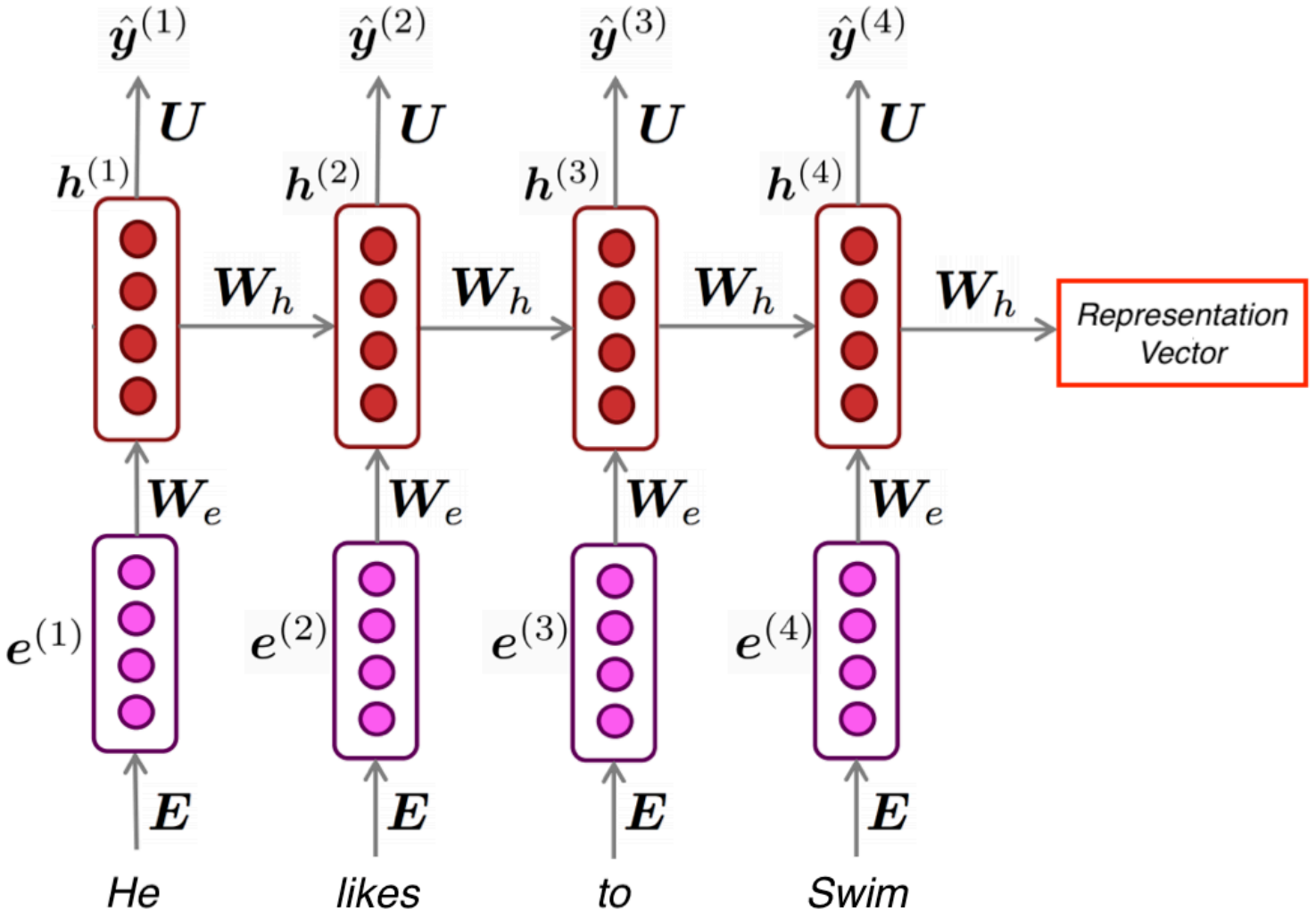
$$\boldsymbol{x}^{(t)} \in \mathbb{R}^{|V|}$$



Source: cs224n slides Stanford

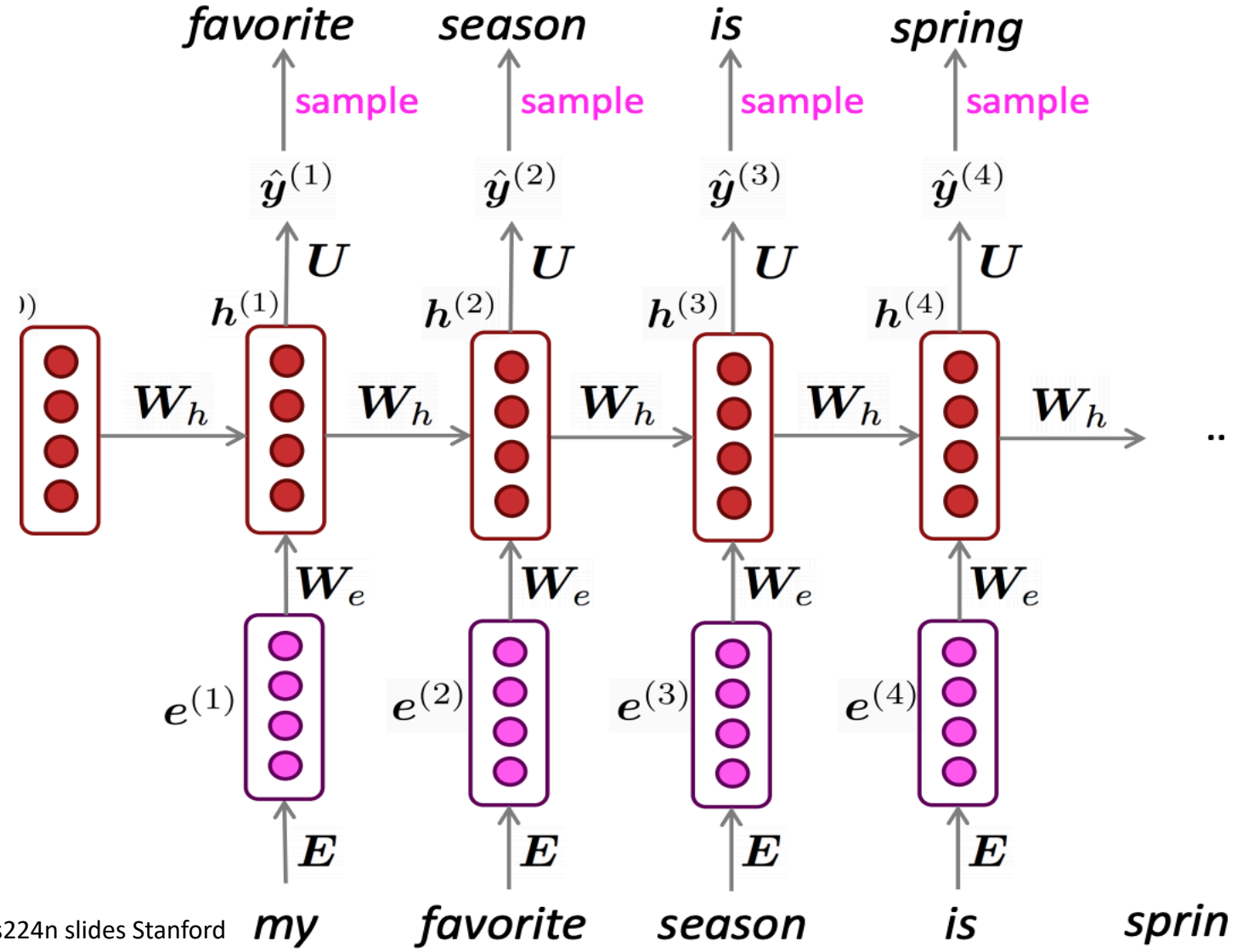
Text Representation using RNN

- Run the entire text through a RNN sequentially (word by word)
- Ignore the outputs
- Use the last state vector as the text representation



Text Generation using RNN

- Starting with an initial state (seed)
- The output of each step will be the input of the next step
- Continue until the output is ”End of Text”.



Source: cs224n slides Stanford

Text Generation Example

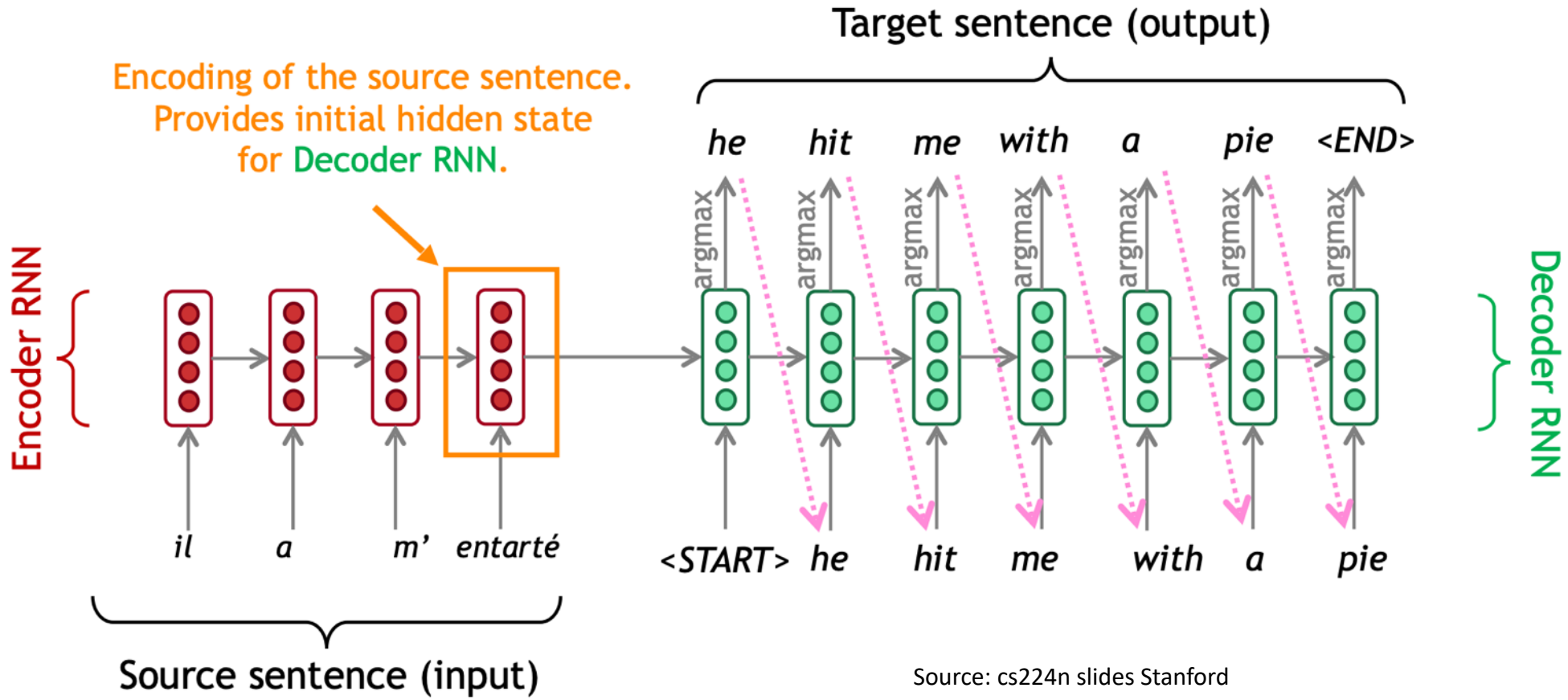
OBAMA-RNN. (source: https://medium.com/@samim/obama-rnn-machine-generated-political-speeches-c8abd18a2ea0)

SEED: Jobs

"The United States will step up to the cost of a new challenges of the American people that will share the fact that we created the problem. They were attacked and so that they have to say that all the task of the final days of war that I will not be able to get this done. The promise of the men and women who were still going to take out the fact that the American people have fought to make sure that they have to be able to protect our part. It was a chance to stand together to completely look for the commitment to borrow from the American people. And the fact is the men and women in uniform and the millions of our country with the law system that we should be a strong stretcks of the forces that we can afford to increase our spirit of the American people and the leadership of our country who are on the Internet of American lives."

Neural Machine Translation (NMT)

- Machine translation using sequence-to-sequence (a.k.a. seq2seq) architecture using two RNNs (encoder and decoder)



Source: cs224n slides Stanford

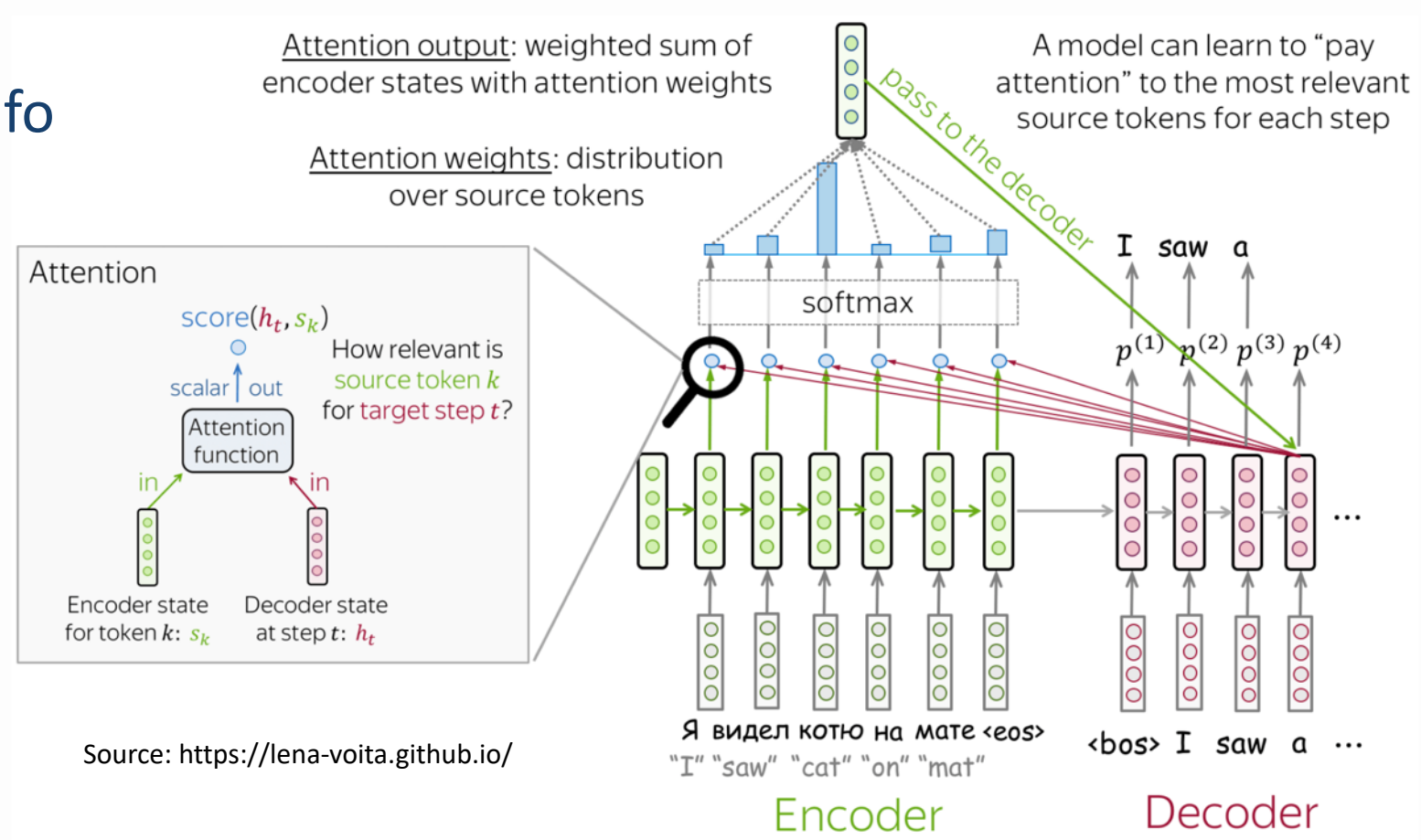
Seq2seq beyond NMT

- NMT is one of the biggest success stories of Modern NLP
- Google Translate switched to NMT in 2016
- There have been many improvements to the original seq2seq NMT system

- The same model (encoder-decoder) is used in many other applications:
  - Dialog system or Chatbot (previous utterance -> next utterance)
  - Image captioning (image representation -> caption)
  - Text summarization (long text -> short text)
  - Code generation (natural language -> programming code)
  - Sentence parsing (input text → output parse as sequence)
  - And many others

Seq2seq with Attention

- Problem with regular seq2seq model:
  - The encoder needs to capture all information about the source sentence.
  - Hard for the encoder
  - Decoder needs relevant info for each step
- Attention:
  - Significantly improves NMT performance.
  - Provides some interpretability
  - is a general and common Deep Learning technique

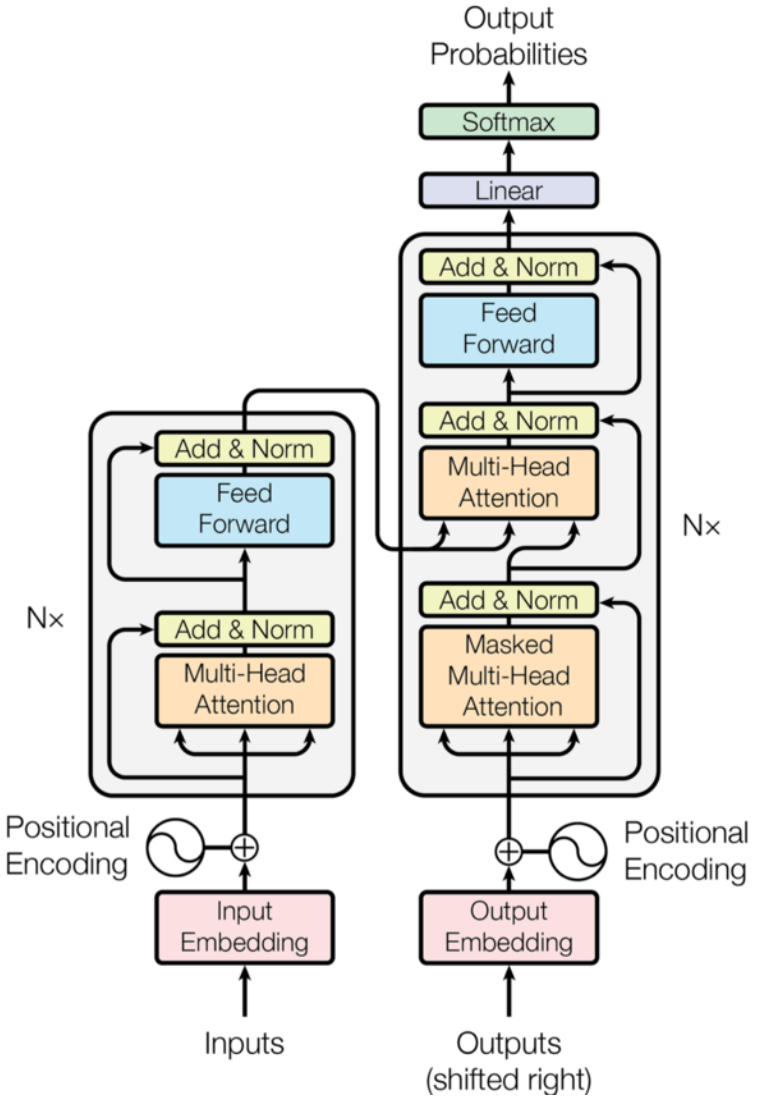


Source: https://lena-voita.github.io/

Transformers (Vaswani et al. 2017)

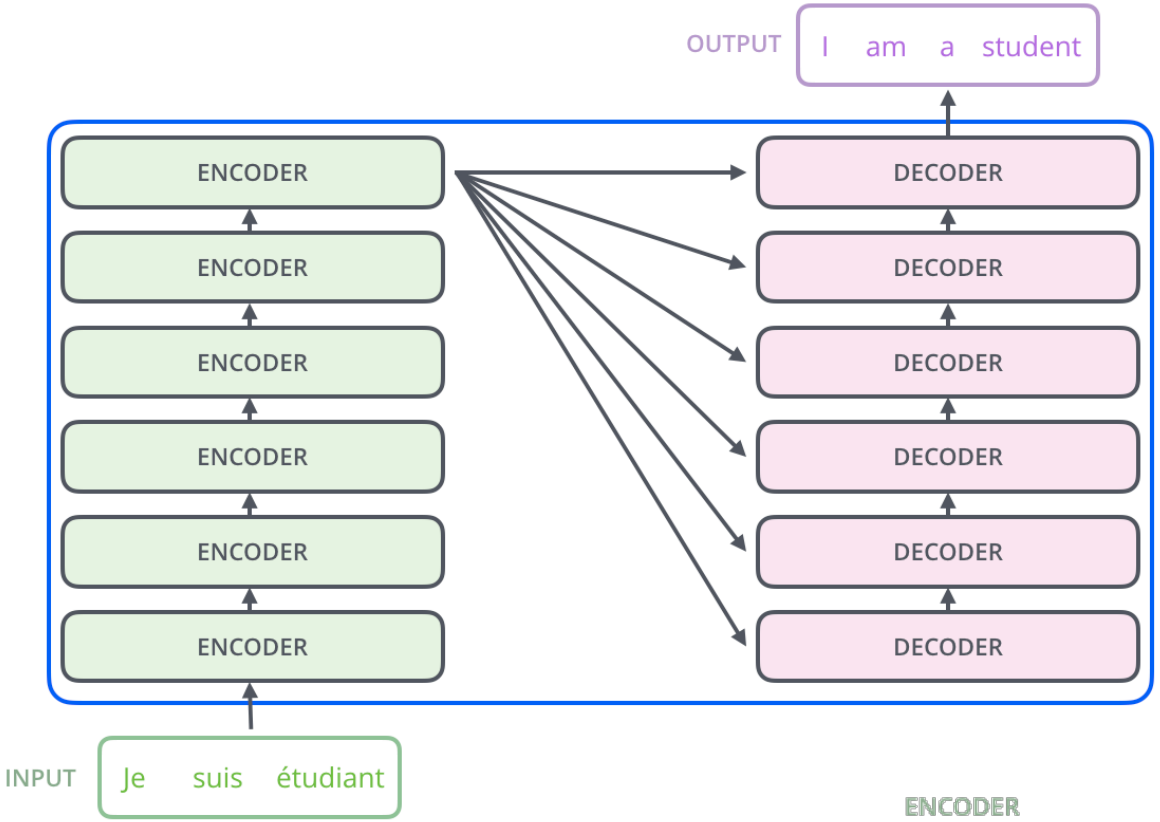
- We successfully used RNNs for both encoder and decoder in seq2seq models. But RNNs are slow to train because they are inherently sequential. We need parallelization.
- Also, RNNs (even LSTMs) become inefficient when there is a long gap between related information.
- In addition, RNNs generally need attention mechanism to deal with long range dependencies.
- Perhaps, "All we need is attention" not RNNs.

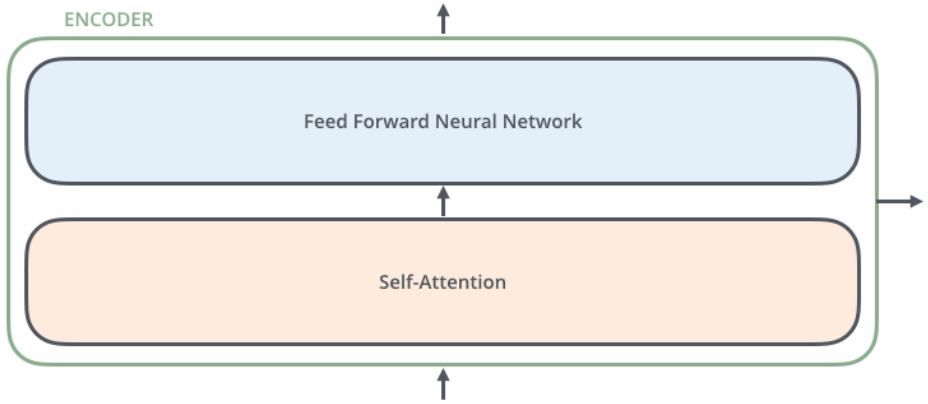
- **Transformer**: a non-recurrent encoder-decoder model fully based on attention



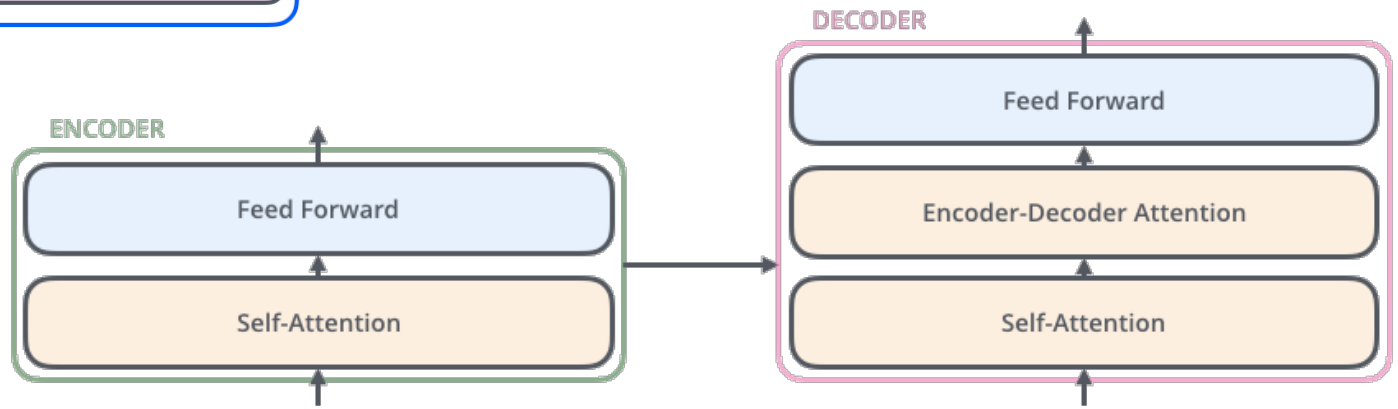
Source: cs224n slides Stanford

Transformer Architecture





- Feed forward parts are independent for each word and can be parallelized.



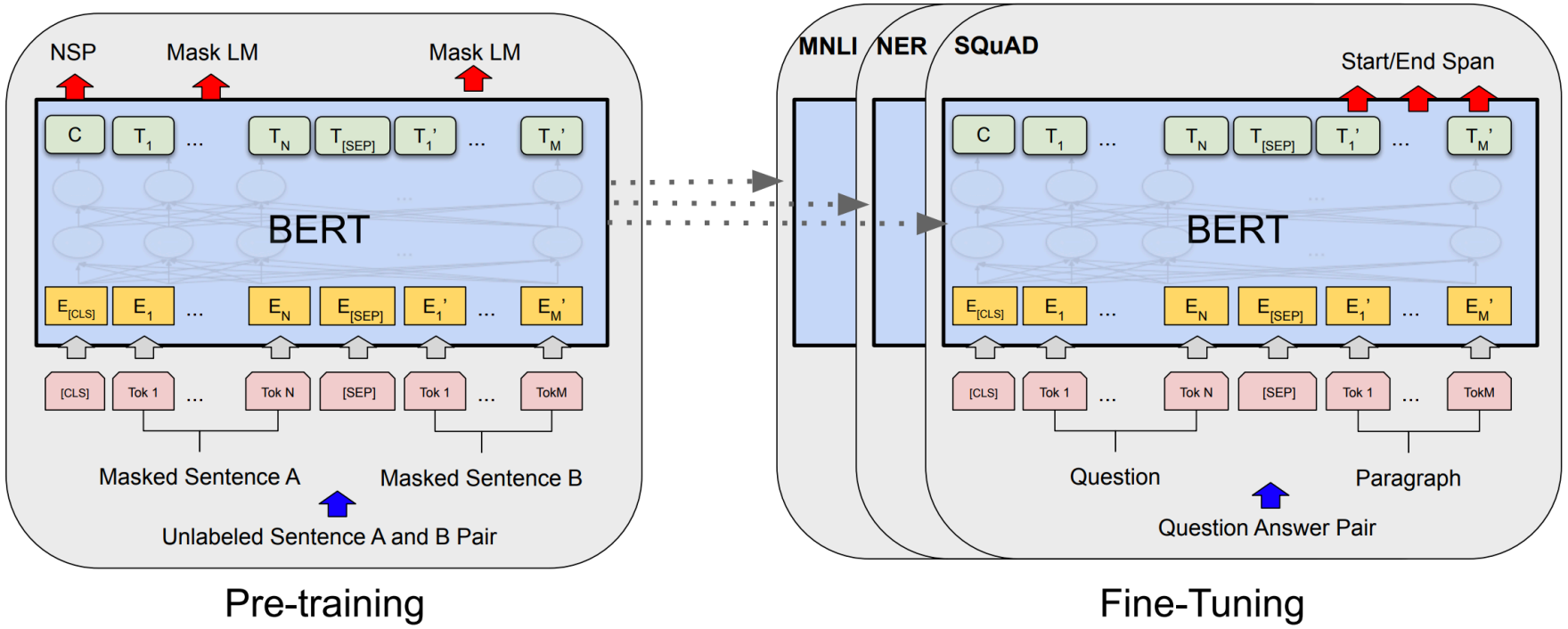
Source: http://jalammar.github.io/illustrated-transformer/

BERT: Pre-training of Deep Bidirectional Transformers (Google AI 2018)

- A novel Transformer-approach, pre-trained on large corpora and open-sourced that can be fine-tuned for any specific NLP task to get high accuracy.
- One of the biggest challenges in NLP is the lack of enough training data. Pre-trained language representation models like BERT are the key to get good results on small task-specific datasets.
- BERT was trained on Wikipedia and Book Corpus, a dataset containing +10,000 books of different genres.
- BERT in a contextual model (word representation is based on the other words in the sentence) and bidirectional. Example: “bank account” vs. “bank of the river”.
- The best part about BERT is that it can be download and used for free.

BERT Variants

- Lots of models with slightly different pre-training objectives, but very similar architecture as BERT, have been trained to achieve the state-of-the-art results on many NLP tasks.



Source: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language

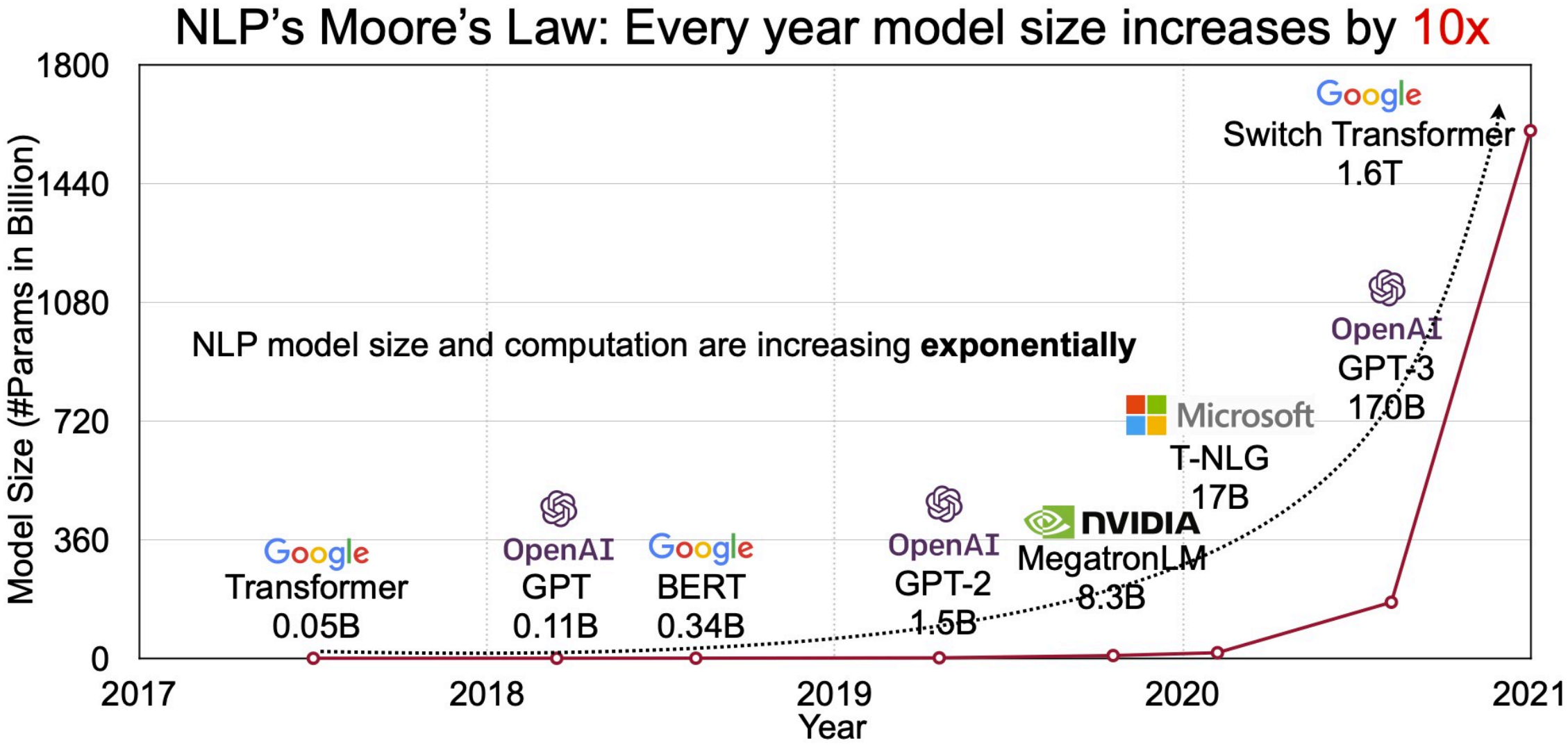
Large Language Models (LLMs)

The last few years saw the rise of Large Language Models (LLMs):

- The original **Transformer** breaks previous performance records for machine translation.
- **BERT** popularizes the pre-training then fine tuning process, as well as Transformer-based contextualized word embeddings. It then rapidly starts to power Google Search and Bing Search.
- **GPT-2** demonstrates the machine's ability to write as well as humans do.
- First **T5**, then **T0** push the boundaries of transfer learning (training a model on one task, and then having it do well on other adjacent tasks) and posing a lot of different tasks as text-to-text tasks.
- **GPT-3** showed that massive scaling of generative models can lead to shocking emergent applications (the industry continues to train larger models like Gopher, MT-NLG…etc).

Source:https://jalammar.github.io/illustrated-retrieval-transformer/

Pre-trained Large Language Models



Source:https://hanlab.mit.edu/projects/efficientnlp_old/

ChatGPT: Optimizing Language Models for Dialogue

We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and reject inappropriate requests. ChatGPT is a sibling model to InstructGPT, which is trained to follow an instruction in a prompt and provide a detailed response.

OpenAI

How Can We Use ChatGPT?

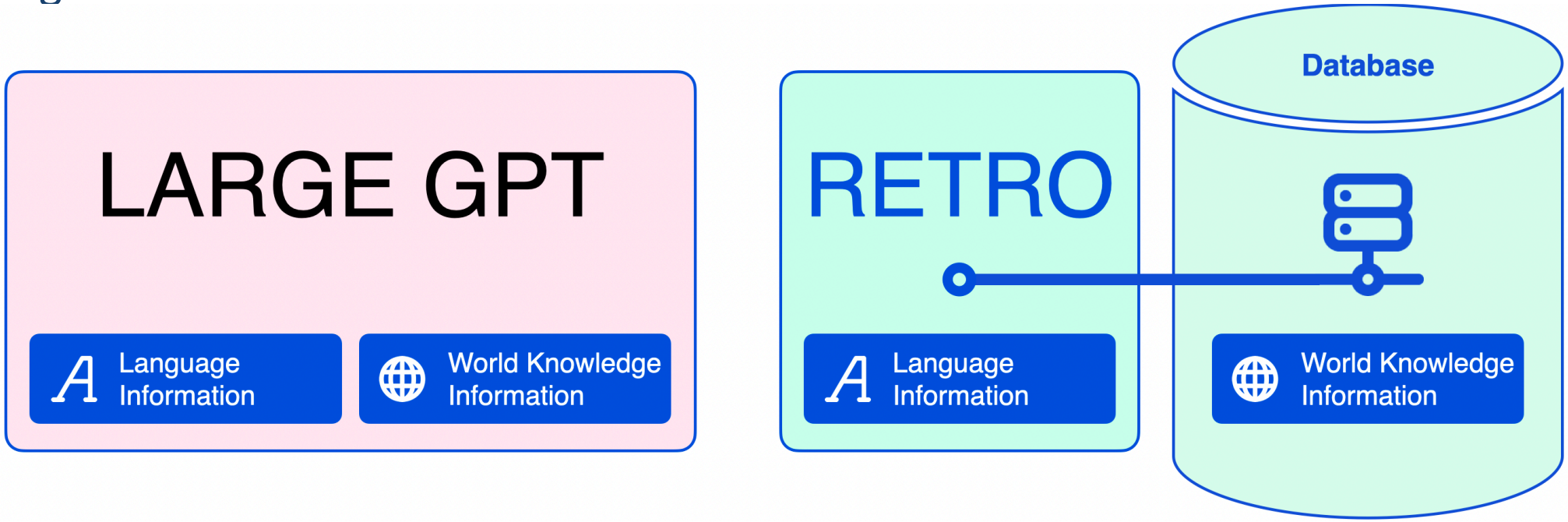
SA

Can my colleague use ChatGPT to generate legitimate content?

It is possible for your colleague to use ChatGPT to generate content, but it is important to note that ChatGPT is a language model and is not capable of generating original content on its own. Instead, it can only generate text based on the input it receives from a user. This means that your colleague would need to provide the necessary input and context for ChatGPT to generate meaningful text. Additionally, it is worth mentioning that the output generated by ChatGPT may not always be accurate or reliable, so it is important to carefully review any text generated by the model before using it for any important or official purposes.

Retrieval-Enhanced Transformer (RETRO) - DeepMind

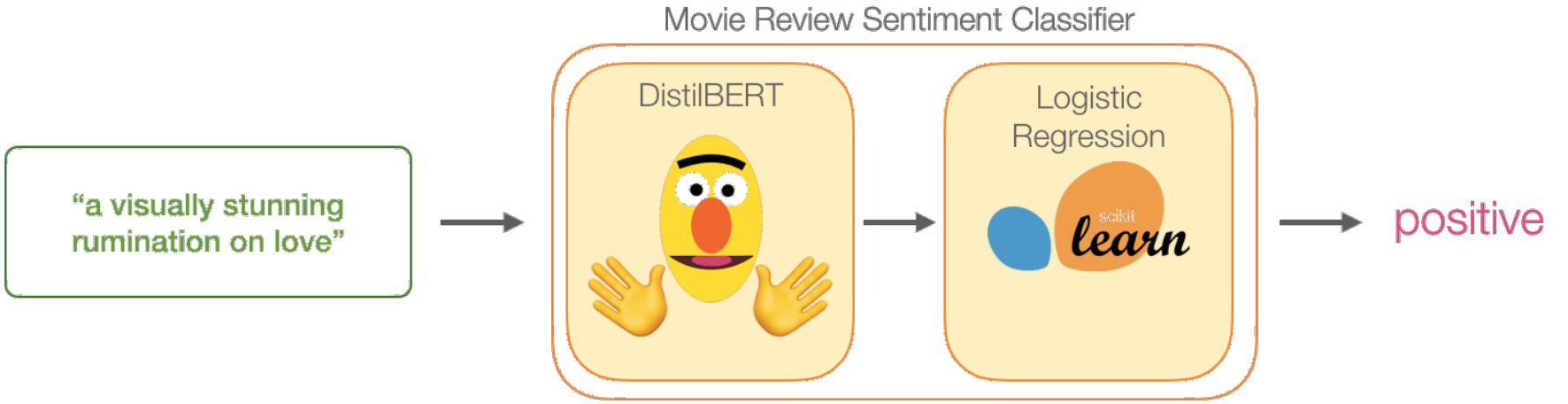
- Large Language Models (LLMS) encode everything they know in their model parameters. While this makes sense for language information, it is inefficient for factual and world-knowledge information.
- By including a retrieval method in the language model, the model can be much smaller. A neural database aids it with retrieving factual information it needs during text generation.



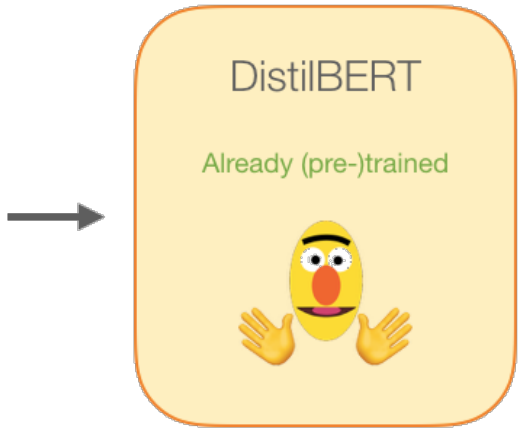
Source:https://jalammar.github.io/illustrated-retrieval-transformer/

Two Example Applications

Text Classification with BERT



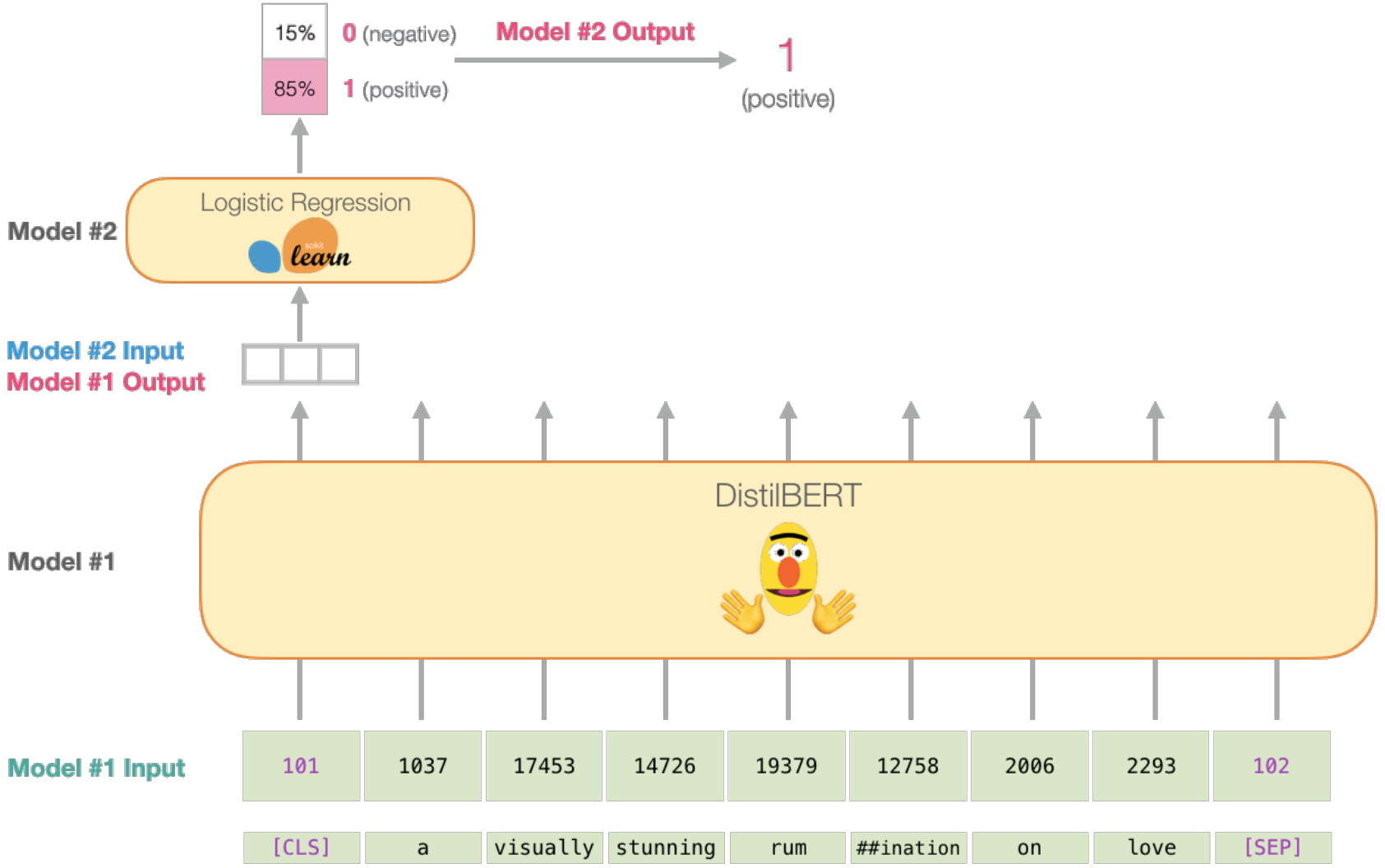
	Sentence	label
0	a stirring , funny and finally transporting re imagining of beauty and the beast and 1930s	1
1	apparently reassembled from the cutting room floor of any given daytime soap	0
…	…	…
1,999	the movie is undone by a filmmaking methodology that 's just experimental enough	1



	Sentence Embeddings				label
	0	1	…	767	
0	-0.215	-0.1402	…	0.201	1
1	-0.172	-0.144	…	0.371	0
…	…	…	…	…	…
1,999	0.124	0.014	…	0.274	1

Source: http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

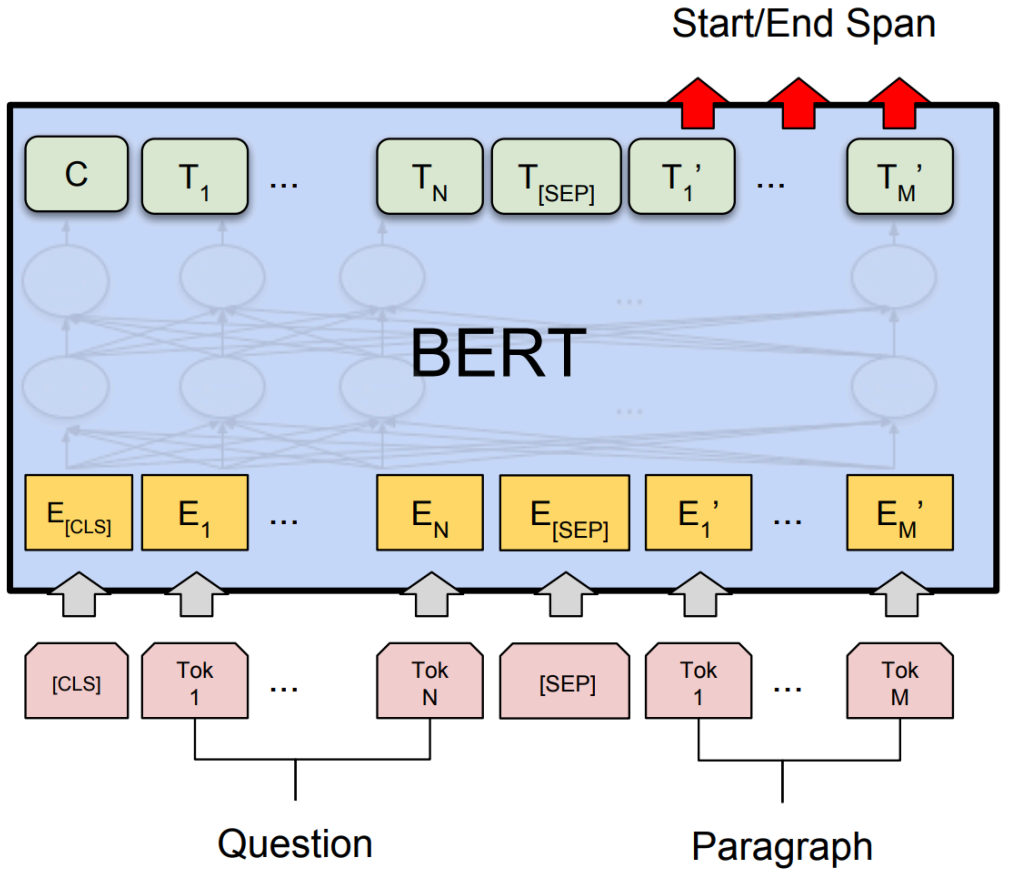
Text Classification with BERT



Source: http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/

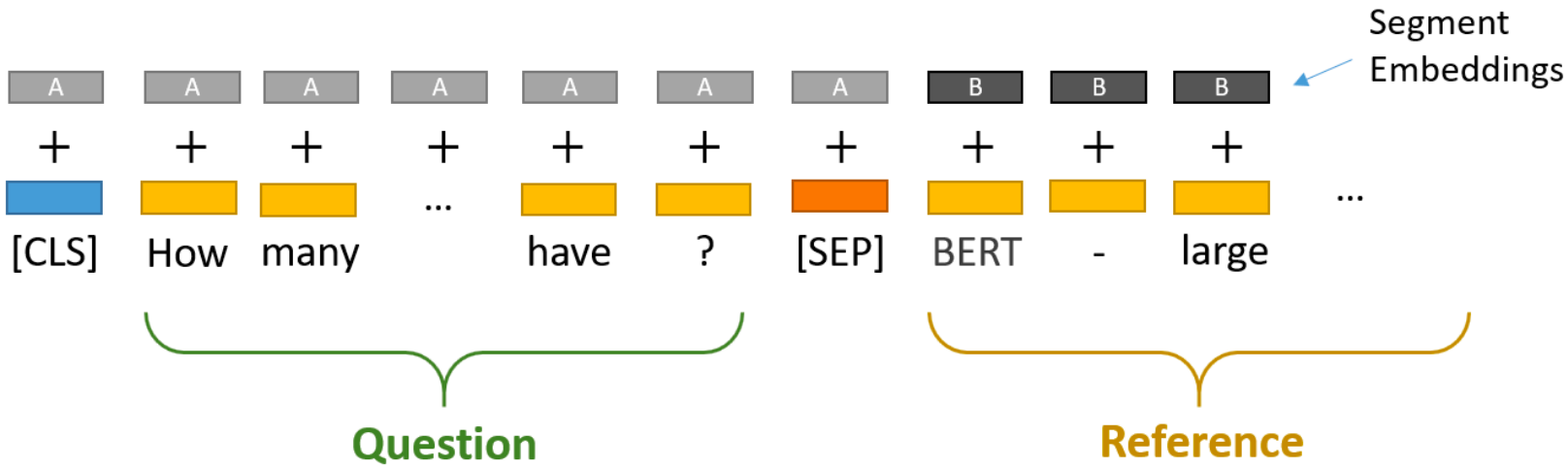
Fine tuning BERT for Q&A

- Train on **S**tanford **Qu**estion and **A**nswering **D**ataset (SQuAD), which consists of questions posed by crowdworkers on a set of Wikipedia articles.
- Given a question, and a passage of text containing the answer, BERT needs to highlight the “span” (start and end) of text corresponding to the correct answer.



Source: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, Google AI Language

BERT Input Format (Segment Embeddings)

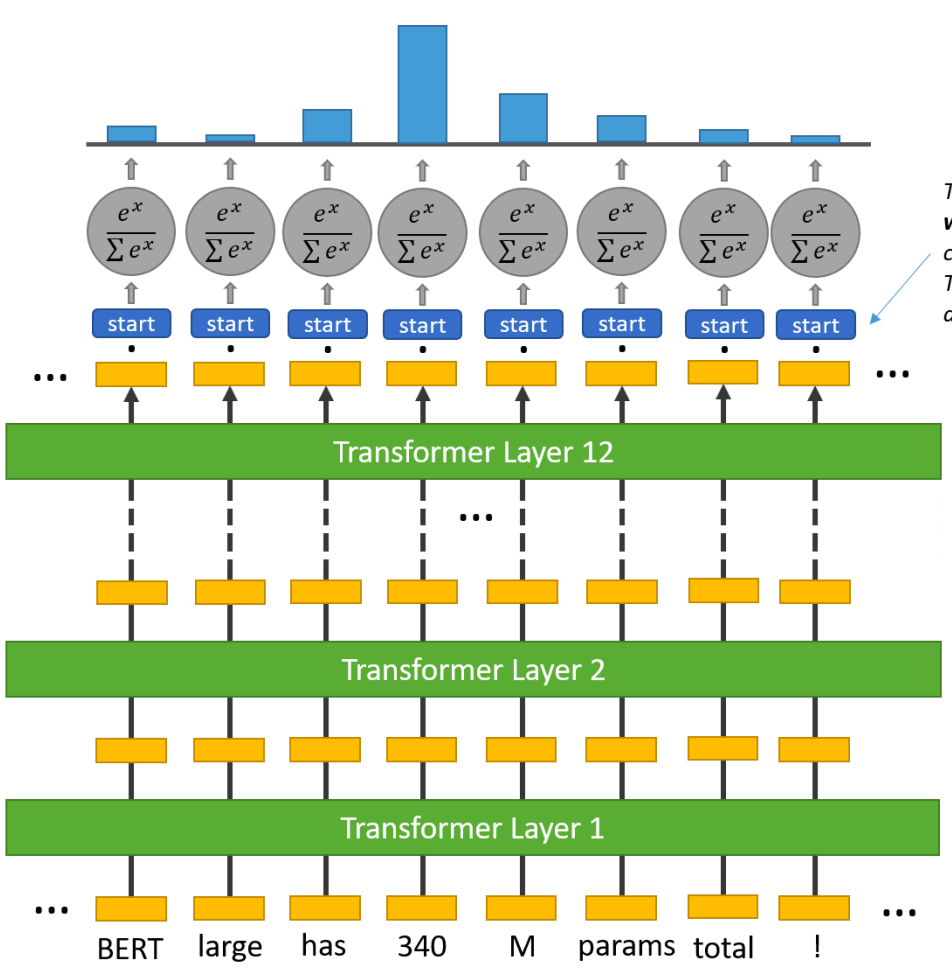


Question: How many parameters does BERT-large have?

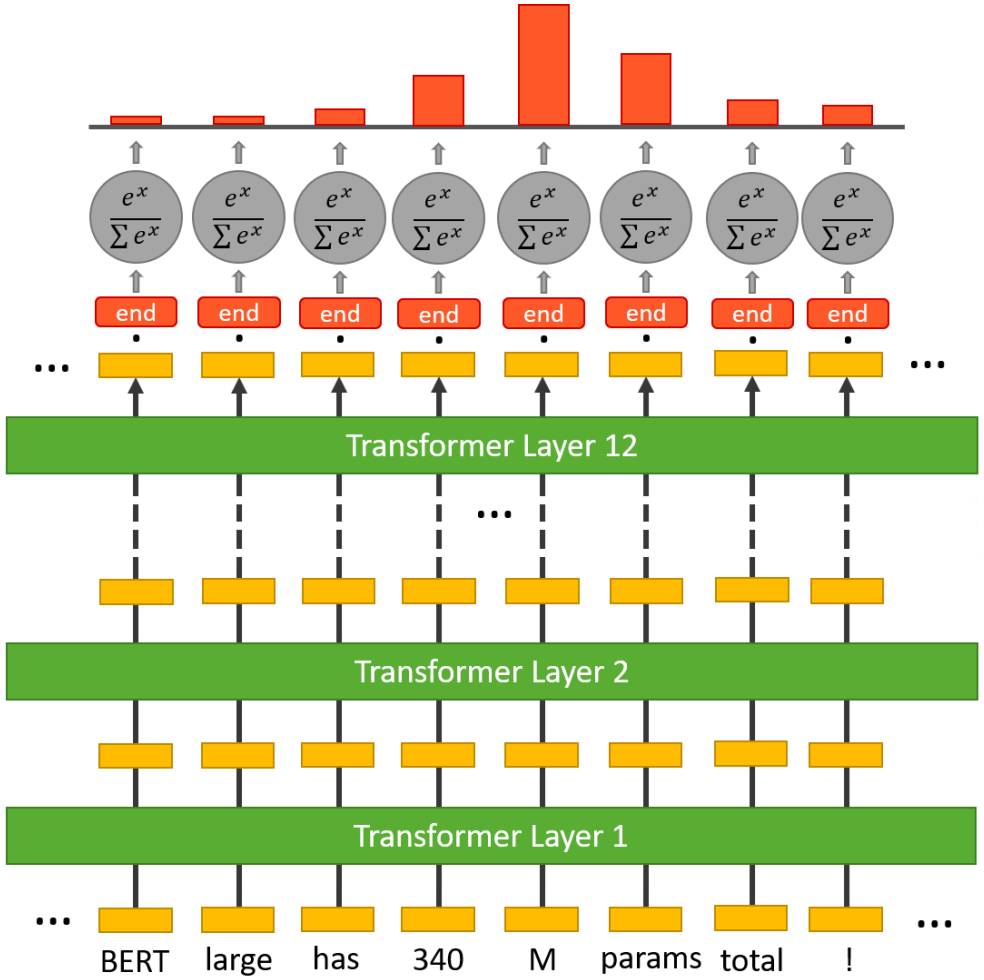
Reference Text: BERT-large is really big... it has 24 layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

Source: https://mccormickml.com/2020/03/10/question-answering-with-a-fine-tuned-BERT/

Start & End Token Classifiers



*This length 768 vector is the **weights** for the start token classifier.*
*The **same weights** are applied to **every position**.*



Source: https://mccormickml.com/2020/03/10/question-answering-with-a-fine-tuned-BERT/

VECTOR INSTITUTE