UNSW
THE UNIVERSITY OF NEW SOUTH WALES

The Rise of Language Models

Never Stand Still | Faculty of Engineering | COMP9444 7b

Sonit Singh

School of Computer Science and Engineering
Faculty of Engineering
The University of New South Wales, Sydney, Australia
sonit.singh@unsw.edu.au

The Rise of Language Models

Photo Credit: Image by Free-photos from Pixabay. https://pixabay.com/photos/road-winding-street-bridge-1030789/

Motivation

- A robot wrote this entire article. Are you scared yet, human?
  https://www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3
- What It's Like To be a Computer: An Interview with GPT-3
  https://www.youtube.com/watch?v=PqbB07n_uQ4

Credit: American Liver Foundation, https://liverfoundation.org/

Natural Language Processing

- Enabling machines to process, represent, understand, and generate languages

In fact, the Chinese NORP market has the three CARDINAL most influential names of the retail and tech space – Alibaba GPE , Baidu ORG , and Tencent PERSON (collectively touted as BAT ORG), and is betting big in the global AI GPE in retail industry space . The three CARDINAL giants which are claimed to have a cut-throat competition with the U.S. GPE (in terms of resources and capital) are positioning themselves to become the 'future AI PERSON platforms'. The trio is also expanding in other Asian NORP countries and investing heavily in the U.S. GPE based AI GPE startups to leverage the power of AI GPE . Backed by such powerful initiatives and presence of these conglomerates, the market in APAC AI is forecast to be the fastest-growing one CARDINAL , with an anticipated CAGR PERSON of 45% PERCENT over 2018 - 2024 DATE .

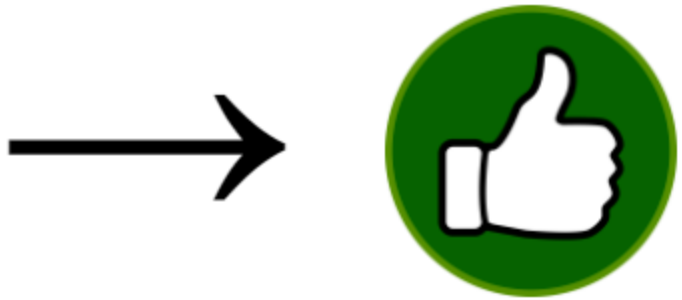
To further elaborate on the geographical trends, North America LOC has procured more than 50% PERCENT of the global share in 2017 DATE and has been leading the regional landscape of AI GPE in the retail market. The U.S. GPE has a significant credit in the regional trends with over 65% PERCENT of investments (including M&As, private equity, and venture capital) in artificial intelligence technology. Additionally, the region is a huge hub for startups in tandem with the presence of tech titans, such as Google ORG , IBM ORG , and Microsoft ORG .

Source: Hugging Face. https://twitter.com/huggingface/status/1230870653194121216

NLP Applications

- Text Classification

"I love this movie.
I've seen it many times
and it's still awesome."

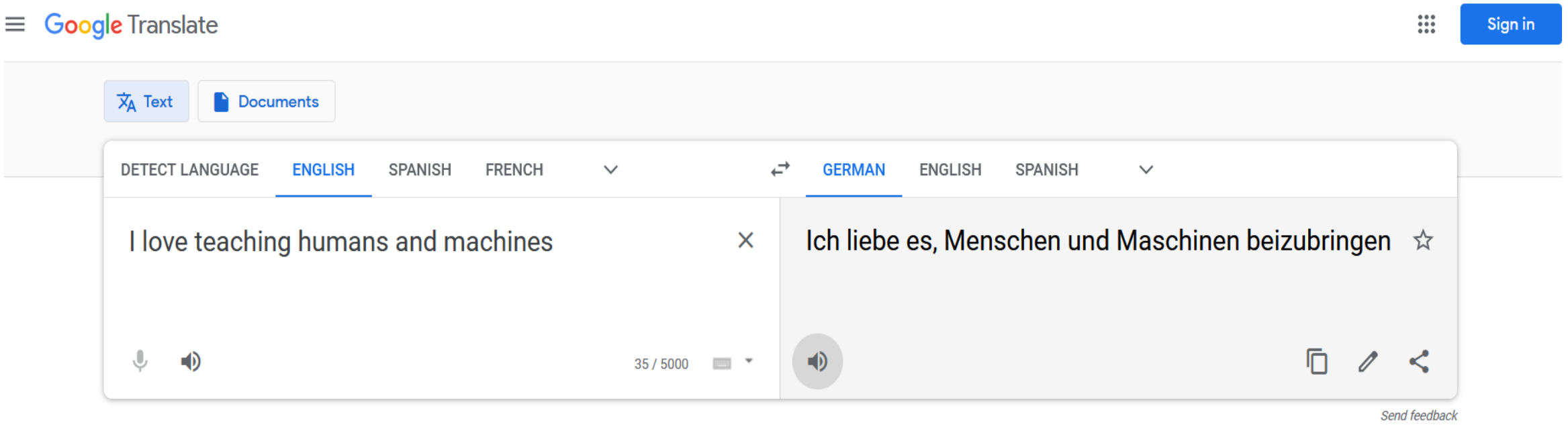


"This movie is bad.
I don't like it it all.
It's terrible."

Image Credit: Sentiment Analysis. https://colab.research.google.com/github/shangeth/Google-ML-Academy/blob/master/2-Deep-Neural-Networks/2_9_ANN_Natural_Language_Processing.ipynb

NLP Applications

- Machine Translation



NLP Applications

- Question Answering/Comprehension

Passage Sentence

In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity.

Question

What causes precipitation to fall?

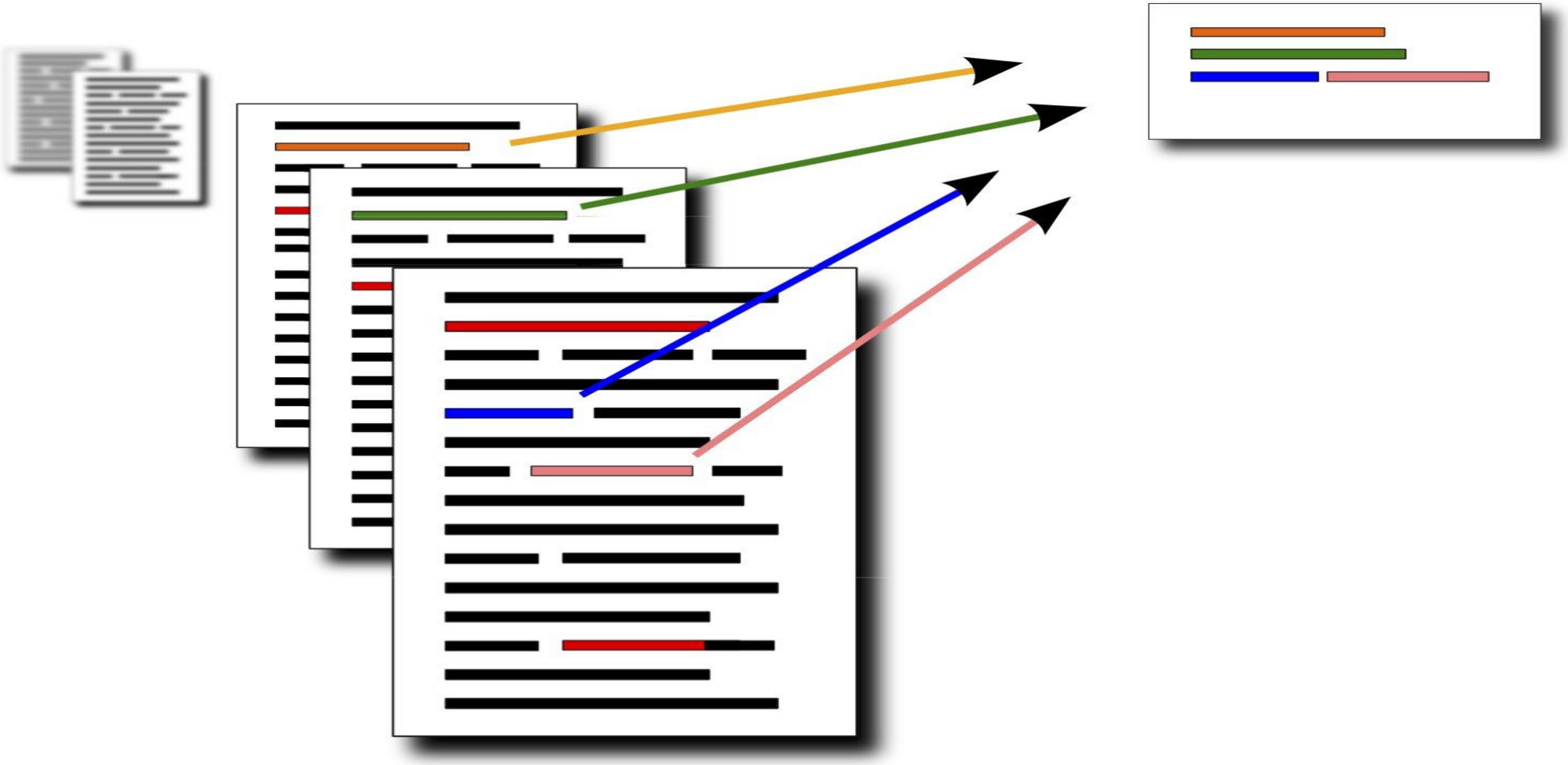
Answer Candidate

gravity

Source: The Stanford Question Answering Dataset (SQUAD). https://rajpurkar.github.io/mlx/qa-and-squad/

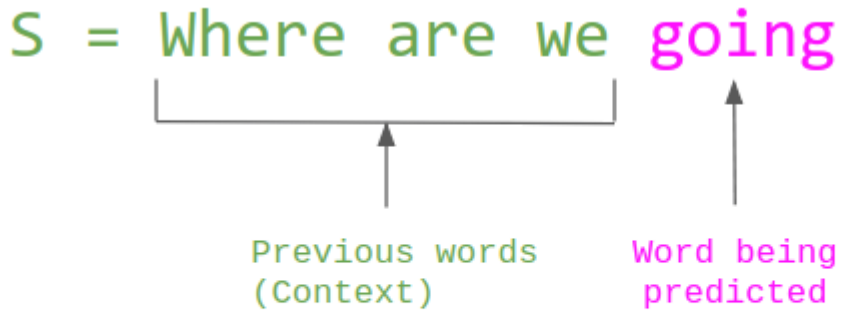
NLP Applications

- Text Summarization



Language Modelling

- A probability distribution over words or word sequences
- Probability distribution over strings of text
  - How likely is a string in a given “language”?

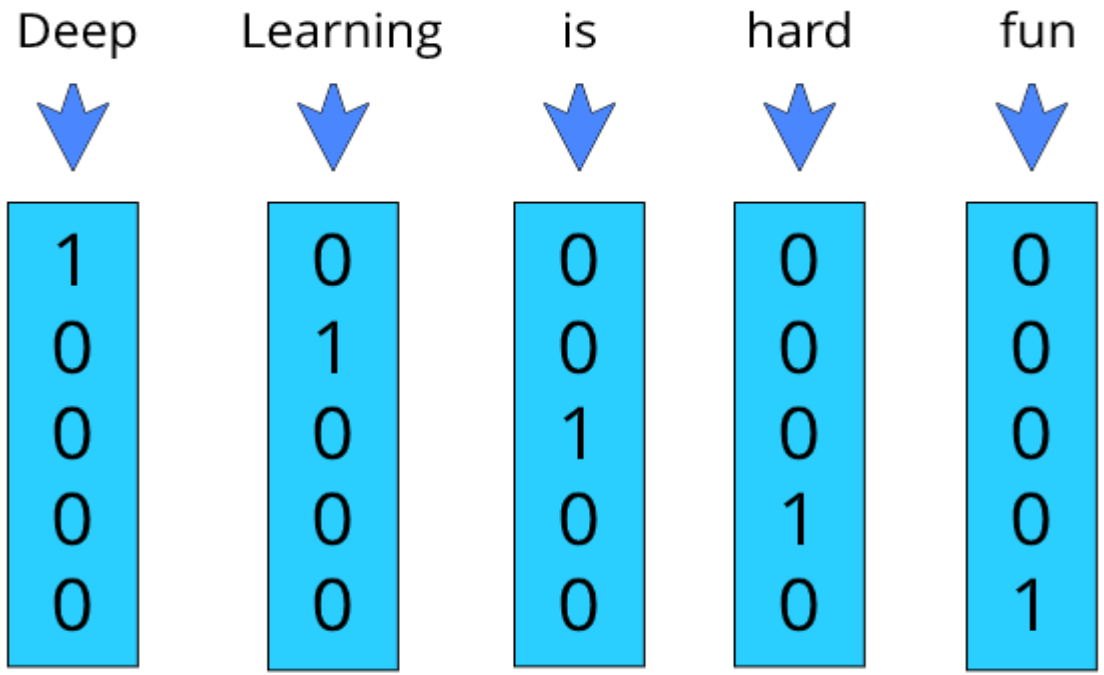


$$P(S) = P(\text{Where}) \times P(\text{are} \mid \text{Where}) \times P(\text{we} \mid \text{Where are}) \times P(\text{going} \mid \text{Where are we})$$

Image Credit: The Gradient. https://thegradient.pub/understanding-evaluation-metrics-for-language-models/

Representing text: One-hot Encoding

- One-hot vector of an ID is a vector filled with 0s, except for a 1 at the position associated with the ID.
  - For e.g., for vocabulary size D=10, the one-hot vector of word (w) having ID=4 is
    e(w) = [0 0 0 1 0 0 0 0 0 0]



- One-hot encoding makes no assumption abou
  word similarity
  - All words are equally similar/different from eac

- This is a natural representation to start with,
  though a poor one

Representing text: Count Vectorizer

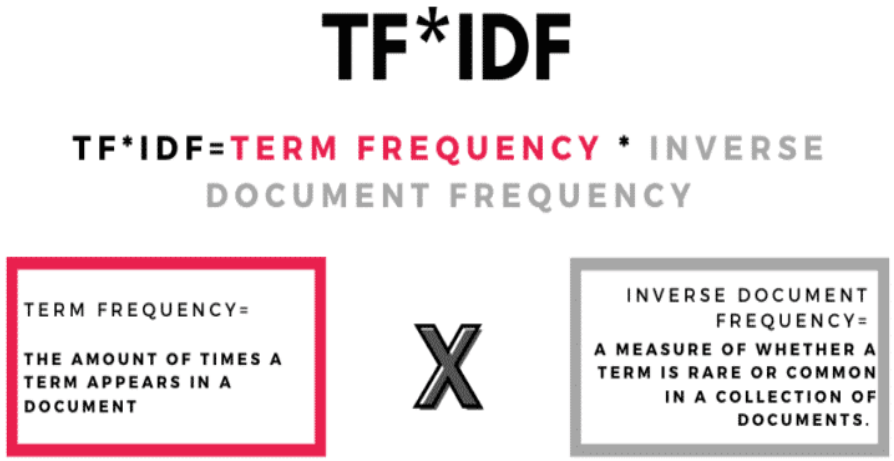
- Counts the number of occurrences of each word appearing in a document
- Converts a collection of text documents to a vector of term/token counts

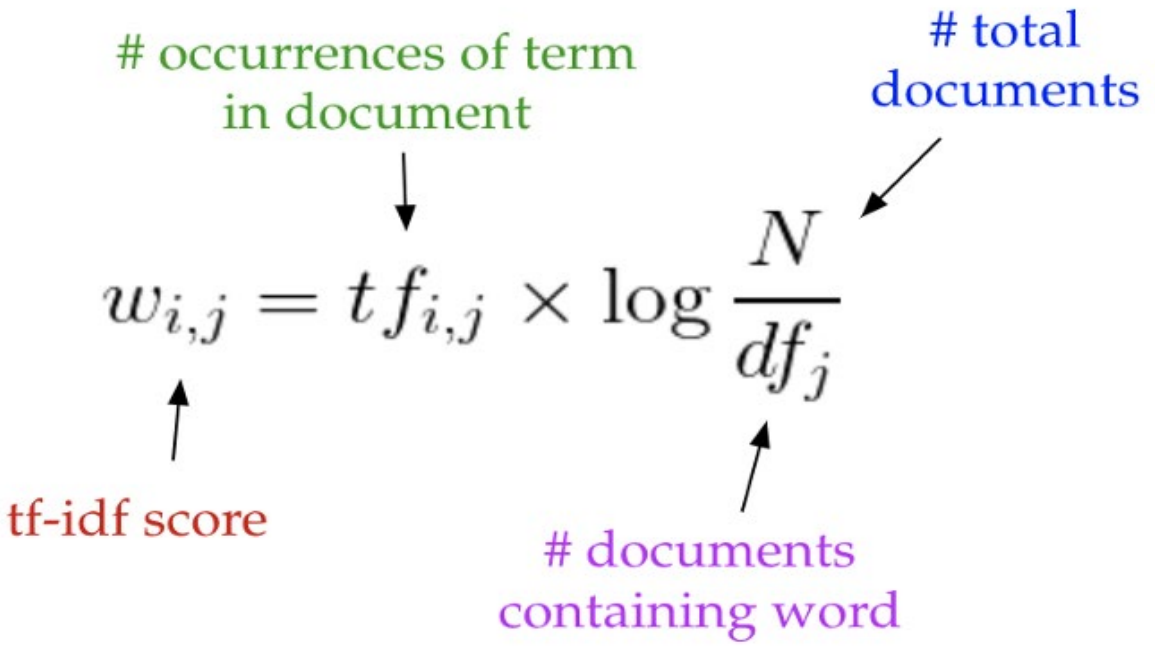
	the	red	dog	cat	eats	food
1. the red dog →	1	1	1	0	0	0
2. cat eats dog →	0	0	1	1	1	0
3. dog eats food →	0	0	1	0	1	1
4. red cat eats →	0	1	0	1	1	0

Image Credit: Ishan Kotian. Fake news classification – NLP @Kaggle. https://www.kaggle.com/lykin22/fake-news-classification-nlp

Representing text: tf-idf vectorizer

- Stands for Term Frequency – Inverse Document Frequency (tf-idf)
- Common algorithm to transform text into a meaningful representation of numbers





- This cheesecake is really **yummy**. I'm going for another slice.
- The restaurant service was very slow.

Representing text as vectors

- One-hot encoding, count and tf-idf vectors are **sparse** and **long**
  - Most values are zeros
  - Length of vector is equal to size of vocabulary (can be more than 50,000)

- Alternative is to look for **dense vectors.** These are
  - short and dense
  - easier to use a features in machine learning (less weights to tune)
  - can capture context

Word Embeddings

- Instead of counting how often each word w occurs near “university”, train a classifier on a binary prediction task:
  - Is w likely to show up near “university”?

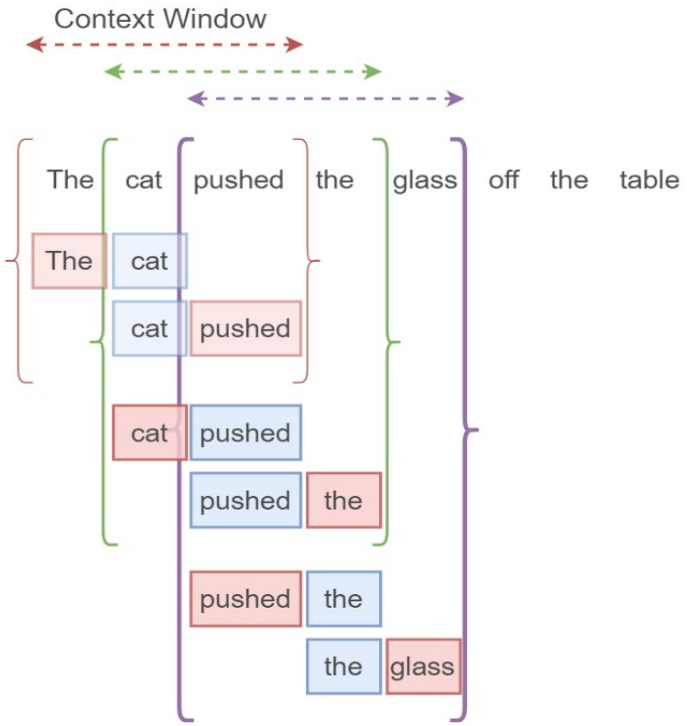
- We don’t actually care about this task, but we take the learned classifier weights as the word embeddings
- Uses free-form text as implicitly supervised training data
  - A word s near “university” acts as gold standard “ground-truth” to the question
  - No need for hand-labeled supervision

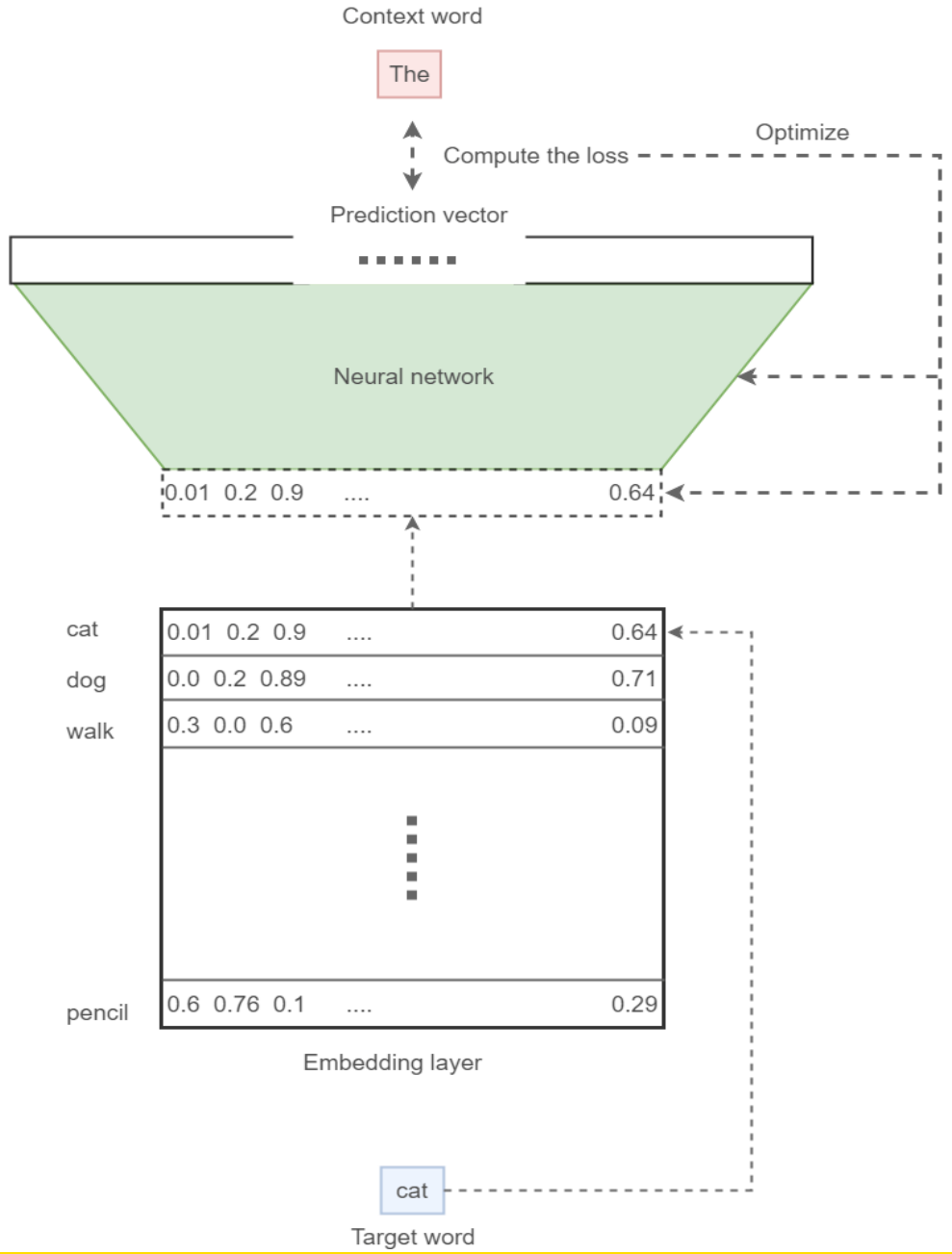
- Word vector algorithms use the context of the words to learn numerical representations for words, so that words used in the same context have similar looking word vectors.

Slide Credit: Dan Jurafsky and James Marting. Speech and Language Processing. Stanford University

Word2Vec

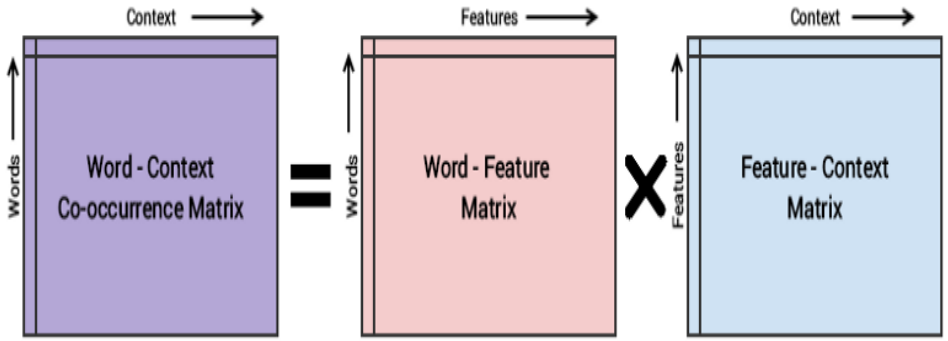
- "You should know a word by the company it keeps"
  - J.R. Firth
- Uses CBOW or Skip-gram model to train the network
- Word2Vec uses skip-gram neural network to predict neighbor context

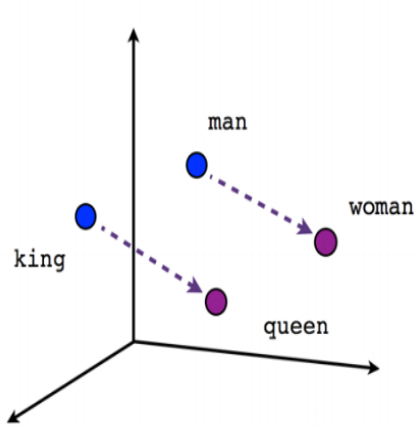




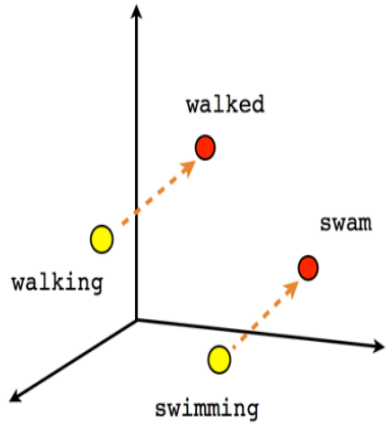
GloVe

- Word2Vec rely on local statistics (local context information of words)
- GloVe captures both local and global statistics (word co-occurrence) to obtain word vectors
- Can derive semantic relationships between words from the co-occurrence matrix
- Requires upfront pass-through entire dataset

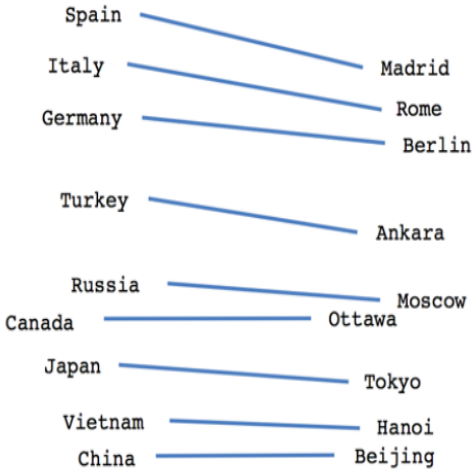




Male-Female



Verb tense



Country-Capital

Credit: GloVe: Global Vectors for Word Representation. https://nlp.stanford.edu/projects/glove/

Statistical Language Modeling

- Models that assign probabilities to the sequences of words
- Focus on n-gram
  - 2-gram (bigram). E.g., “going to”, “on the”
  - 3-gram (trigram). E.g., “what are you”, “please turn your”

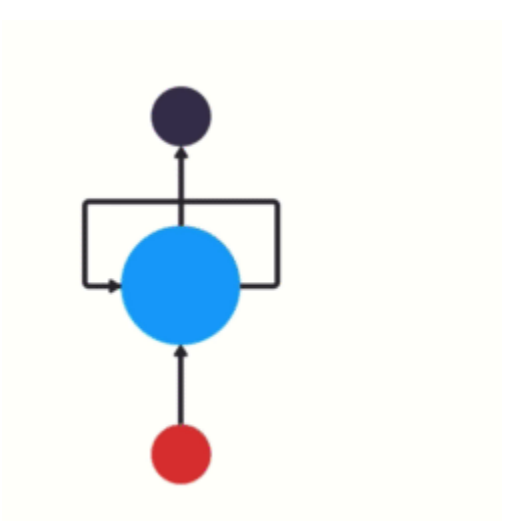
- Naïve approach:
  - Calculate the probability of word w, given history, h: P(w/h)
  - A computation challenge with significant size of corpus
  - Do approximation using chain rule

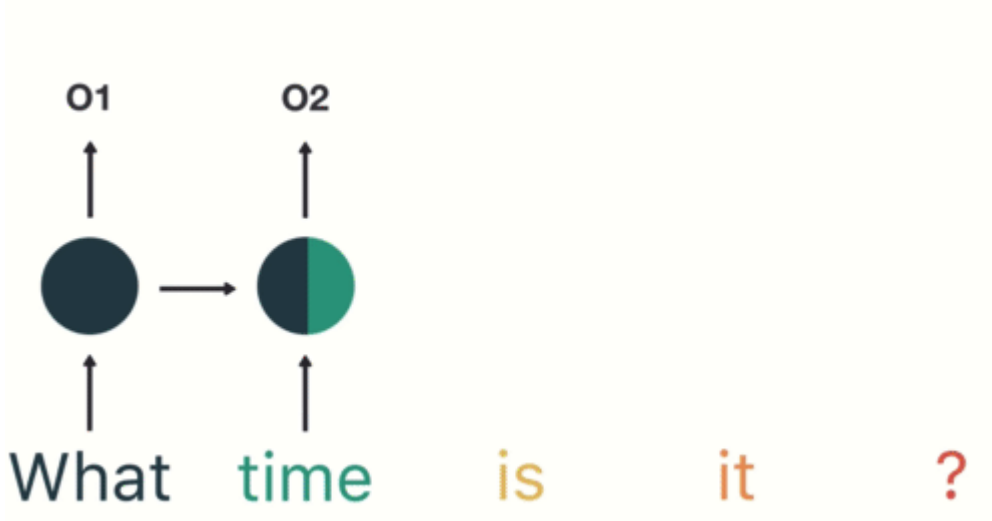
- Bigram language model
  - Approximates the probability of a word given all the previous words by using the conditional probability of one preceding word
  - Based on Markov assumption: we can predict the probability of some future unit without looking too far in the past
  - Use Maximum Likelihood Estimation (MLE) to estimate the probability function

Source: Language Models – N-gram https://towardsdatascience.com/introduction-to-language-models-n-gram-e323081503d9

Recurrent Neural Networks (RNNs)

- A class of artificial neural networks suitable for modeling sequential or time-series data
- Exhibit dynamic behavior where connections between nodes form a directed graph along a temporal sequence
- RNNs have a looping mechanism that acts as a highway to allow information to flow from one step to the next. This information is the hidden state, which is a representation of previous inputs.

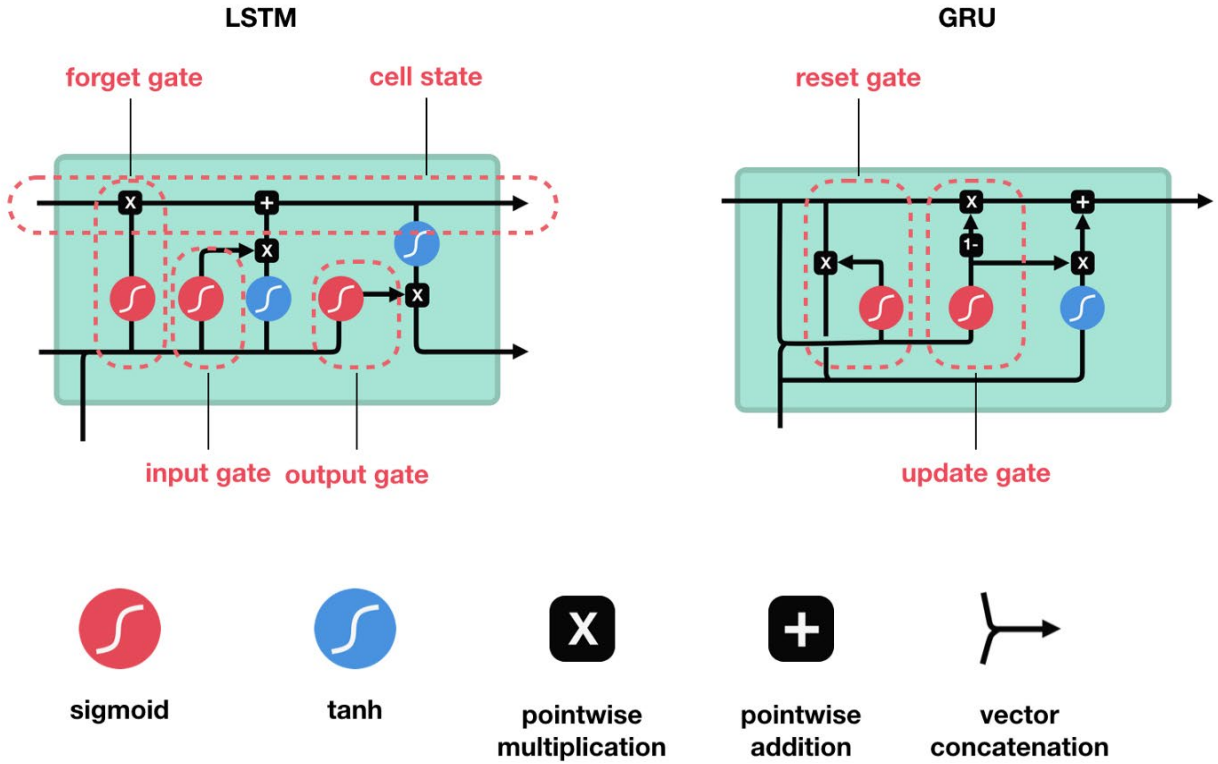




Credit: Michael Phi. Illustrated guide to RNNs. https://towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9

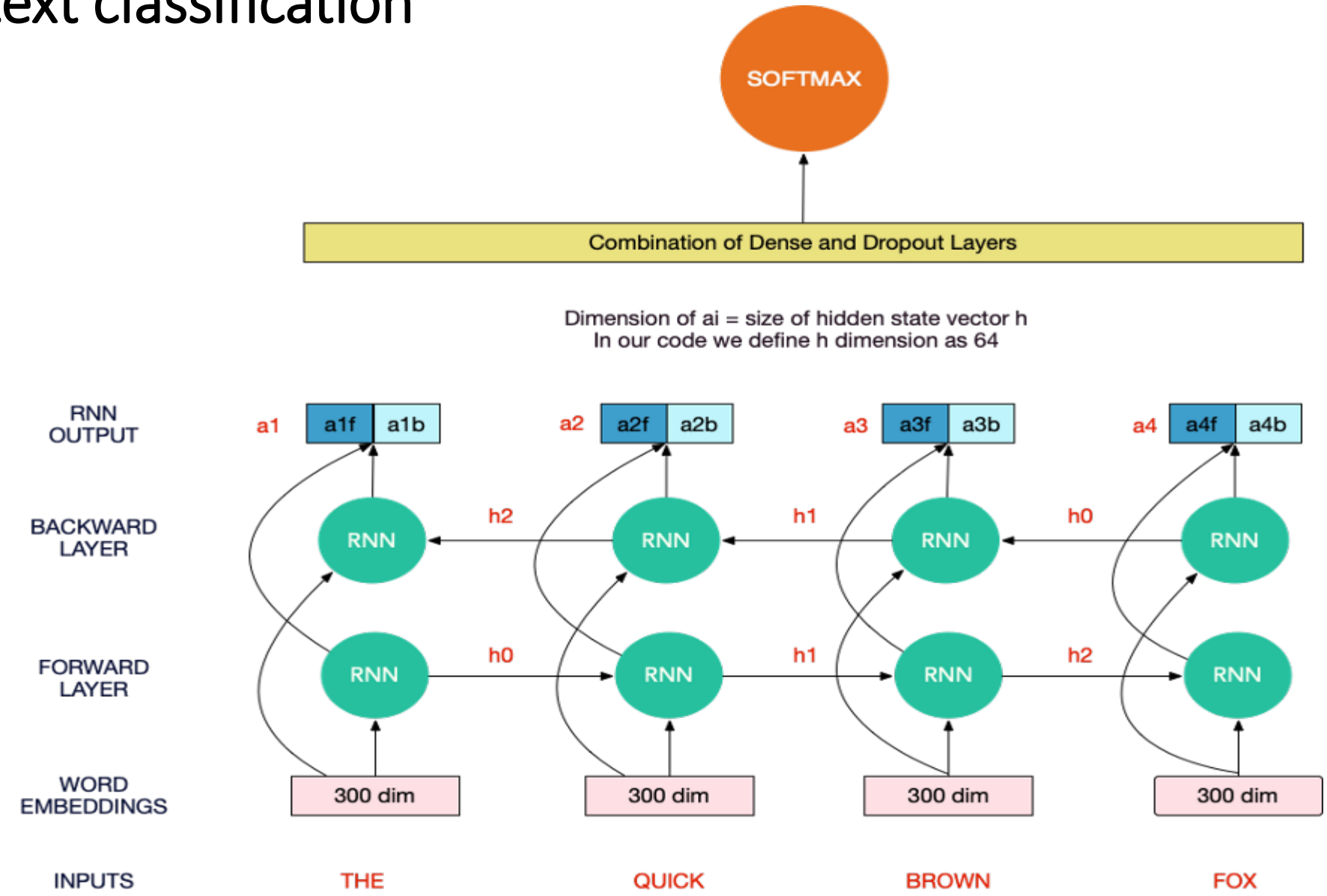
LSTMs/GRUs

- Vanilla RNNs suffers from vanishing gradient problem
  - As the RNNs processes more steps, it has troubles retaining information from previous steps.
  - Due to back-propagation, the earlier layers fail to do any learning as the internal weights are barely being adjusted due to extremely small gradients.
  - Does not learn the long-range dependencies across time steps
- LSTMs and GRUs are two special RNNs, capable of learning long-term dependencies using mechanisms called **gates**.
- These gates are different tensor operations that can learn what information to add or remove to the hidden state.



Credit: Michael Phi. Towards Data Science. https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

End-to-End text classification



Seq2Seq model

- Model has two parts: Encoder and Decoder (Encoder-Decoder Framework)

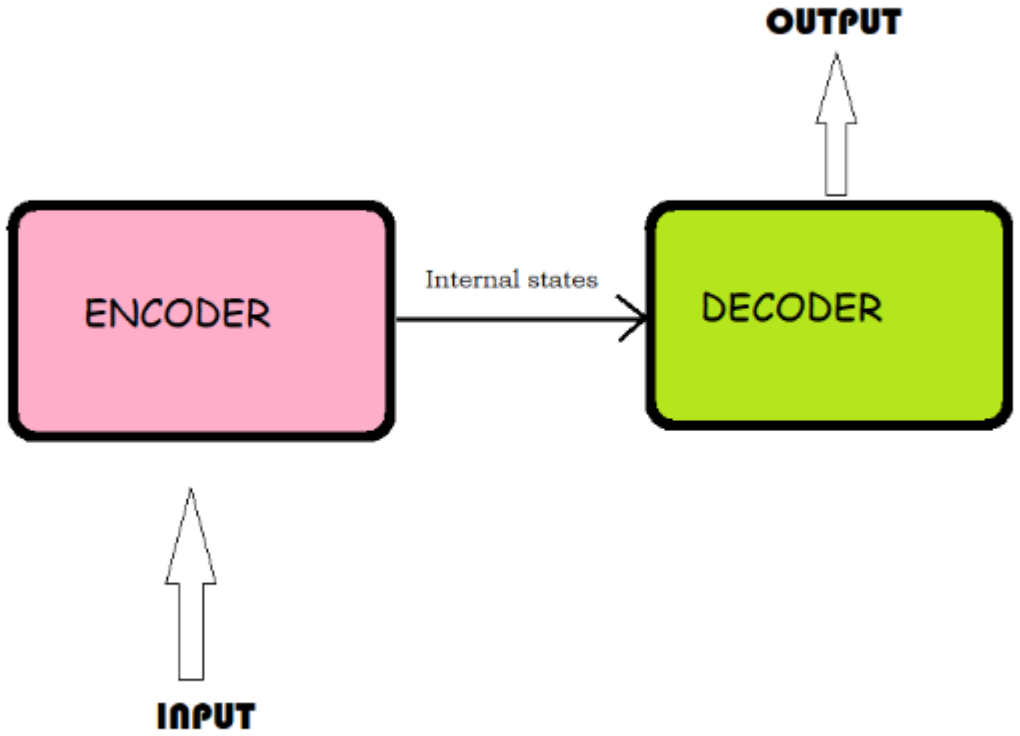
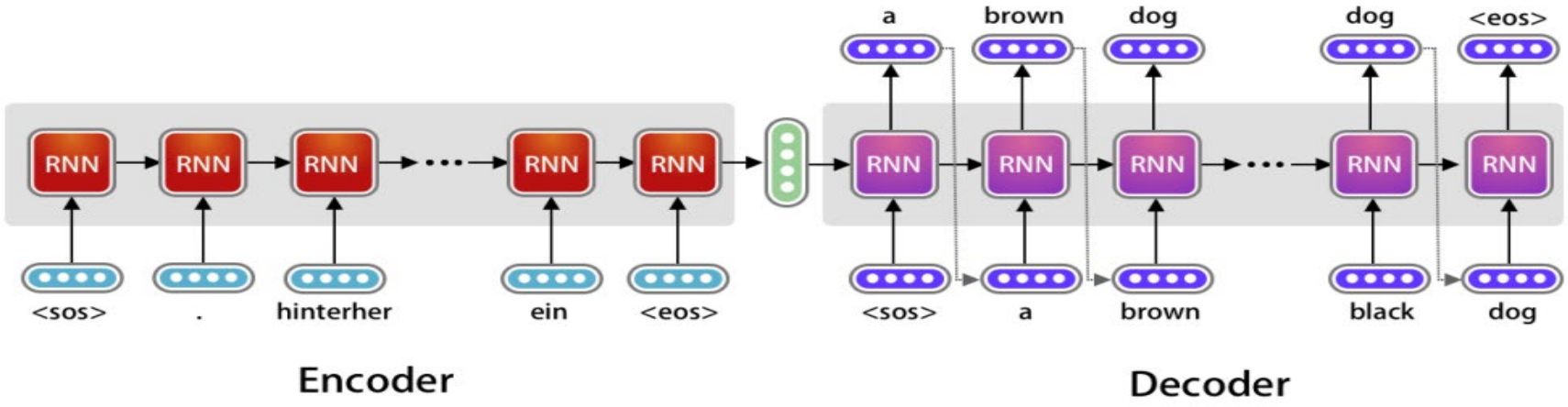


Image Credit: Machine Translation using Recurrent Neural Network and PyTorch. http://www.adeveloperdiary.com/data-science/deep-learning/nlp/machine-translation-recurrent-neural-network-pytorch/

Seq2Seq used for various NLP applications

- Machine Translation



- Automatic Email Reply

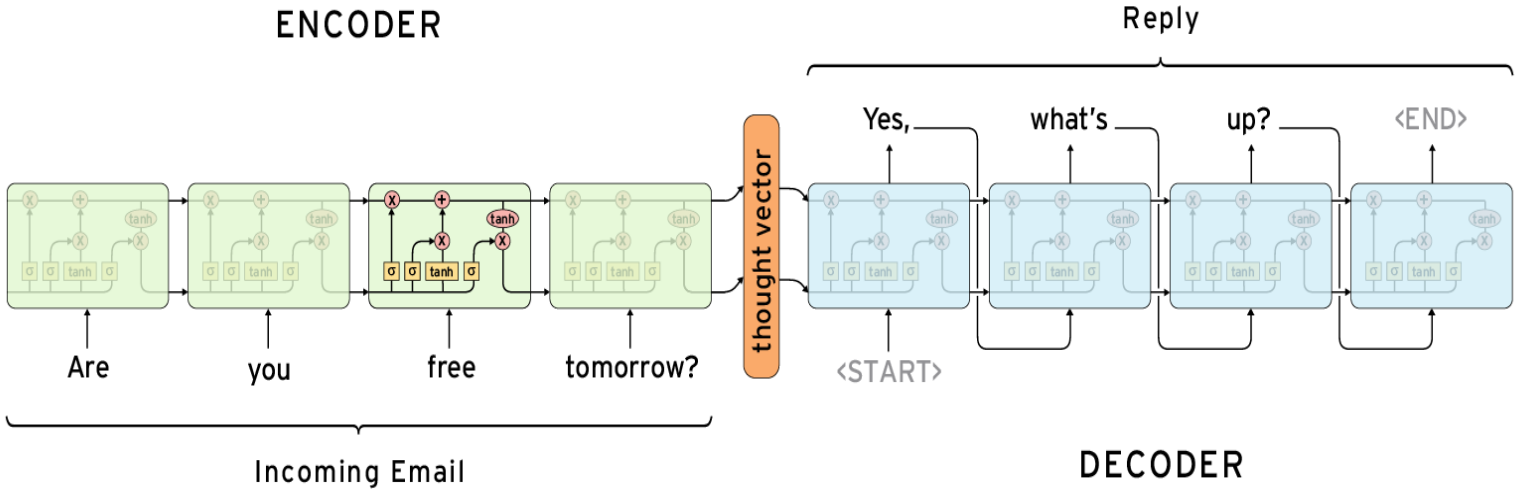


Image Credit: Machine Translation using Recurrent Neural Network and PyTorch. http://www.adeveloperdiary.com/data-science/deep-learning/nlp/machine-translation-recurrent-neural-network-pytorch/

Seq2Seq used for various NLP applications

- Google's Multilingual Neural Machine Translation

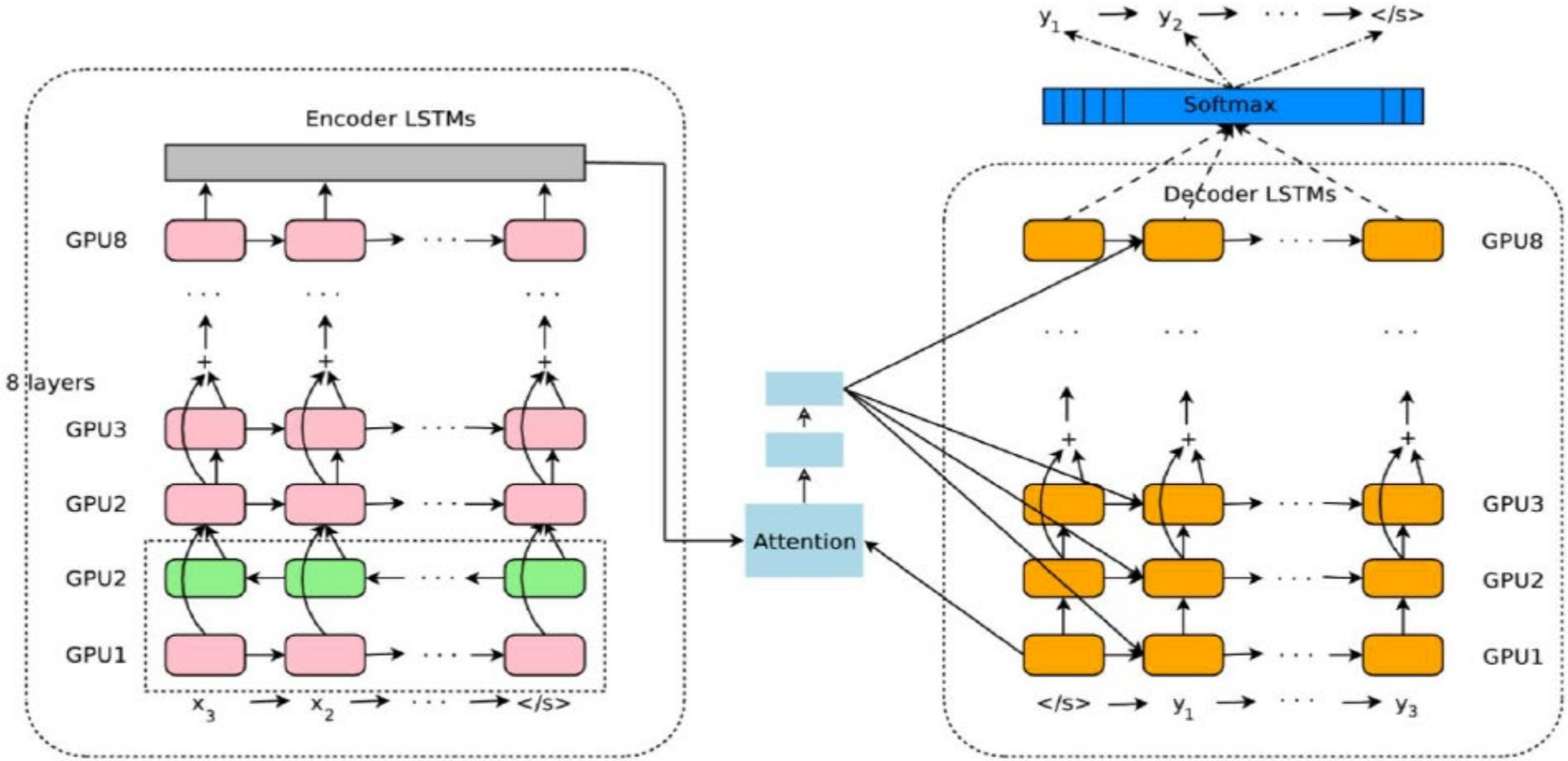
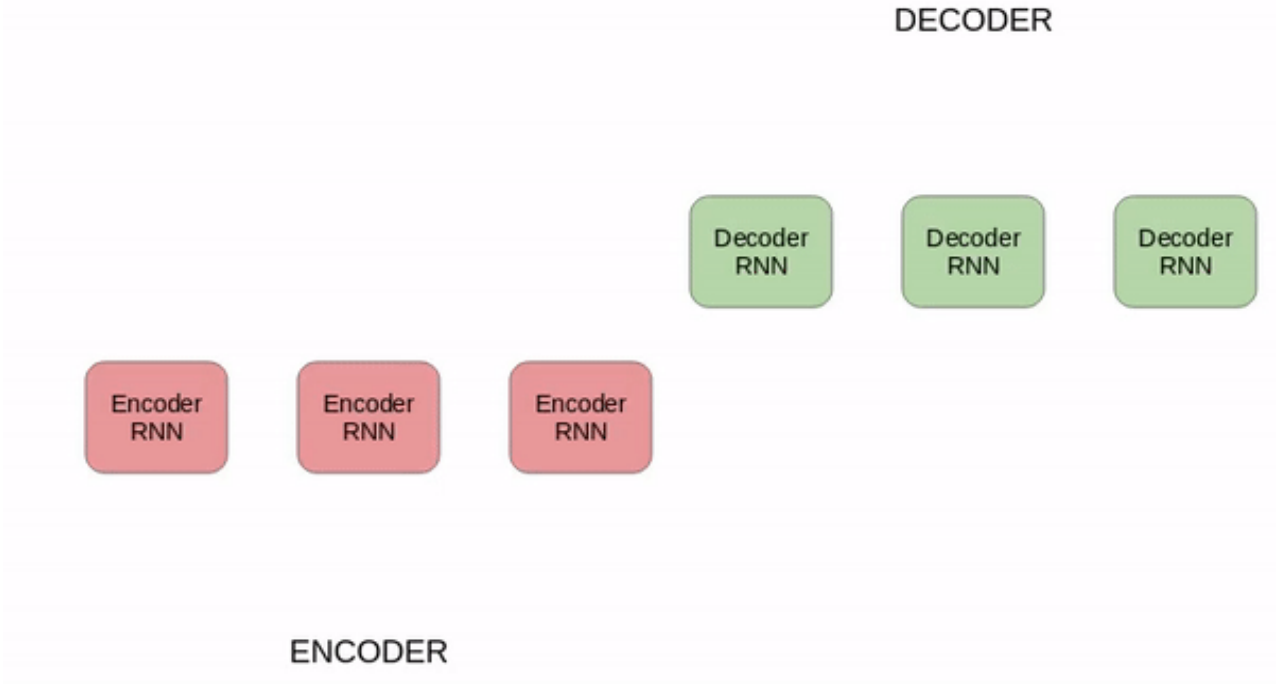


Image Credit: https://arxiv.org/abs/1611.04558?context=cs.AI

Issues with RNNs for seq2seq tasks

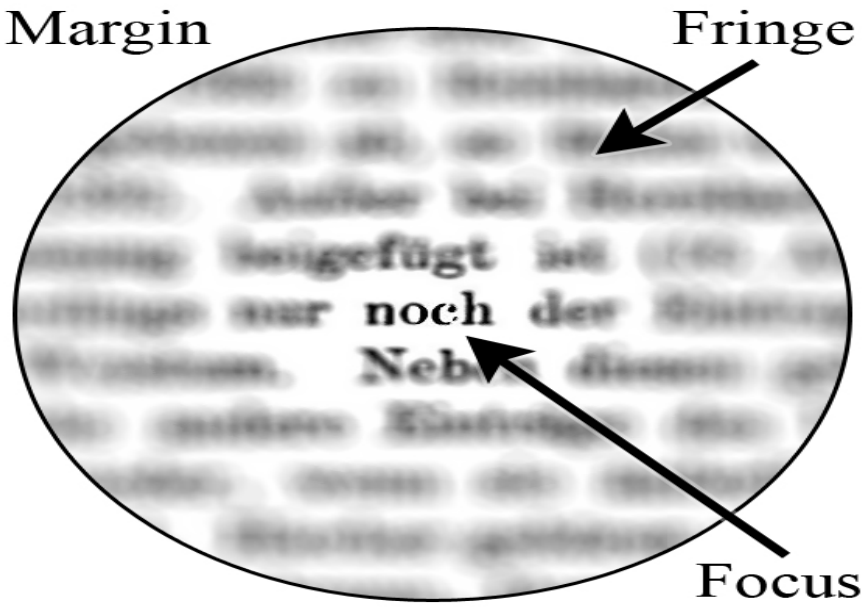
- Despite being very successful for various NLP tasks, there were challenges:
  - dealing with long-range dependencies
  - the sequential nature of the architecture prevents parallelization
- Attention mechanism helped to overcome first issue to certain extent



Credit: Encoder-Decoder Architecture for Machine Translation. https://www.analyticsvidhya.com/blog/2019/06/understanding-transformers-nlp-state-of-the-art-models/

Attention Mechanism

- A set of mechanisms that limit some processing to a subset of incoming stimuli (reducing computational demands)
- Attention in neural networks
  - A mechanism that learns to focus on a subset of the input that is relevant to the task.



task context representation

attended feature $\rightarrow \hat{v} = f(h, V)$

learned attention function (neural net)

set of attention candidates

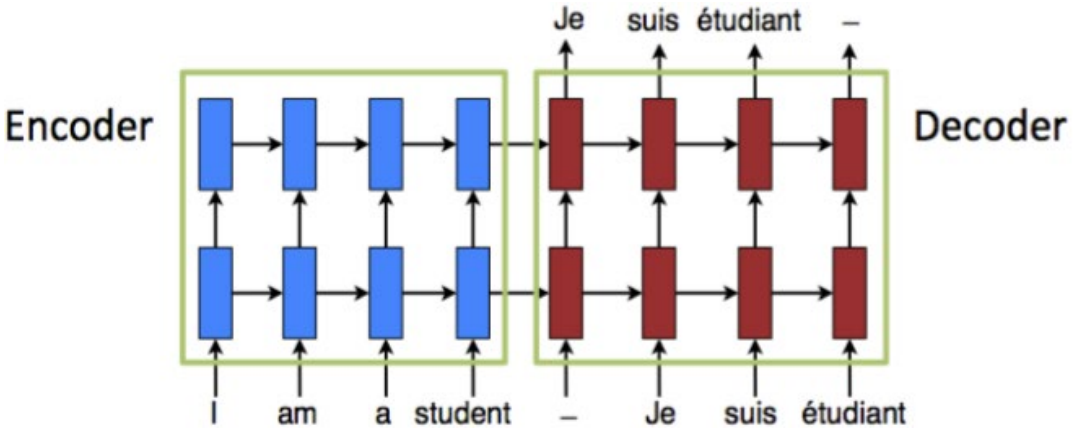
Neural Machine Translation by jointly Learning to Align and Translate

- NMT attempts to build and train a single, large neural network that reads a sentence and outputs a correct translation.
- The NMT system is a seq2seq model

  (Encoder-Decoder framework)
- Issue: The encoder-decoder framework compresses

  all the necessary information of a source sentence into

  a fixed-length vector.



- Cho et al. (2014) showed that the performance of a basic encoder-decoder deteriorates as the length of an input sentence increases.
- Bahdanau et al. (2015) proposed an extension to NMT which learns to align and translate jointly.
- The basic idea of “attention” is that at each time the model generates a word in a translation, it searches for a set of positions in a source sentence where the most relevant information is concentrated
- The model predicts a target word based on the context vectors associated with these source positions and all the previous generated target words

Credit: Bahdanau et al. (2015). Neural Machine Translation by jointly learning to Align and Translate. ICLR 2015.

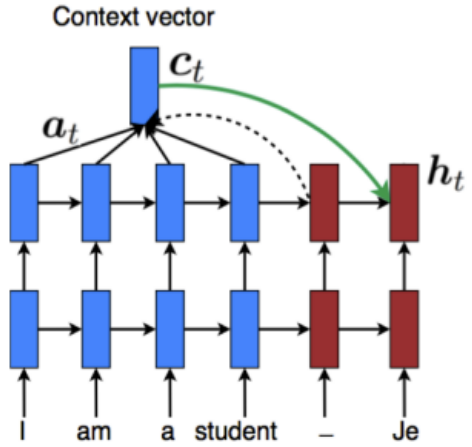
Neural Machine Translation by jointly Learning to Align and Translate

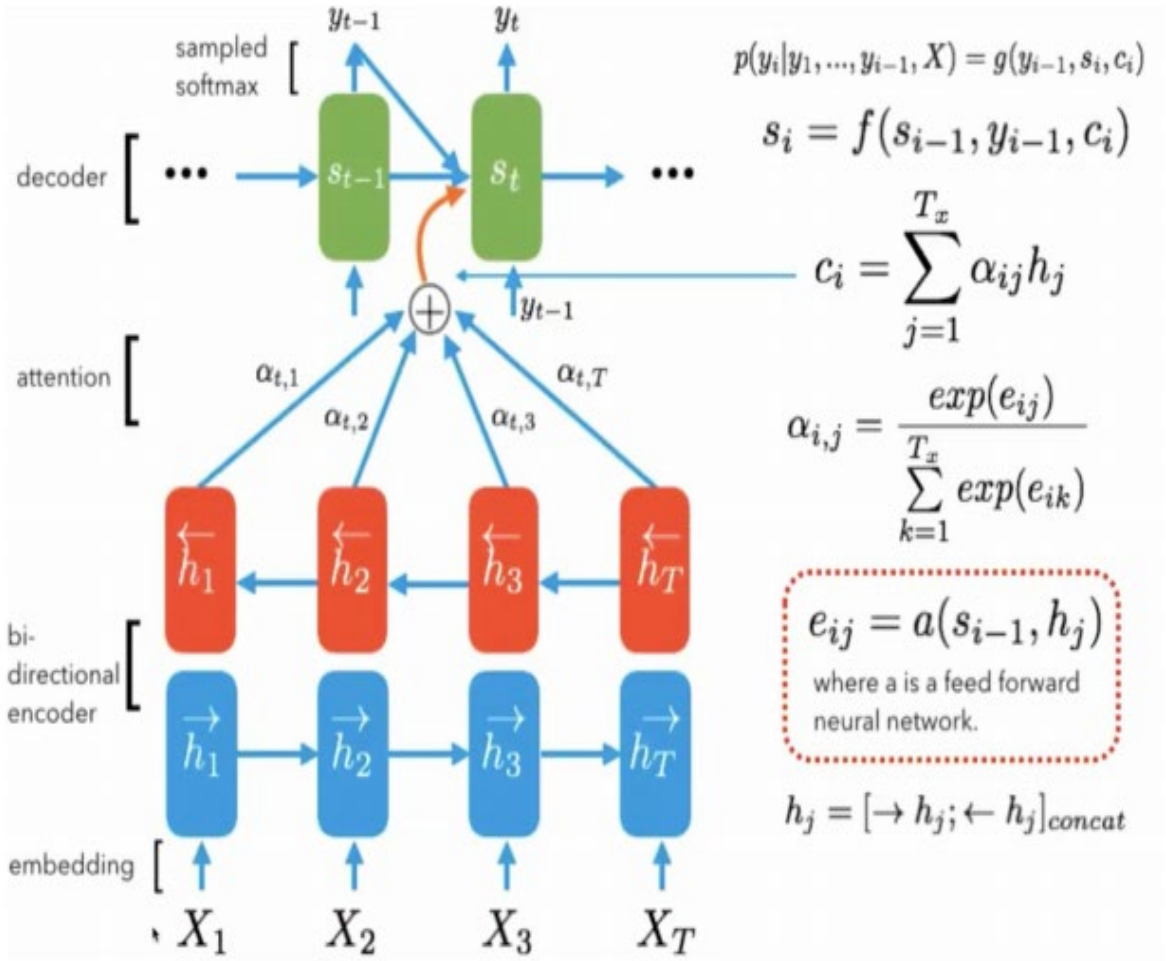
- Solution: Enable the network to pay attention to specific areas of the input by adding new (weighted) connections

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$$e_{ij} = a(s_{i-1}, h_j)$$

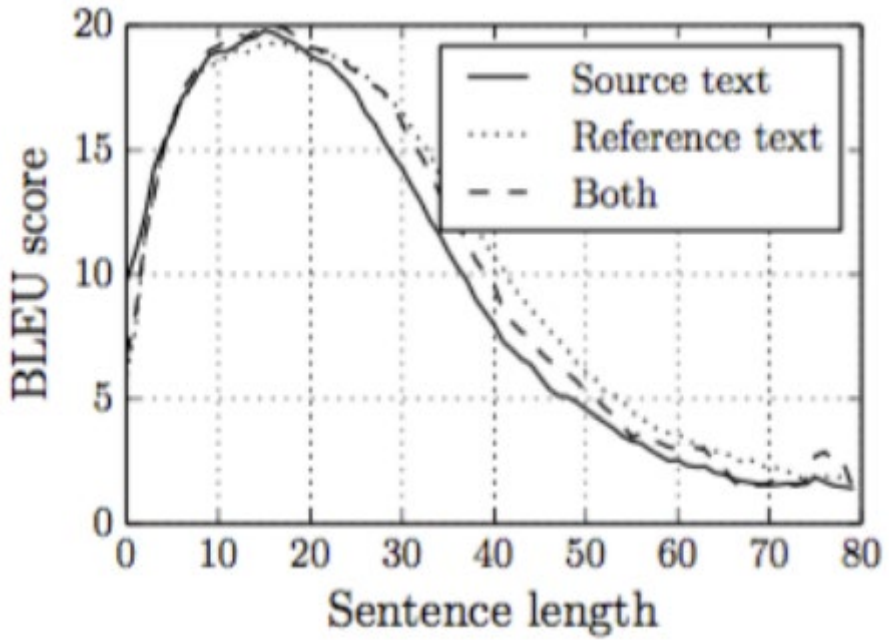




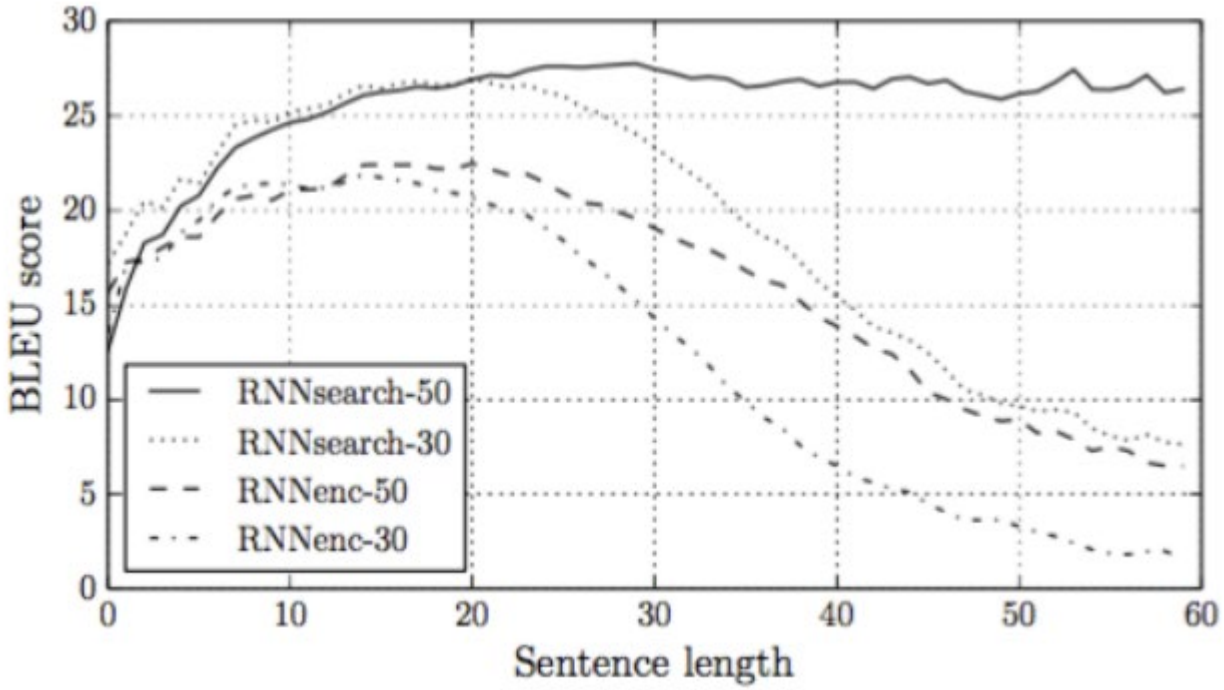
Credit: Bahdanau et al. (2015). Neural Machine Translation by jointly learning to Align and Translate. ICLR 2015.

NMT by jointly Learning to Align and Translate

Before Attention: Long sentences are very hard as they are compressed to a fixed length vector

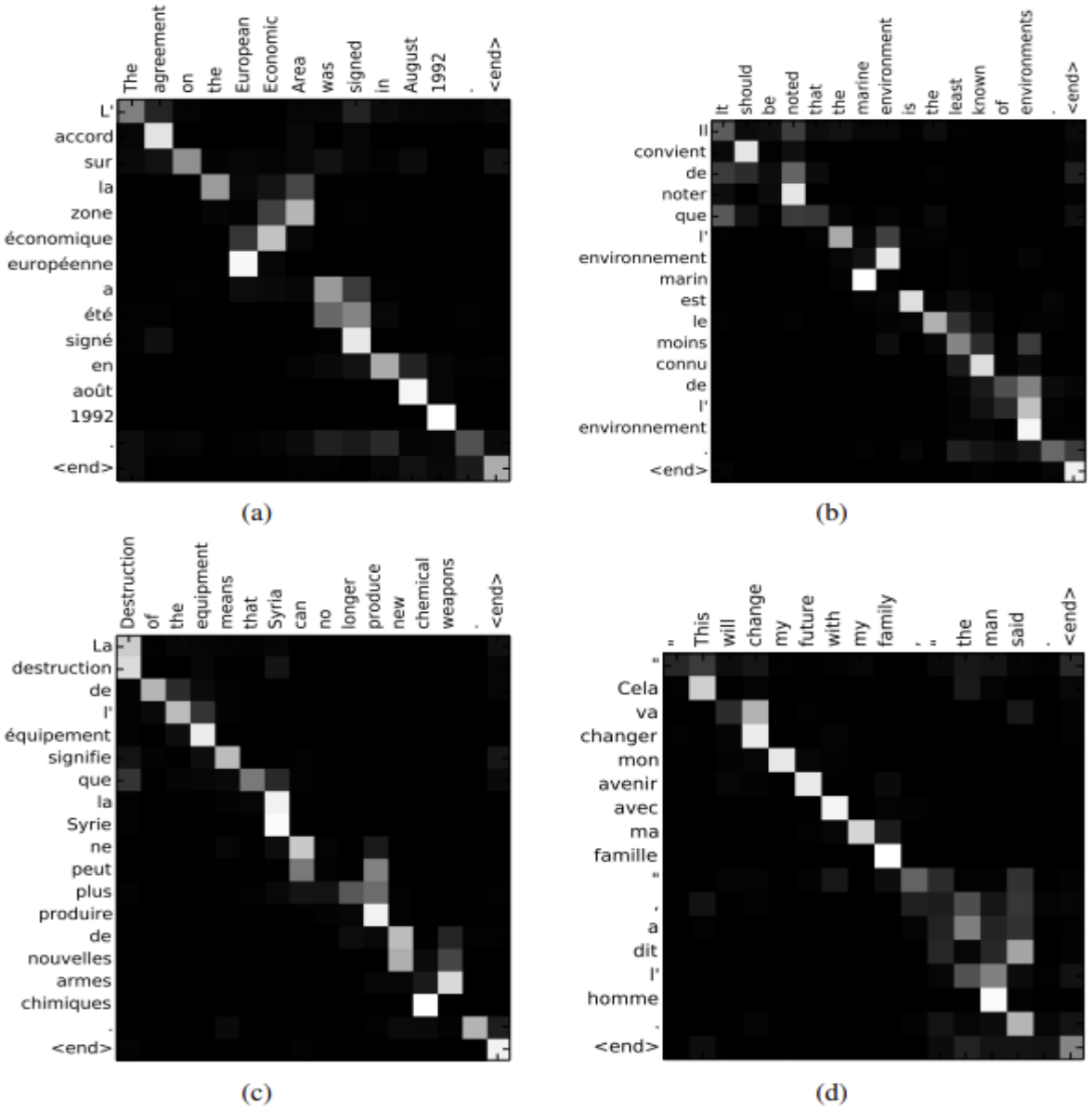


After Attention: The attention mechanism helps to overcome the issue



Credit: Bahdanau et al. (2015). Neural Machine Translation by jointly learning to Align and Translate. ICLR 2015.

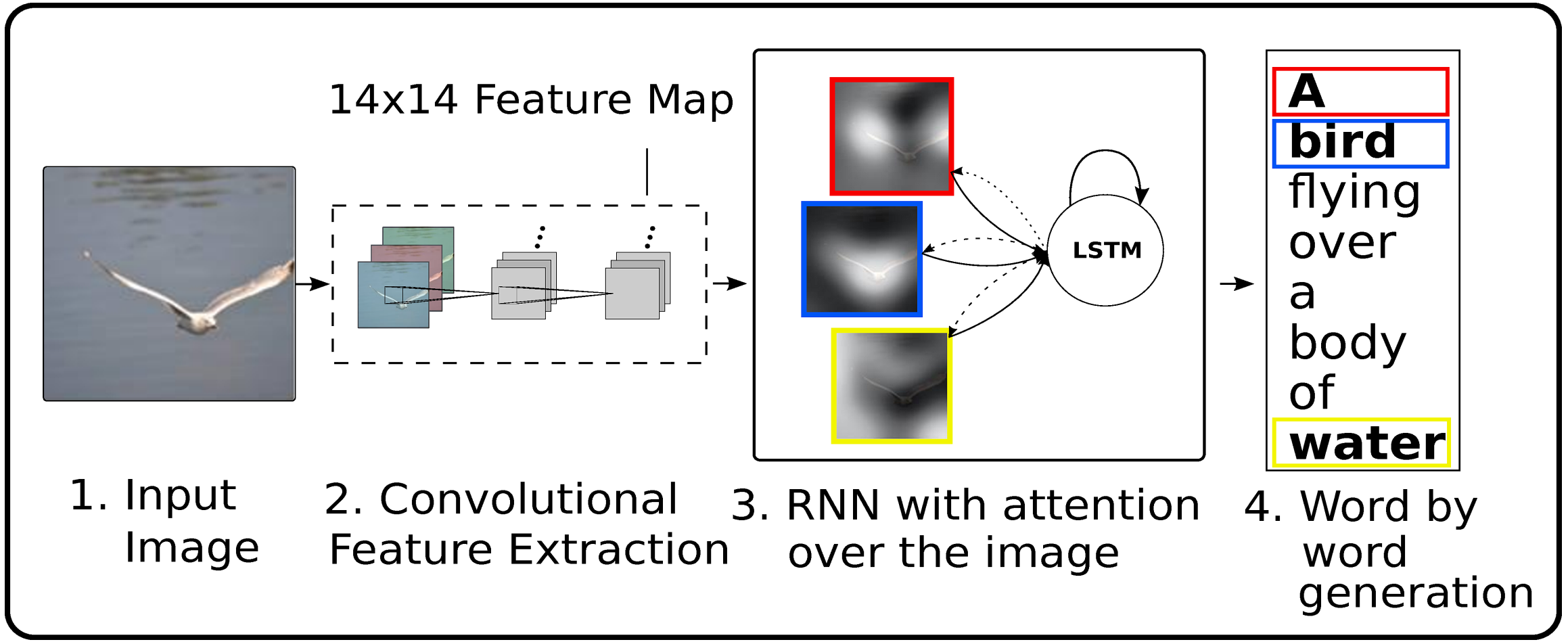
NMT by jointly Learning to Align and Translate



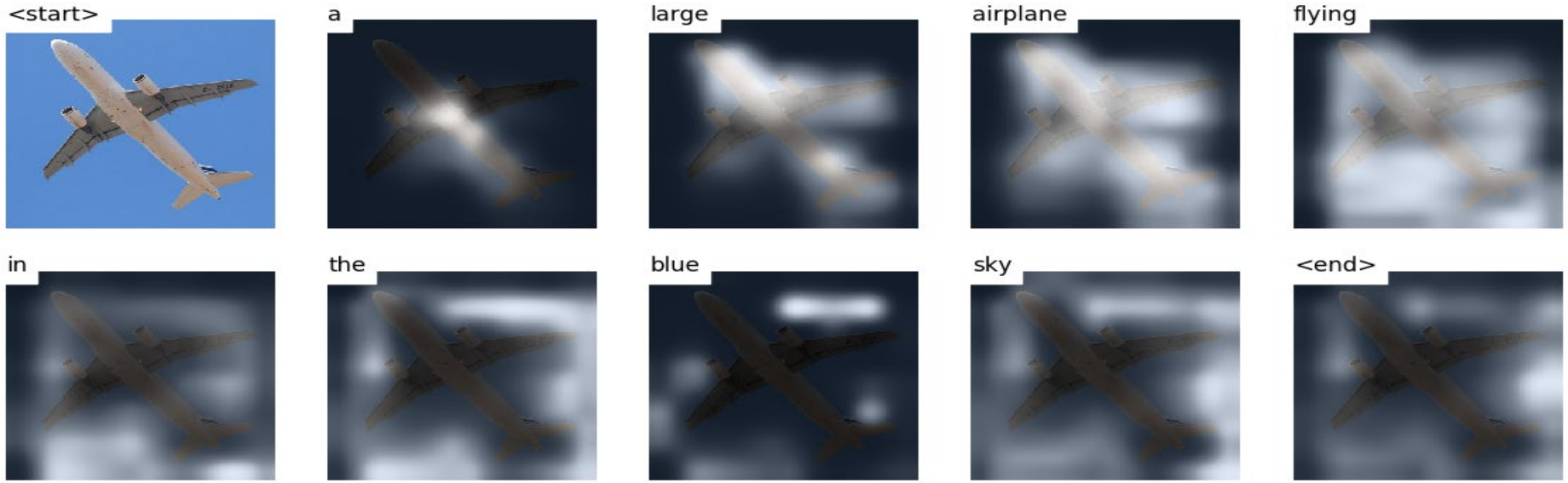
Credit: Bahdanau et al. (2015). Neural Machine Translation by jointly learning to Align and Translate. (CLR 2015.

Show, Attend and Tell (Xu et al., 2015)

- "Rather than compressing an entire image into static representation, attention allows for **salient features** to dynamically come to forefront as needed"



Show, Attend and Tell (Xu et al., 2015)

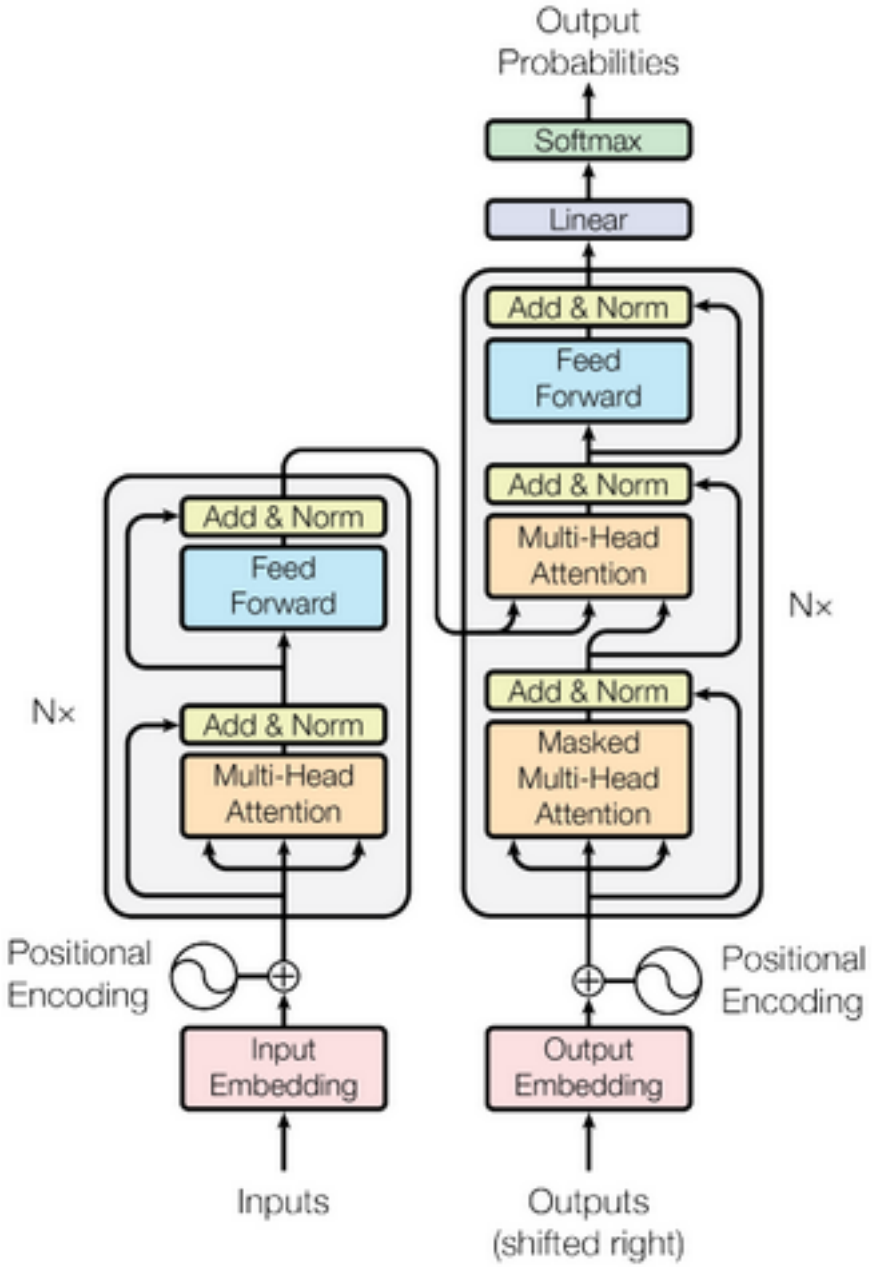


Source: Xu et al., (2015). Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Transformers

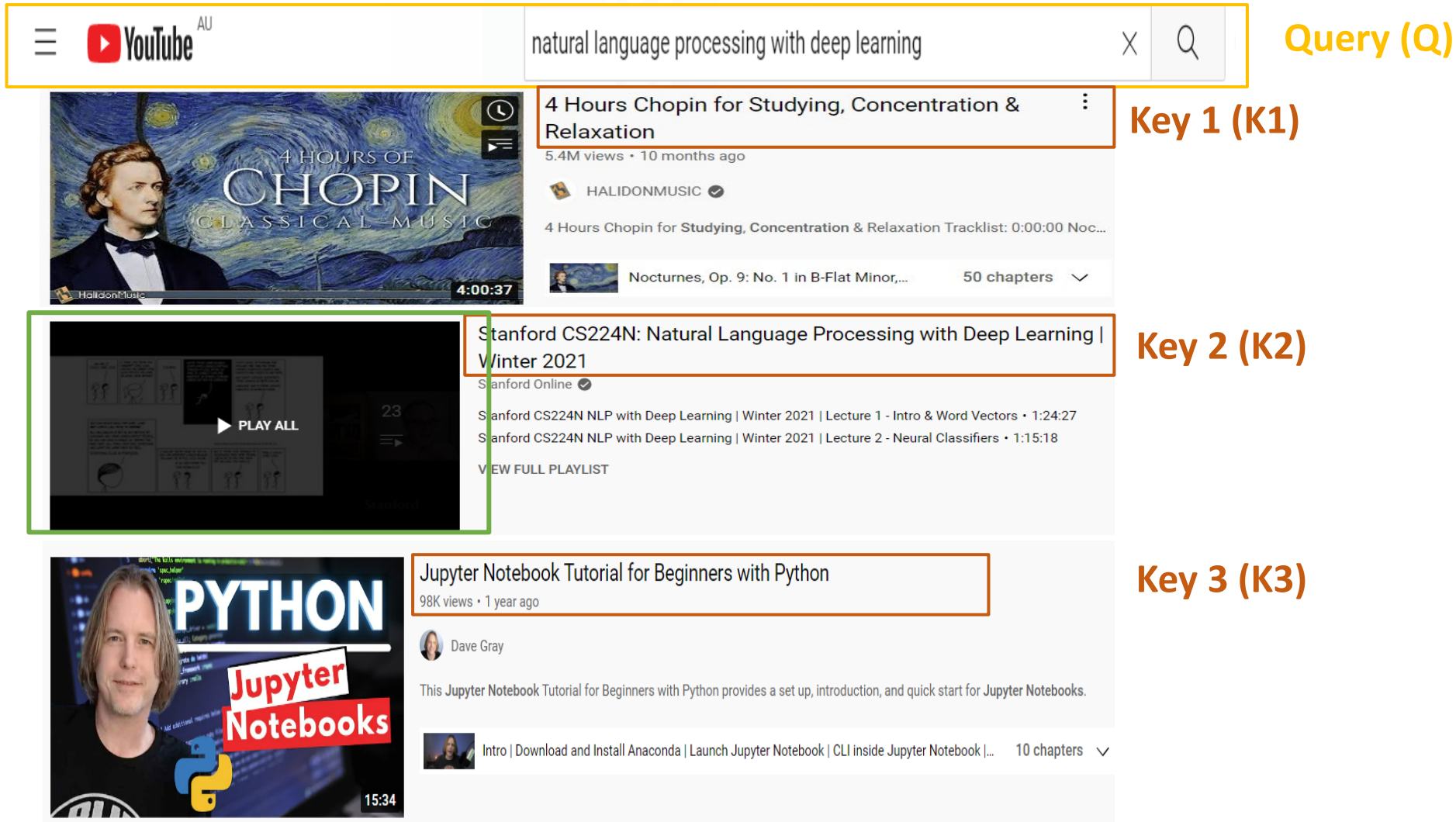
- Attention is All You Need
- Novel architecture relies entirely on **self-attention** to compute representations of its input and output without using sequential RNNs or convolutions.
- Aim is to solve seq2seq tasks while handling long-range dependencies

"Griezmann's announcement comes as a bit of a shock. After enduring the drama surrounding his potential last summer, many thought he was committed to Atletico for more than a year, but the Frenchman seems to have changed his mind."



Self-attention in transformers

- Understanding self-attention with Search
  - Query (Q)
  - Key (K)
  - Value (V)

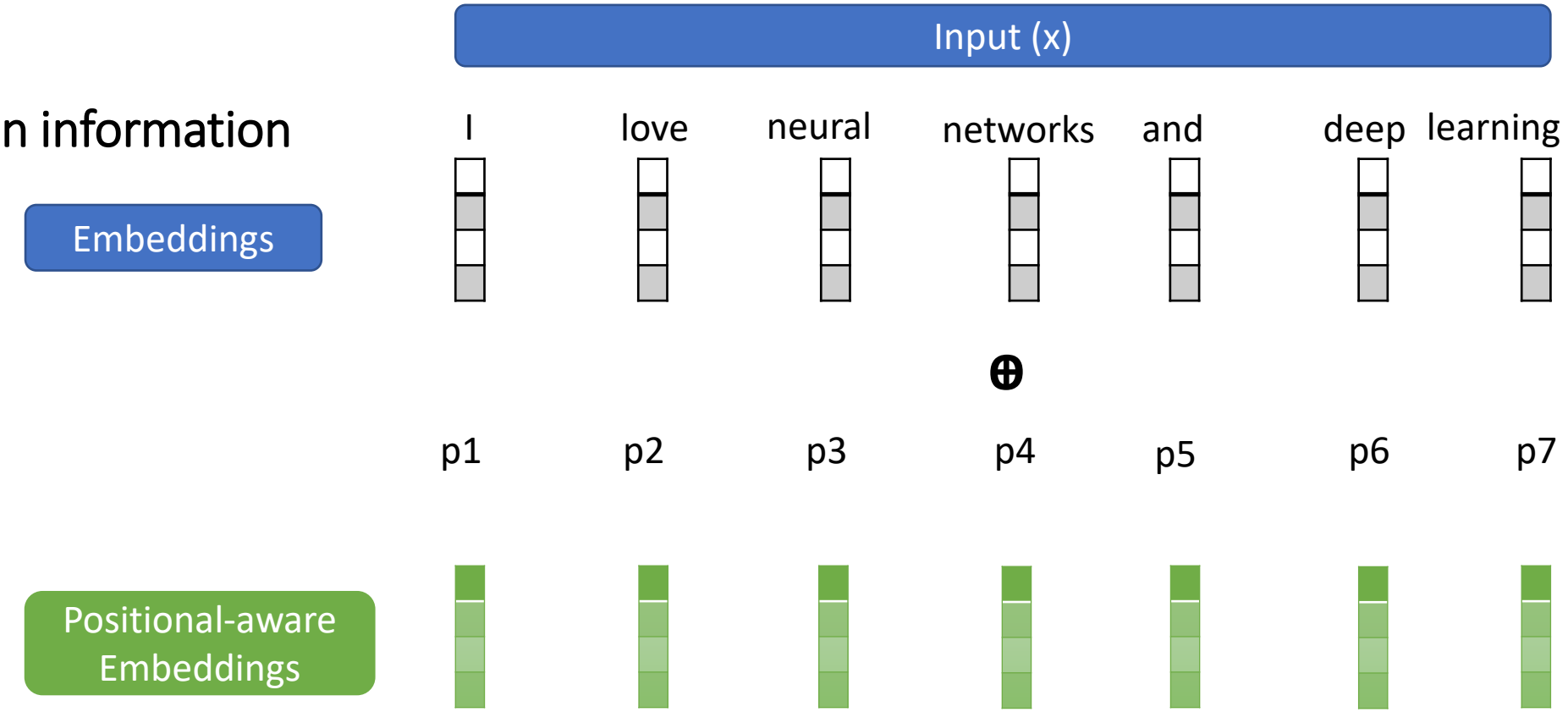


Self-attention in transformers

Self-attention:

Identify and attend to most important features in input

Step 1: Encode position information



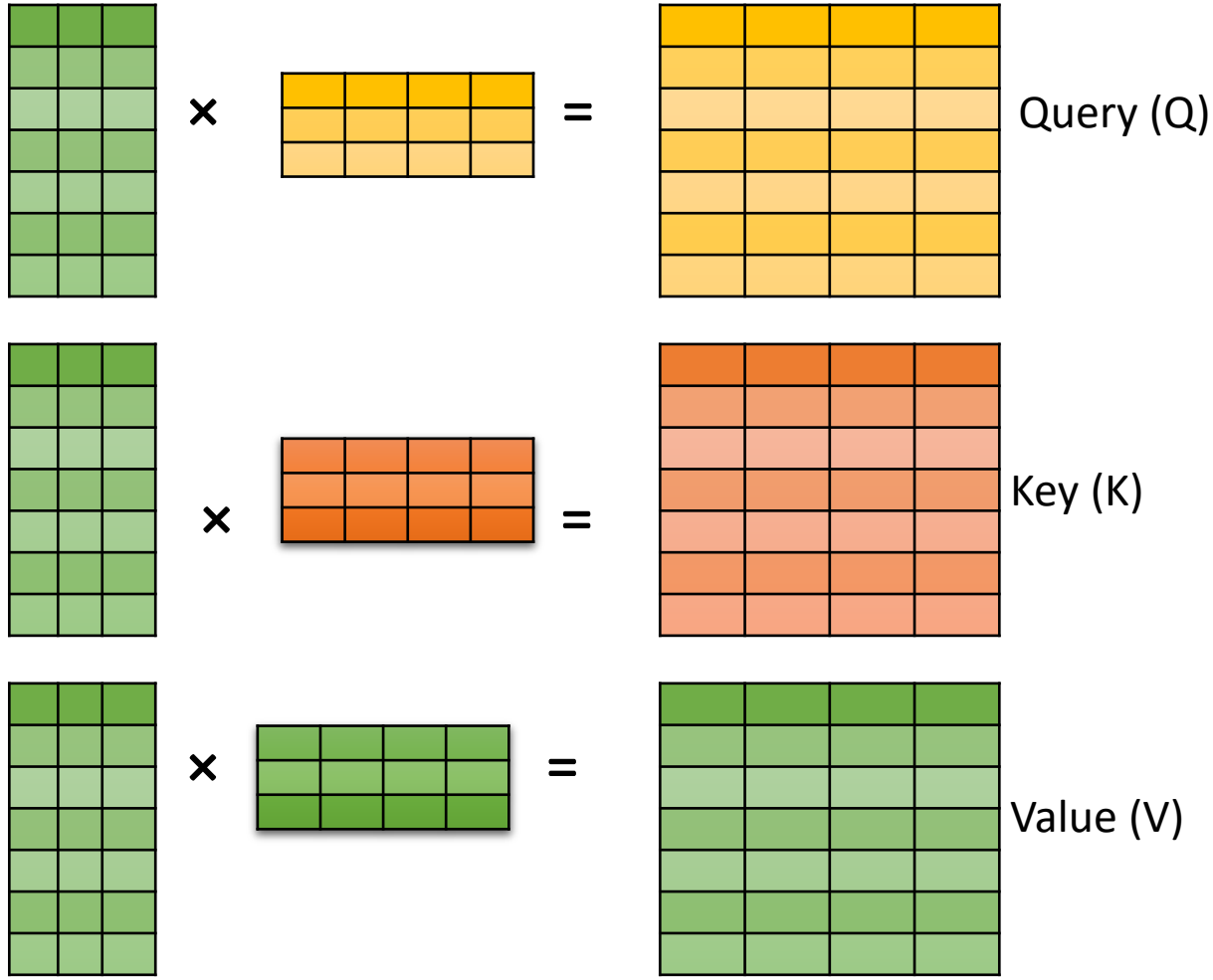
Self-attention in transformers

Self-attention:

Identify and attend to most important features in input

Step 1: Encode position information

Step 2: Extract query (Q), key (K), and value (V)



Self-attention in transformers

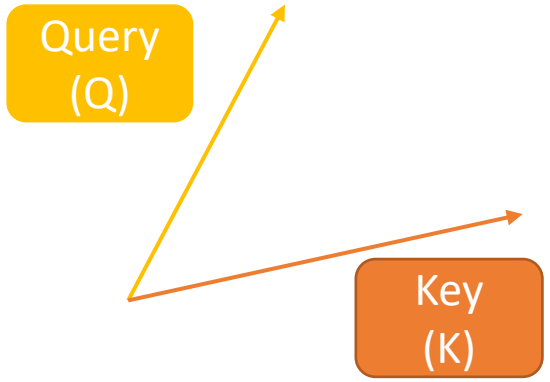
Self-attention:

Identify and attend to most important features in input

Step 1: Encode position information

Step 2: Extract query (Q), key (K), and value (V)

Step 3: Compute attention weighting



$$\frac{Q.\,K^T}{\text{Scaling}}$$

Self-attention in transformers

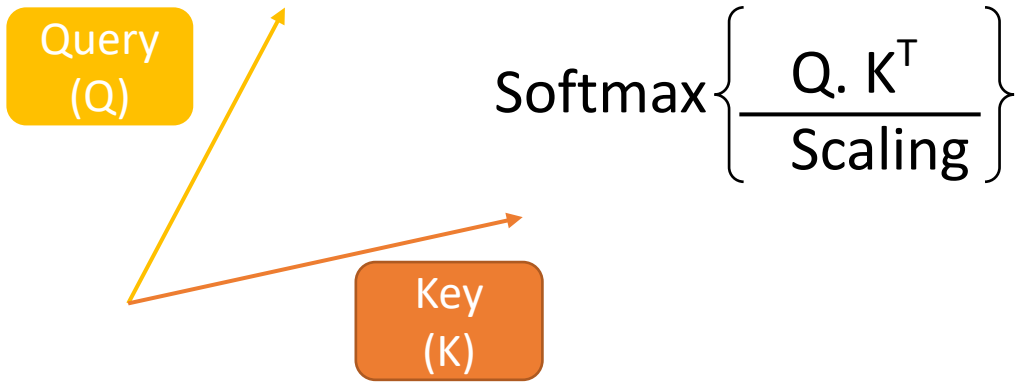
Self-attention:

Identify and attend to most important features in input

Step 1: Encode position information

Step 2: Extract query (Q), key (K), and value (V)

Step 3: Compute attention weighting



Self-attention in transformers

Self-attention:

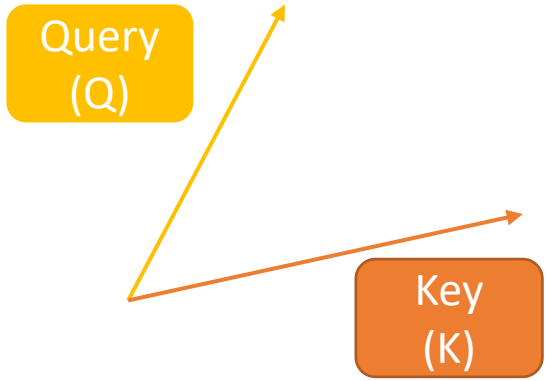
Identify and attend to most important features in input

Step 1: Encode position information

Step 2: Extract query (Q), key (K), and value (V)

Step 3: Compute attention weighting

Step 4: Extract features with high attention



$$\text{Softmax}\left\{\frac{Q.K^T}{\text{Scaling}}\right\} \times V = A(Q, K, V)$$

Self-attention in transformers

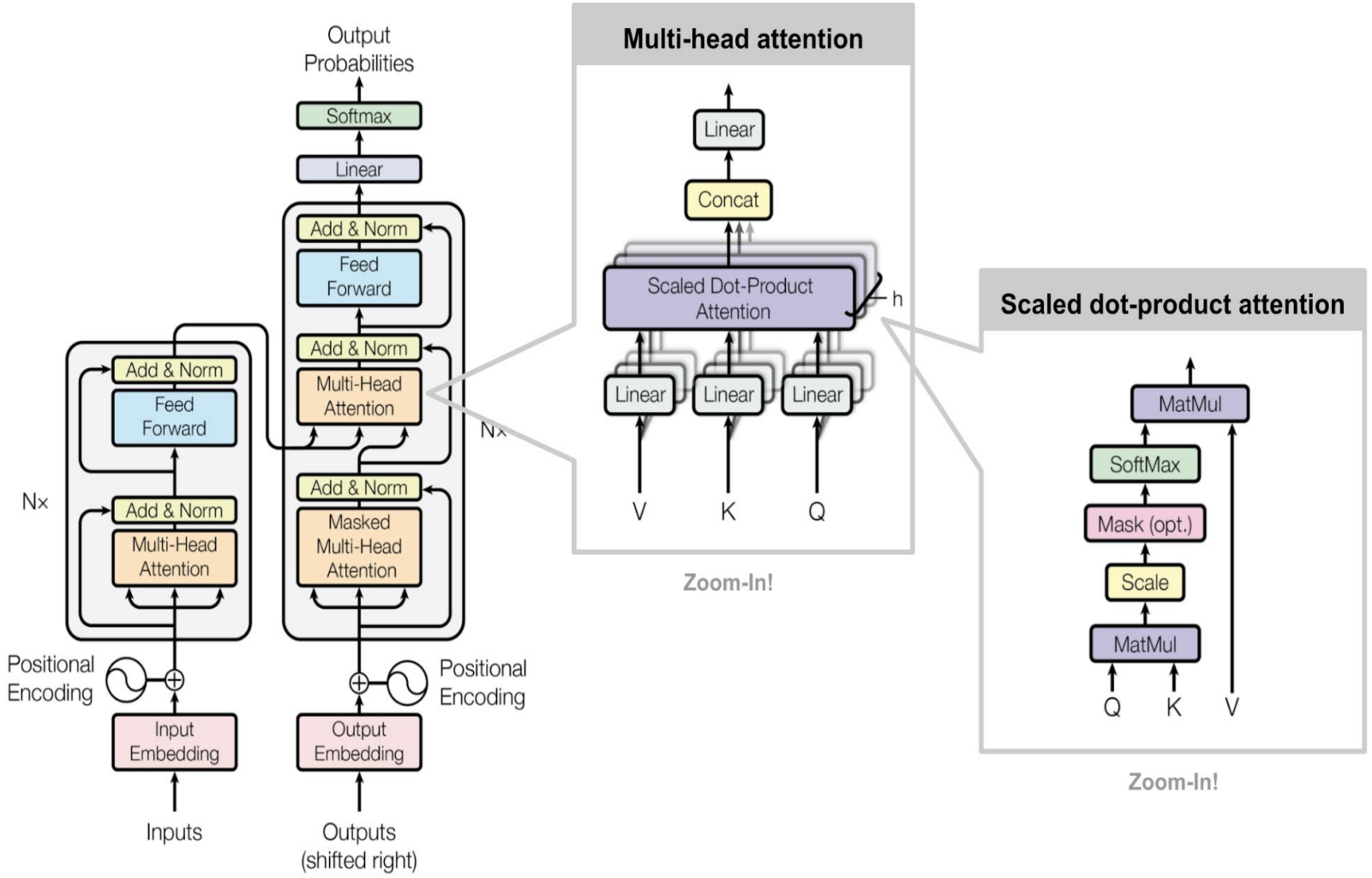
Self-attention:

Step 1: Encode position information

Step 2: Extract query (Q), key (K), and value (V

Step 3: Compute attention weighting

Step 4: Extract features with high attention



BERT

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
- Idea to have **deeply bidirectional, unsupervised language representation**
- Example sentence: "I accessed the bank account"
  - Unidirectional contextual model would represent "bank" based on "I accessed the"
  - BERT represent "bank" based on both its previous and next context – "I accessed the --- account"
- Why bidirectional in BERT possible?
  - Not possible to train bidirectional models by simply conditioning each word on its previous and next words, since this would allow the word that's being predicted to indirectly "see itself" in a multi-layer model.
  - BERT rely on two tasks
    - Masking words in the input
    - Next sentence prediction

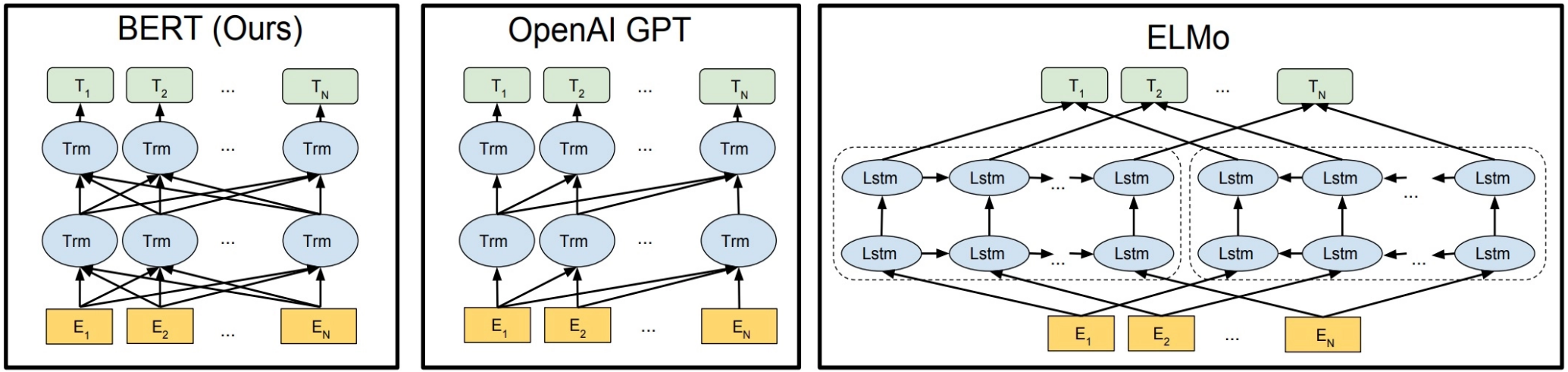
Input: The man went to the $[MASK]_1$. He bought a $[MASK]_2$ of milk .
Labels: $[MASK]_1$ = store; $[MASK]_2$ = gallon

Sentence A = The man went to the store.
Sentence B = He bought a gallon of milk.
Label = IsNextSentence

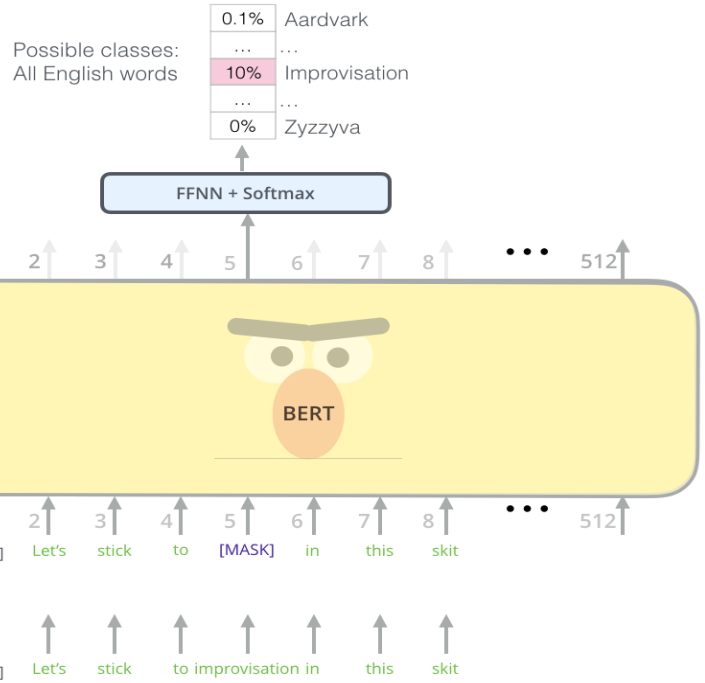
Sentence A = The man went to the store.
Sentence B = Penguins are flightless.
Label = NotNextSentence

Credit: Devlin et al., BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Google AI Blog. https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

BERT



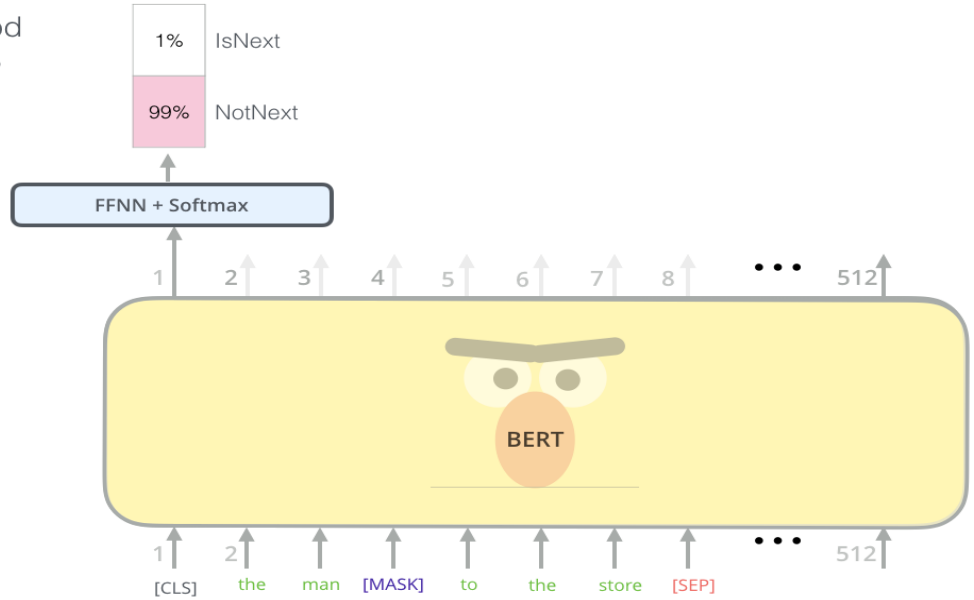
Use the output of the masked word's position to predict the masked word



Randomly mask 15% of tokens

Input

Predict likelihood that sentence B belongs after sentence A



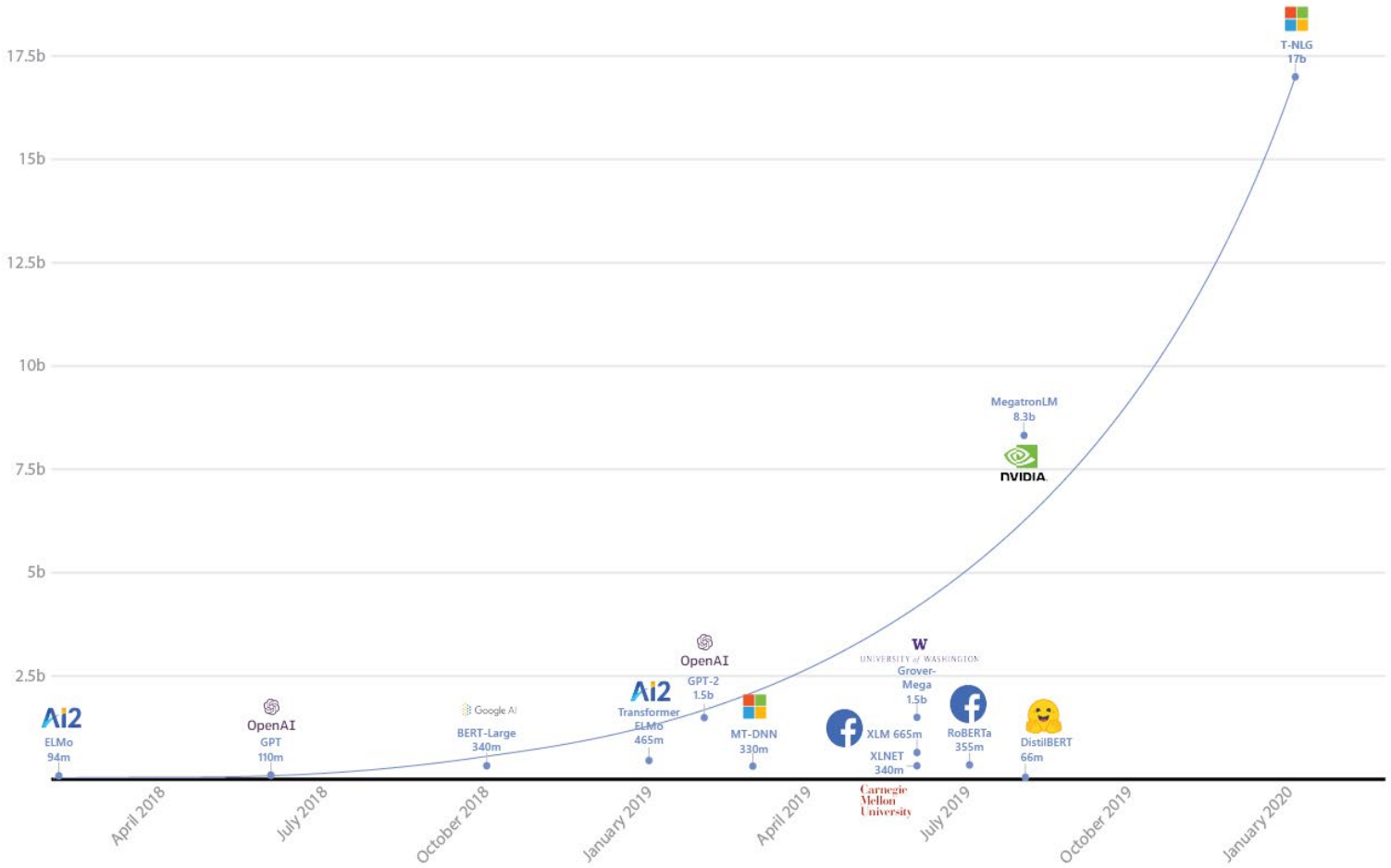
Tokenized Input

Input

Credit: Jay Alammar. The illustrated BERT, ELMo, and co. https://jalammar.github.io/illustrated-bert/

Turing-NLG

- A Transformer based generative language model
- Based on DeepSpeed (DL library) to make distributed training of large models easier
- 17 billion parameters



Credit: Rosset, Corby. (2020). Turing-NLG: A 17-billion parameter language model by Microsoft, Microsoft Research
Microsoft Turing-NLG. https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

OpenAI GPT-3

- Key idea: Improve task-agnostic few-shot performance
- Evaluation on various NLP tasks under few-shot learning, one-shot learning, and zero-shot learning demonstrates GPT-3 promising results
- Technical details:
  - An autoregressive language model
  - GPT-3 has 96 layers with each layer having 96 attention heads
  - trained on datasets with 500 billion tokens
  - Word embedding size of 12888
  - Context window size is of 2048 tokens
  - Uses alternating dense and locally banded sparse attention patterns
- Compute:
  - Trained on more than 576 GB of text data including common crawl, Books, and Wikipedia
  - About 175 billion parameters
  - costs OpenAI around $4.6 million

Credit: Brown et al., (2020). Language models are few-shot learners.
Priya Shree. Medium: The journey of OpenAI GPT models. https://medium.com/walmartglobaltech/the-journey-of-open-ai-gpt-models-32d95b7b7fb2

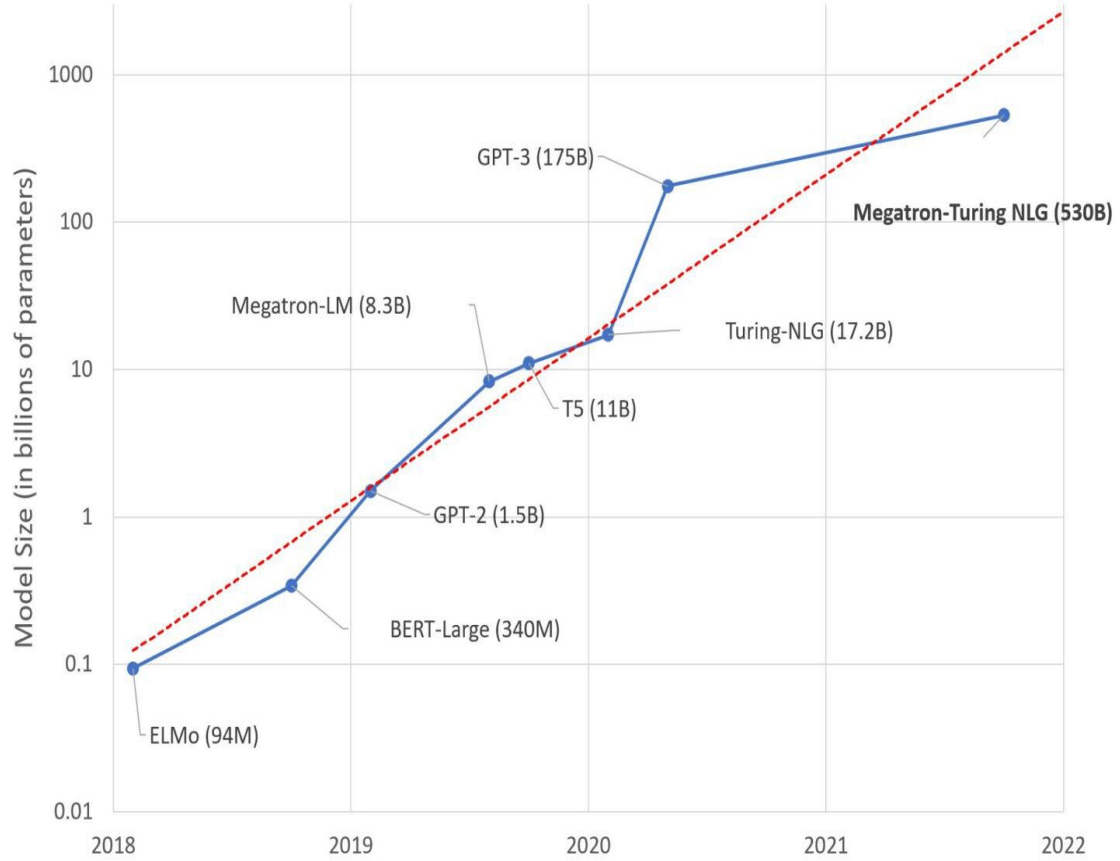
MT-NLG

Behold the Megatron: Microsoft and Nvidia build massive language processor

MT-NLG is a beast that fed on over 4,000 GPUs

Katyanna Quach — Tue 12 Oct 2021 // 00:36 UTC

- Largest transformer model to date
- 530 billion parameters
- Trained on a wide variety of NLP tasks such as auto-completing sentences, question and answering, reading and reasoning
- Outperforms various tasks with little to no fine-tuning, in a few-shot or zero-shot learning
- Trained using Nvidia's Selene machine learning supercomputer
- Estimated to cost $85 million



Source: https://turing.microsoft.com/
The Register. https://www.theregister.com/2021/10/12/nvidia_microsoft_mtnlg/

Where is the journey heading?

- Are bigger language models better?
  - Bigger models are better at generalizing on various downstream tasks
  - Issues: Bias, carbon footprints

ARTICLE OPEN ACCESS

On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? 🦜

Authors: Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, Shmargaret Shmitchell Authors Info & Claims

FAccT '21: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency • March 2021 • Pages 610–623 • https://doi.org/10.1145/3442188.3445922

Published: 03 March 2021

Large computer language models carry environmental, social risks

It takes an enormous amount of computing power to fuel the model language programs, Bender said. That takes up energy at tremendous scale, and that, the authors argue, causes environmental degradation. And those costs aren't borne by the computer engineers, but rather by marginalized people who cannot afford the environmental costs.

Another risk comes from the training data itself, the authors say. Because the computers read language from the Web and from other sources, they can pick up and perpetuate racist, sexist, ableist, extremist and other harmful ideologies.

Yann LeCun
27 October 2020 at 13:01 ·

"It's entertaining, and perhaps mildly useful as a creative help," LeCun wrote. "But trying to build intelligent machines by scaling up language models is like building high-altitude airplanes to go to the moon. You might beat altitude records, but going to the moon will require a completely different approach."

Credit: University of Washington. https://www.washington.edu/news/2021/03/10/large-computer-language-models-carry-environmental-social-risks/

Where is the journey heading?

Deep Learning, Deep Pockets?

As you would expect, training a 530-billion parameter model on humongous text datasets requires a fair bit of infrastructure. In fact, Microsoft and NVIDIA used hundreds of DGX A100 multi-GPU servers. At $199,000 a piece, and factoring in networking equipment, hosting costs, etc., anyone looking to replicate this experiment would have to spend close to $100 million dollars. Want fries with that?

Source: Hugging Face. https://huggingface.co/blog/large-language-models

Summary

- Large language modeling is like race between titans
- Large language model size is increasing 10x every year for the last few years. Moore's Law
- Issues:
  - Environmental cost (carbon footprint)
  - Hard to replicate
  - Very complex
- The need:
  - To build practical and efficient solutions
  - Within everyone's capability and reach to solve real-world problems

Credit: Brown et al., (2020). Language models are few-shot learners.
Priya Shree. Medium: The journey of OpenAI GPT models. https://medium.com/walmartglobaltech/the-journey-of-open-ai-gpt-models-32d95b7b7fb2

References

[1] Rosenfield R. *Two decades of statistical language modeling: where do we go from here?,* Proceedings of IEEE, 2000.

[2] Hochreiter and Schmidhuber, *Long Short-Term Memory*, Neural Computation, 1997.

[3] Cho et al., *Learning phrase representations using RNN Encoder-Decoder for Statistical Machine Translation*. EMNLP 2014.

[4] Mikolov et al., *Distributed representations of words and phrases and their compositionality*, NeurIPS 2013. [Google Word2Vec]

[5] Pennington et al., *GloVe: Global Vectors for Word Representation*, EMNLP 2014. [Stanford GloVe]

[6] McCann et al., *Learned in translation: contextualized word vectors*, NeurIPS 2017. [Salesforce CoVe]

[7] Peters et al., *Deep contextualized word representations*, NAACL 2018. [AllenNLP ELMo]

[8] Howard and Ruder, *Universal Language Model Fine-tuning for Text Classification*, ACL 2018. [FastAI ULMFiT]

[9] Radford et al., *Improving Language Understanding by Generative Pre-Training*. [OpenAI GPT-1]

[10] Vaswani et al., *Attention is All You Need*, NeurIPS 2017. [Google Transformer]

[11] Devlin et al.., *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, NAACL-HLT 2019. [Google BERT]

[12] Yang et al., *XLNet: Generalized Autoregressive Pretraining for Language Understanding*, NeurIPS 2019. [Google XLNet]

[13] Shoeybi et al., *Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism*, ArXiv 2019. [Nvidia Megatron-LM]

[14] Rosset et al., *Turing-NLG: A 17-billion-parameter language model by Microsoft*, Microsoft Research Blog, 2020. [Microsoft Turing-NLG]

[15] Brown et al., *Language Models are Few-Shot Learners*, ArXiv 2020. [OpenAI GPT-3]

[16] Clark et al., *ELECTRA: Pre-training text encoders as discriminators rather than generators*, ICLR 2020. [Stanford + Google ELECTRA]

[17] *DeepSpeed and Megatron-powered Megatron-Turing Natural Language Generation model* (MT-NLG). [Microsoft + Nvidia Megatron-NLG]

[18] https://www.topbots.com/leading-nlp-language-models-2020/

Questions?