

Large Language Models

And NLP Applications

Sri Vallabha Deevi, Ph.D.

Director, Data Science, Tiger Analytics.
Adjunct Faculty, Data Science & AI, IIT Madras.

Table of contents

1. Deep Learning
2. NLP Models
3. Large Language Models
4. NLP Applications

Deep Learning

Deep Learning

- Learn hierarchical data representations
- Deep structured learning / hierarchical learning
- Artificial Neural Networks
- Many layers ($\approx 10,000$ variables)
- Huge demand on computational power

Artificial Neuron

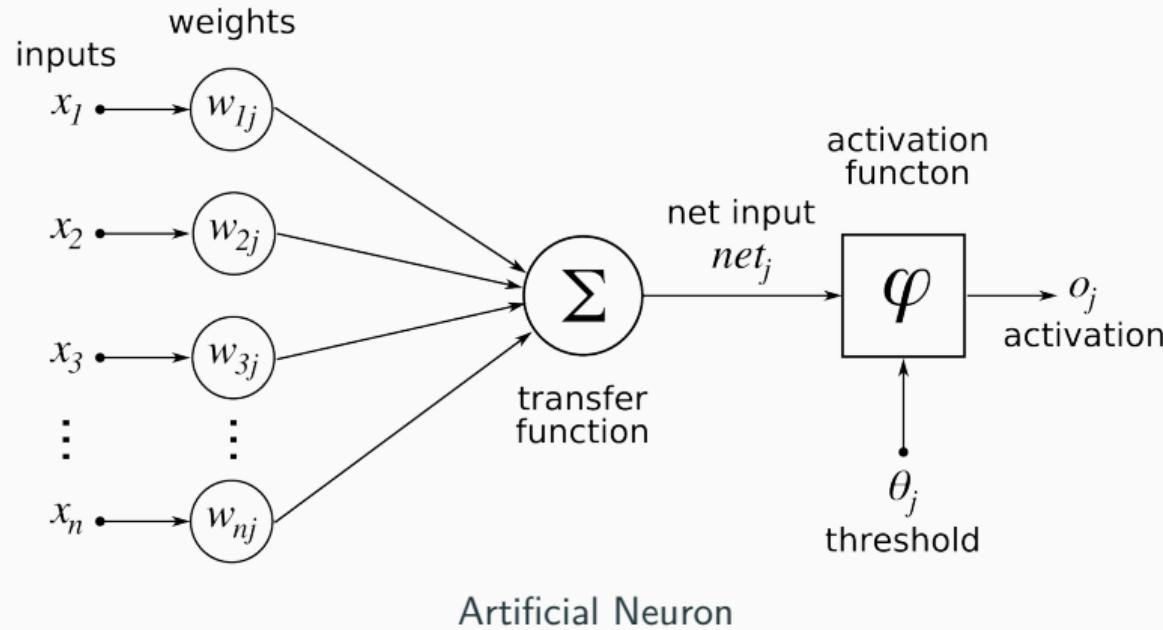
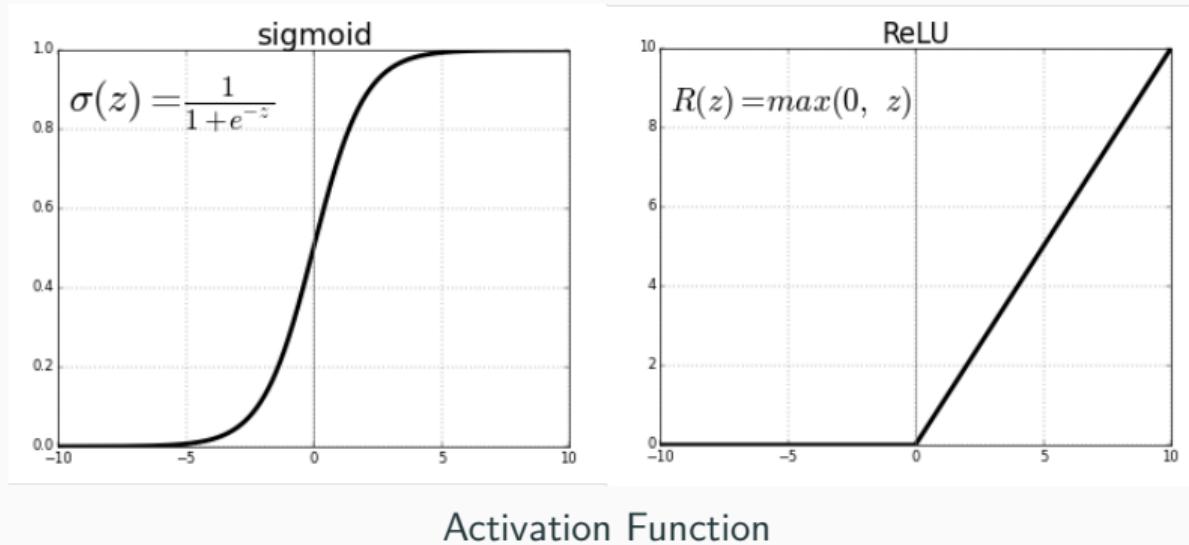


Image: By Chrislb (created by Chrislb) [GFDL (<http://www.gnu.org/copyleft/fdl.html>) or CC-BY-SA-3.0 (<http://creativecommons.org/licenses/by-sa/3.0/>)], via Wikimedia Commons

Activation Function



Artificial Neural Network

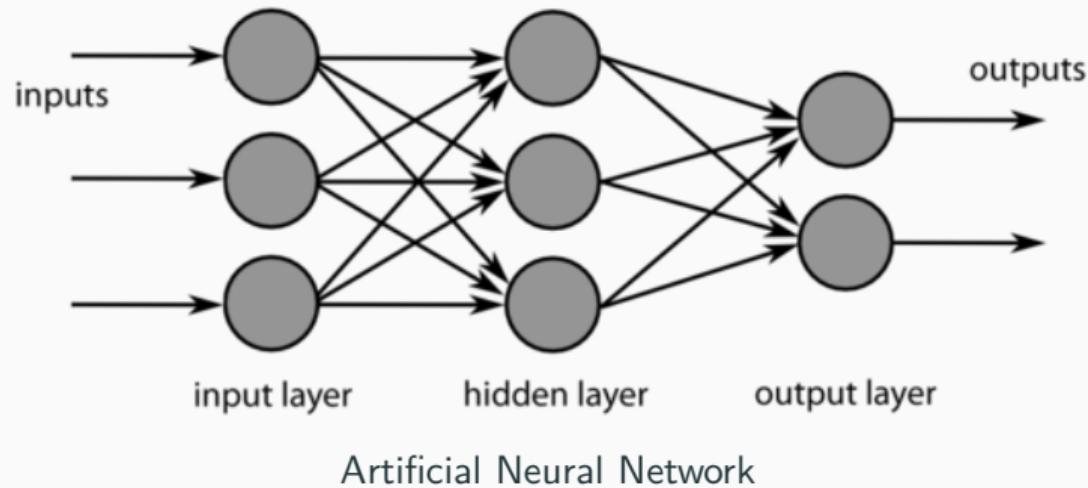
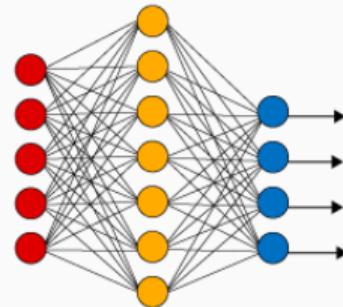


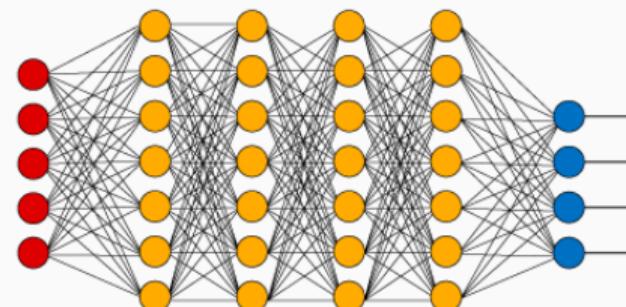
Image: By MultiLayerNeuralNetwork_english.png: Chrislb [CC BY-SA 3.0], via Wikimedia Commons

Deep Neural Network

Simple Neural Network



Deep Learning Neural Network



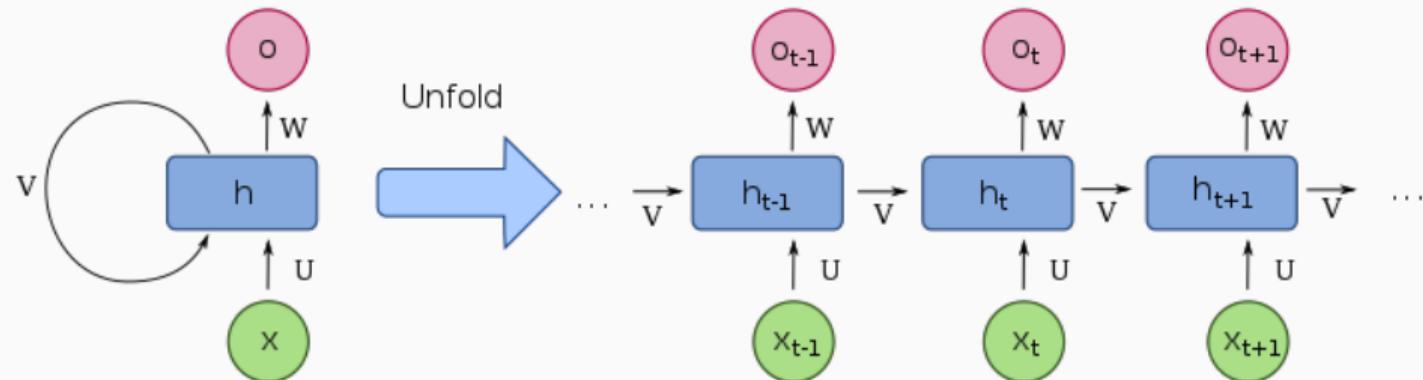
● Input Layer

● Hidden Layer

● Output Layer

Deep Neural Network

Recurrent Neural Network



Recurrent Neural Network

Image: By François Deloche (Own work) [CC BY-SA 4.0], via Wikimedia Commons

Convolutional Neural Network

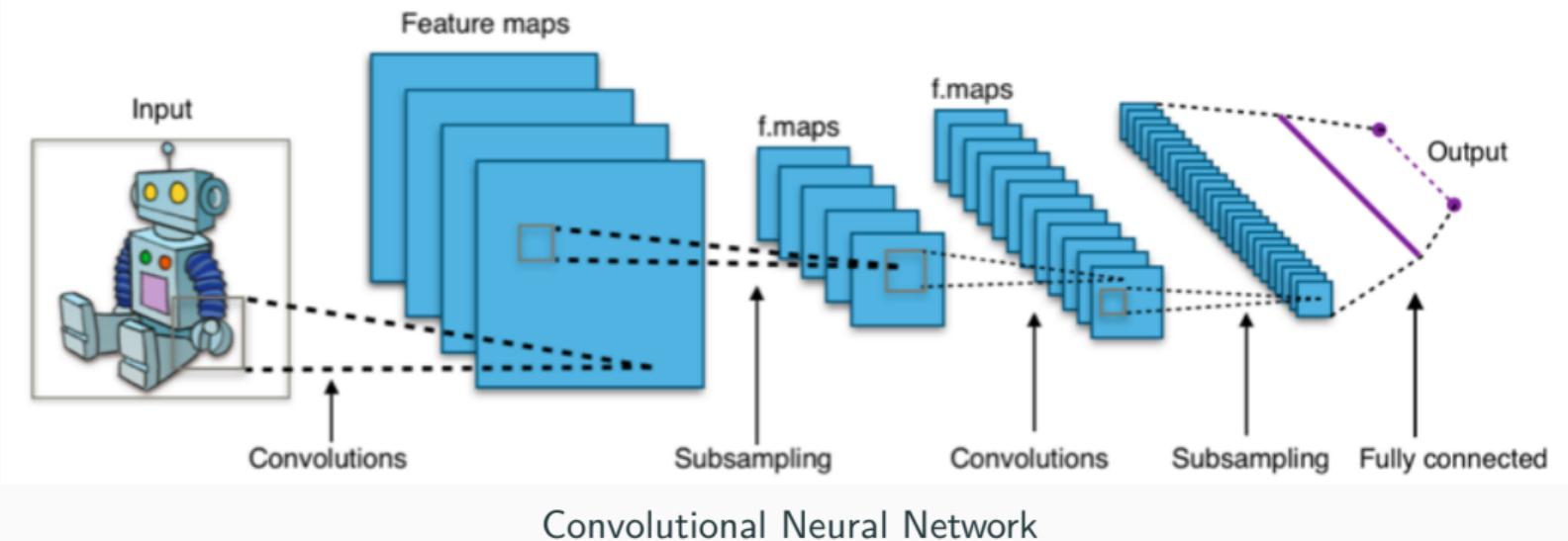
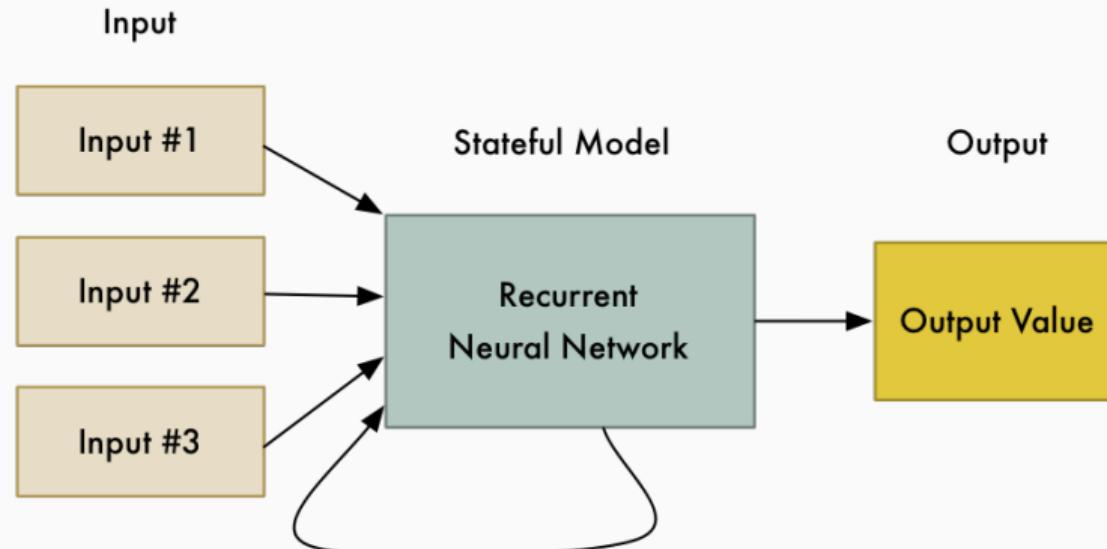


Image: By Aphex34 (Own work) [CC BY-SA 4.0], via Wikimedia Commons

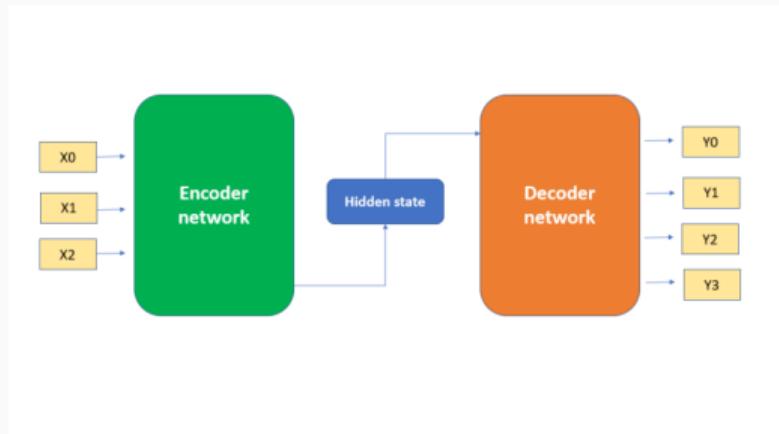
NLP Models

Recurrent Neural Network



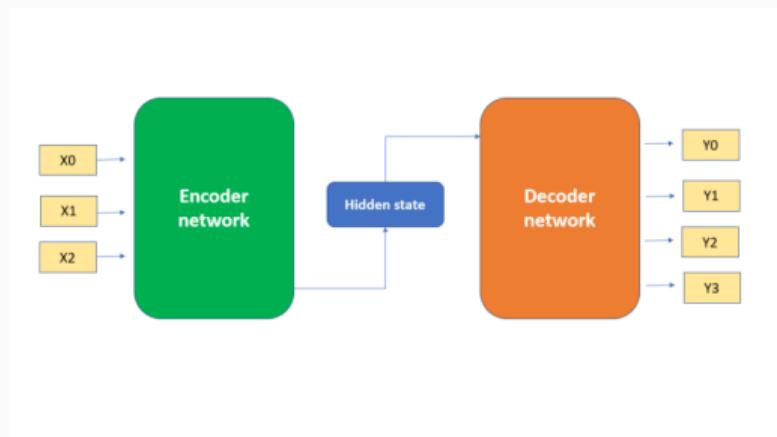
*Save the model's current state
and use that as one input
of our next calculation.*

Encoding

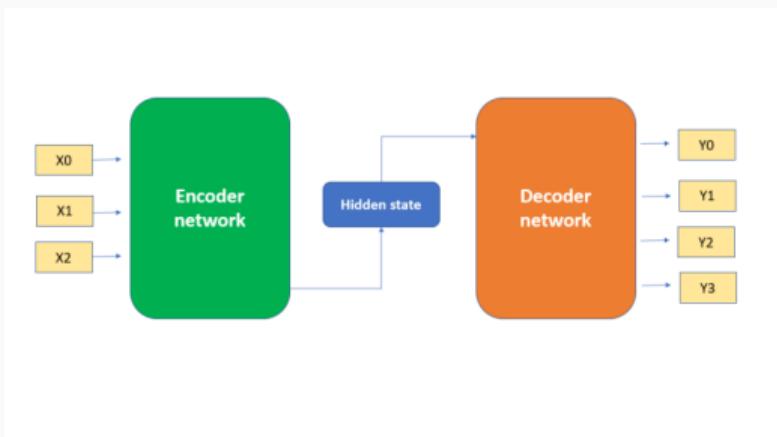


Encoding

- Converts a string of arbitrary length to a fixed length sequence of numbers
 - hidden state / context vector

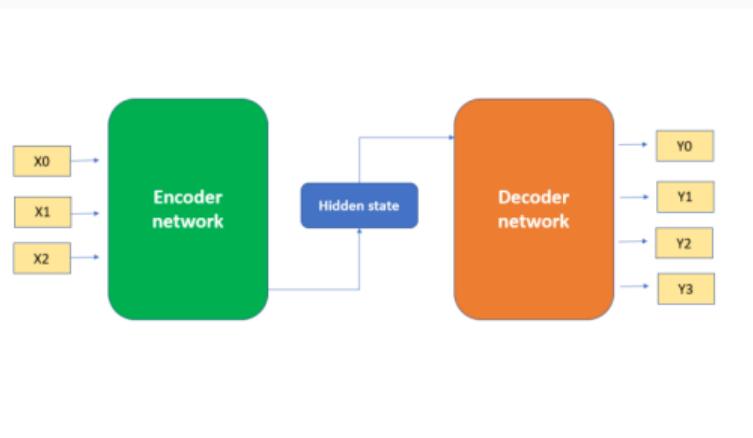


Encoding



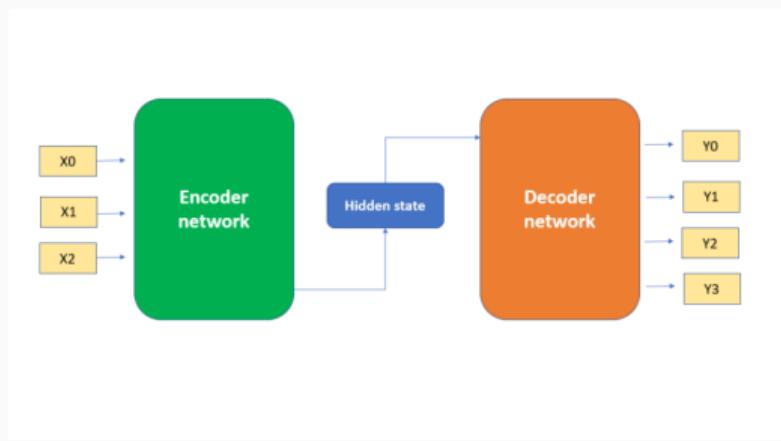
- Converts a string of arbitrary length to a fixed length sequence of numbers
 - hidden state / context vector
- Hidden state can be used for many ML tasks - classification, text generation etc.

Encoding



- Converts a string of arbitrary length to a fixed length sequence of numbers
 - hidden state / context vector
- Hidden state can be used for many ML tasks - classification, text generation etc.
- Hidden state can be used to decode text to different languages using decoder models

Encoding



- Converts a string of arbitrary length to a fixed length sequence of numbers
 - hidden state / context vector
- Hidden state can be used for many ML tasks - classification, text generation etc.
- Hidden state can be used to decode text to different languages using decoder models
- Decoder generates an output string of variable length in the target language

Image: <https://vitalflux.com/encoder-decoder-architecture-neural-network/>

Encoder decoder RNNs

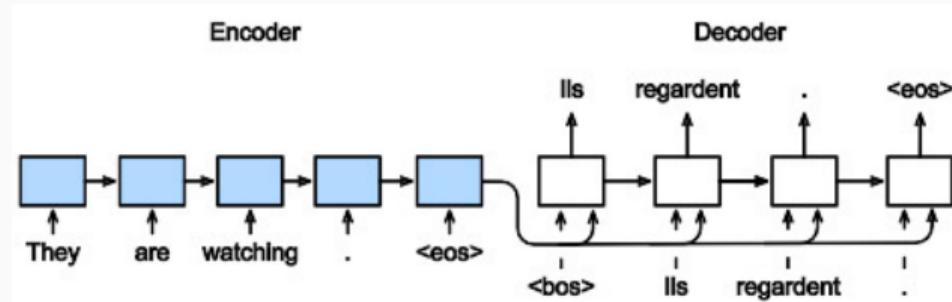
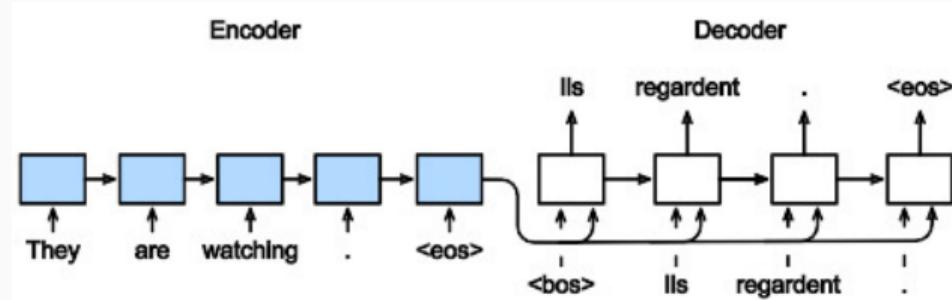


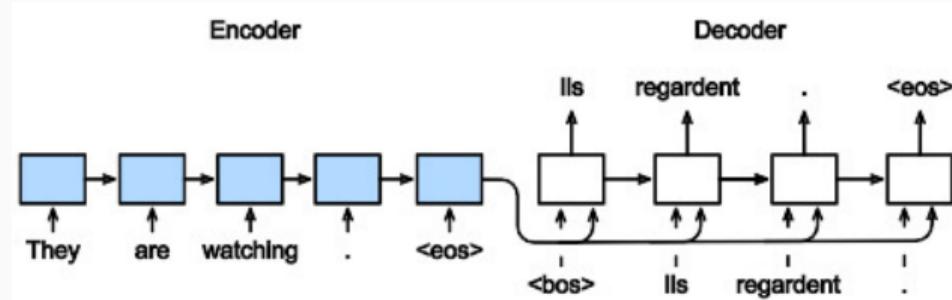
Image: https://d2l.ai/chapter_recurrent-modern/seq2seq.html

Encoder decoder RNNs



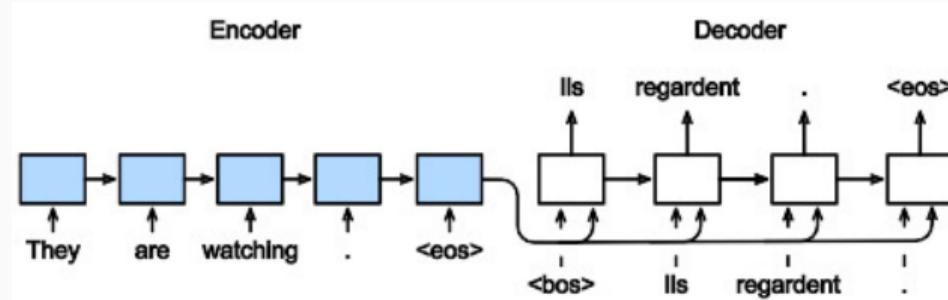
- Earliest application was machine translation

Encoder decoder RNNs



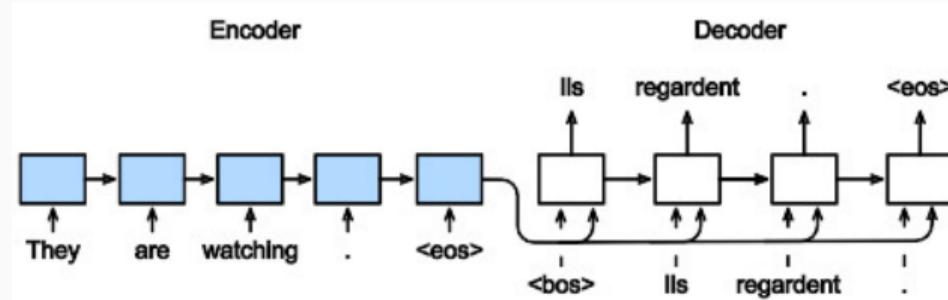
- Earliest application was machine translation
- RNNs with encoding can learn any translations, given parallel corpus

Encoder decoder RNNs



- Earliest application was machine translation
- RNNs with encoding can learn any translations, given parallel corpus
- No manual intervention required

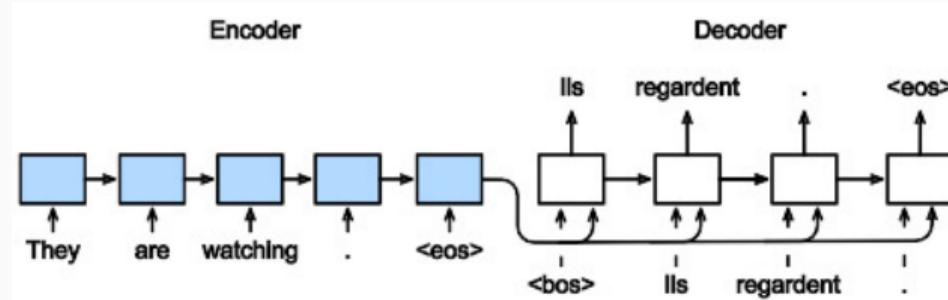
Encoder decoder RNNs



- Earliest application was machine translation
- RNNs with encoding can learn any translations, given parallel corpus
- No manual intervention required
- As good as statistical machine translation with minimal training.

Image: https://d2l.ai/chapter_recurrent-modern/seq2seq.html

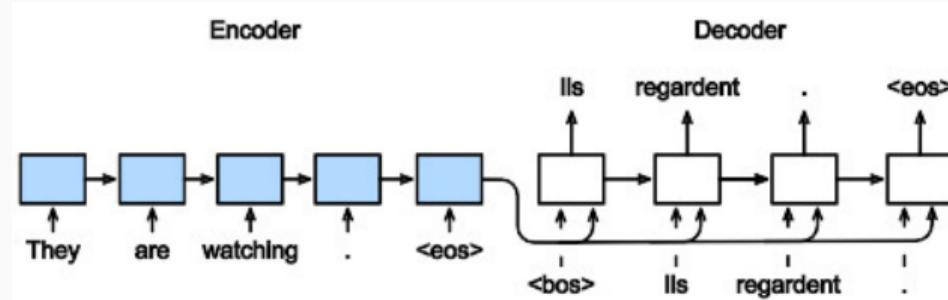
Encoder decoder RNNs



- Earliest application was machine translation
- RNNs with encoding can learn any translations, given parallel corpus
- No manual intervention required
- As good as statistical machine translation with minimal training.
- However,

Image: https://d2l.ai/chapter_recurrent-modern/seq2seq.html

Encoder decoder RNNs



- Earliest application was machine translation
- RNNs with encoding can learn any translations, given parallel corpus
- No manual intervention required
- As good as statistical machine translation with minimal training.
- However, **performance drops for very long sentences**

Image: https://d2l.ai/chapter_recurrent-modern/seq2seq.html

Encoder decoder RNNs with attention

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

Encoder decoder RNNs with attention

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

- To free the network from fixed length internal representation

Encoder decoder RNNs with attention

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

- To free the network from fixed length internal representation
- Uses hidden state from encoder step for each word in output sentence

Encoder decoder RNNs with attention

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

- To free the network from fixed length internal representation
- Uses hidden state from encoder step for each word in output sentence
- Adaptively learns to choose this subset at every decoding step

Encoder decoder RNNs with attention

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

- To free the network from fixed length internal representation
- Uses hidden state from encoder step for each word in output sentence
- Adaptively learns to choose this subset at every decoding step
- Performs better than networks without attention

Encoder decoder RNNs with attention

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

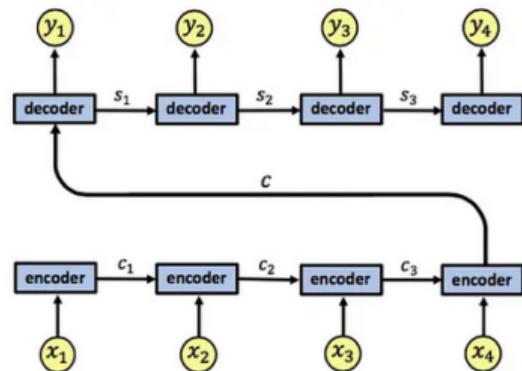
The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

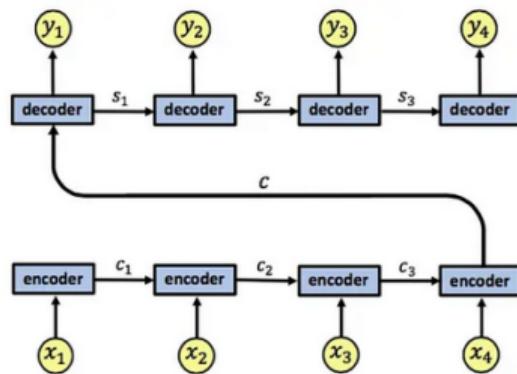
- To free the network from fixed length internal representation
- Uses hidden state from encoder step for each word in output sentence
- Adaptively learns to choose this subset at every decoding step
- Performs better than networks without attention

Attention

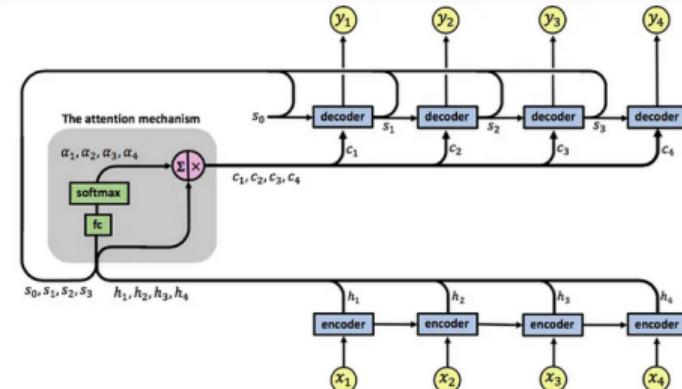


Normal RNNs

Attention



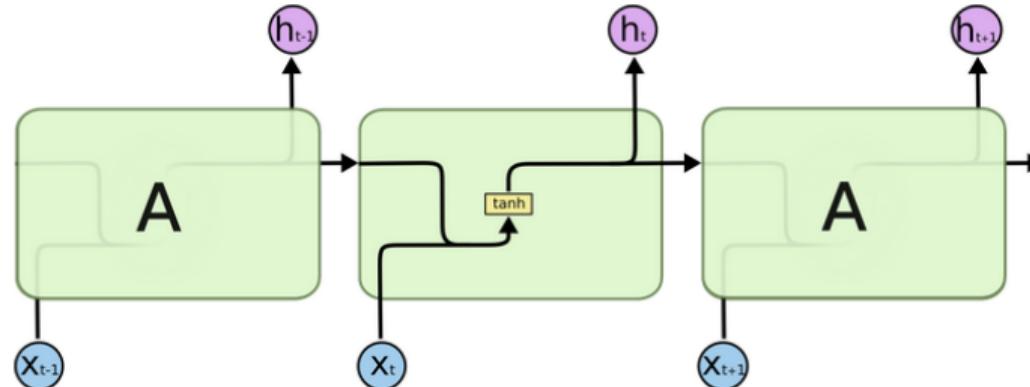
Normal RNNs



RNNs with attention

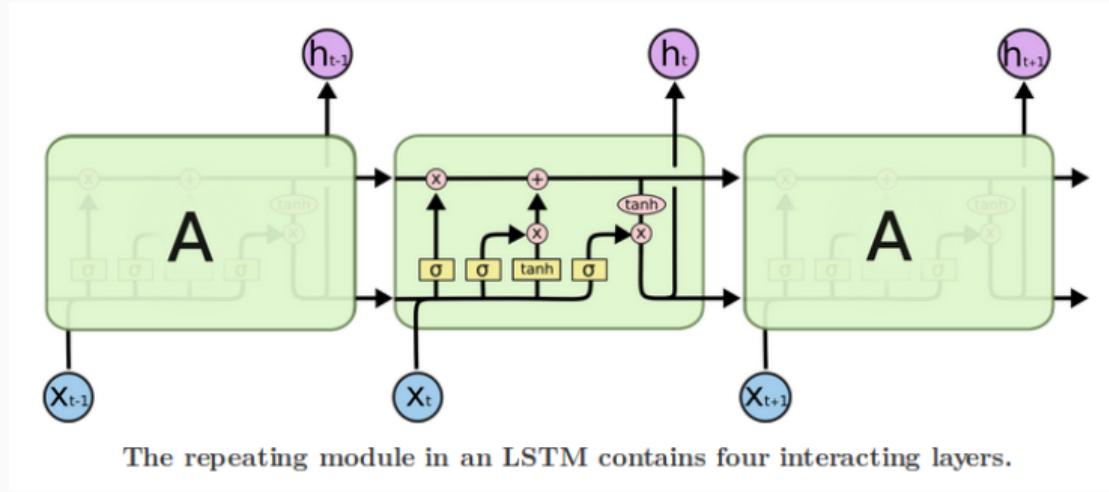
Image: <https://medium.datadrivendev.com/attention-in-rnns-321fbcd64f05>

Simple RNN network

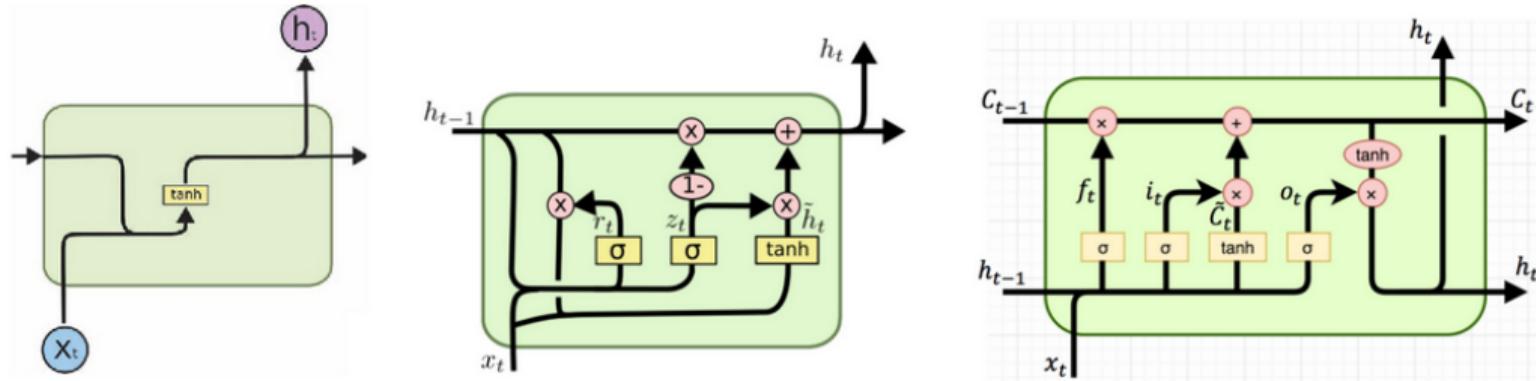


The repeating module in a standard RNN contains a single layer.

LSTM



Basic NLP Models



Basic architectures of RNN, GRU and LSTM cells

However, the training and inference are still word by word (sequential).

Transformers

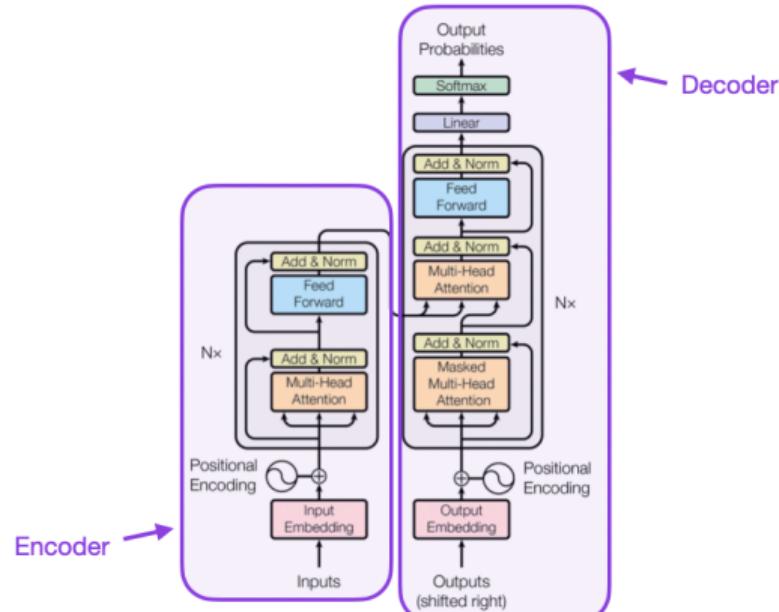


Figure 1: The Transformer - model architecture.

Image: <https://jinglescode.github.io/2020/05/27/illustrated-guide-transformer/>

Transformers

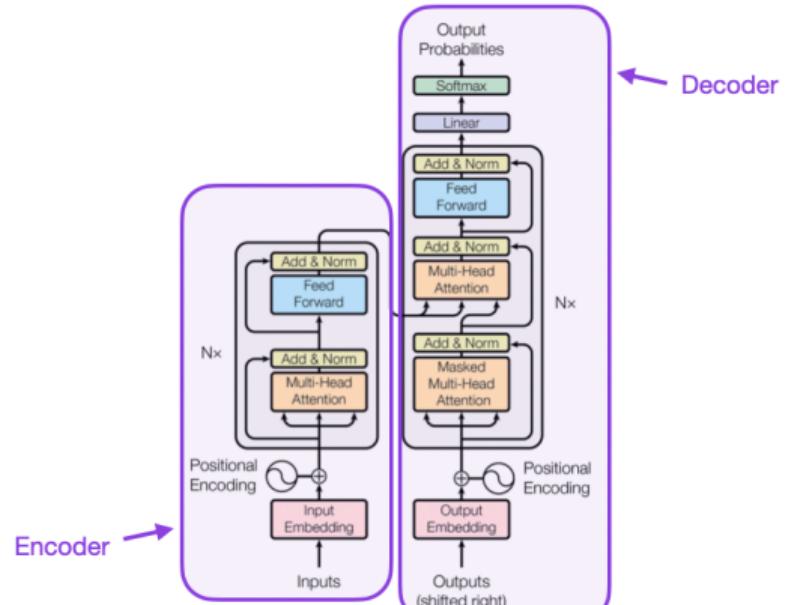


Figure 1: The Transformer - model architecture.

- Transformers are a newer variant

Image: <https://jinglescode.github.io/2020/05/27/illustrated-guide-transformer/>

Transformers

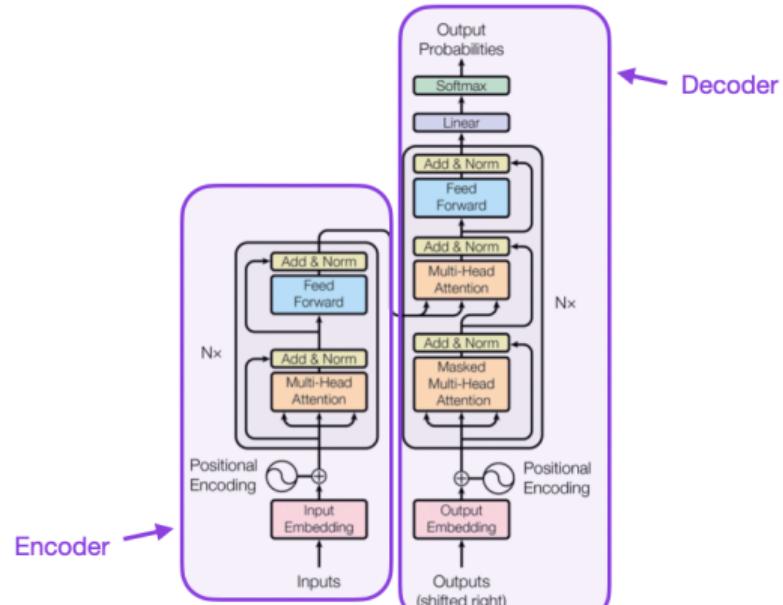


Figure 1: The Transformer - model architecture.

- Transformers are a newer variant
- Perform parallel computations instead of sequential computations

Image: <https://jinglescode.github.io/2020/05/27/illustrated-guide-transformer/>

Transformers

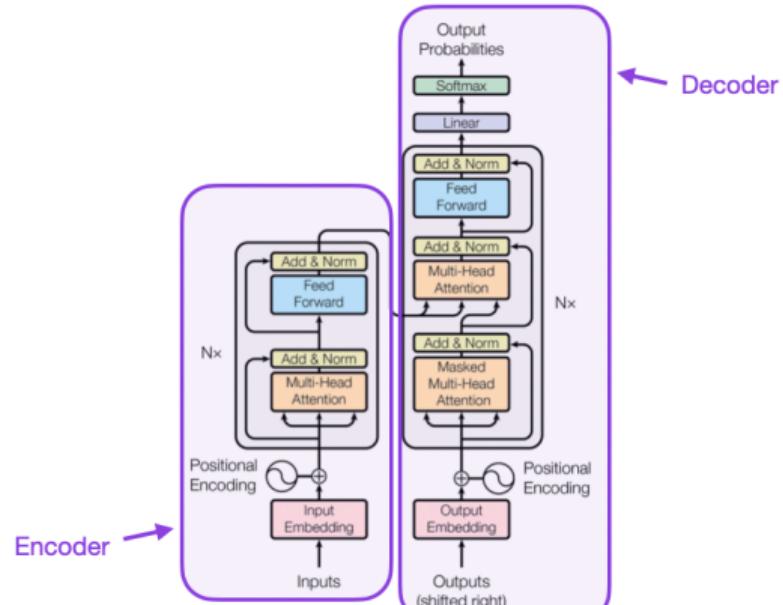


Figure 1: The Transformer - model architecture.

- Transformers are a newer variant
- Perform parallel computations instead of sequential computations
- Faster training and good performance over long sequences

Image: <https://jinglescode.github.io/2020/05/27/illustrated-guide-transformer/>

Transformers

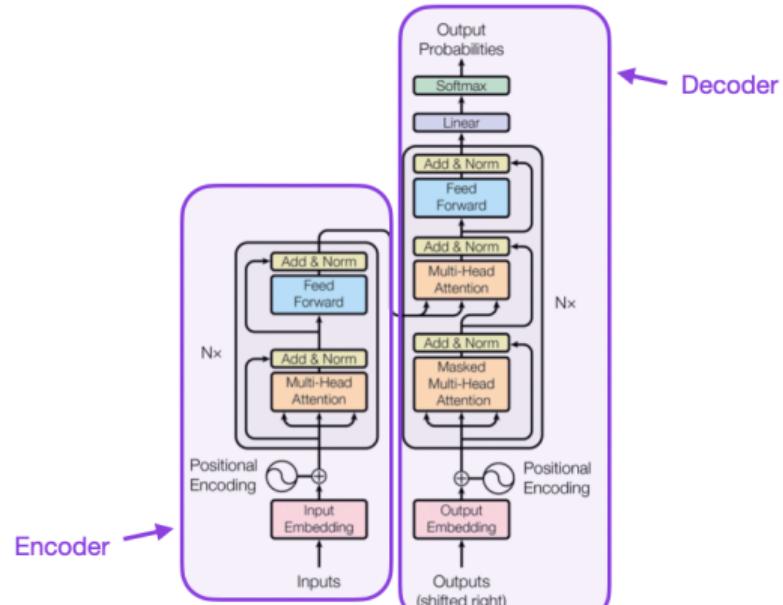


Figure 1: The Transformer - model architecture.

- Transformers are a newer variant
- Perform parallel computations instead of sequential computations
- Faster training and good performance over long sequences
- Maximum token size for responses is fixed during training

Image: <https://jinglescode.github.io/2020/05/27/illustrated-guide-transformer/>

Transformers

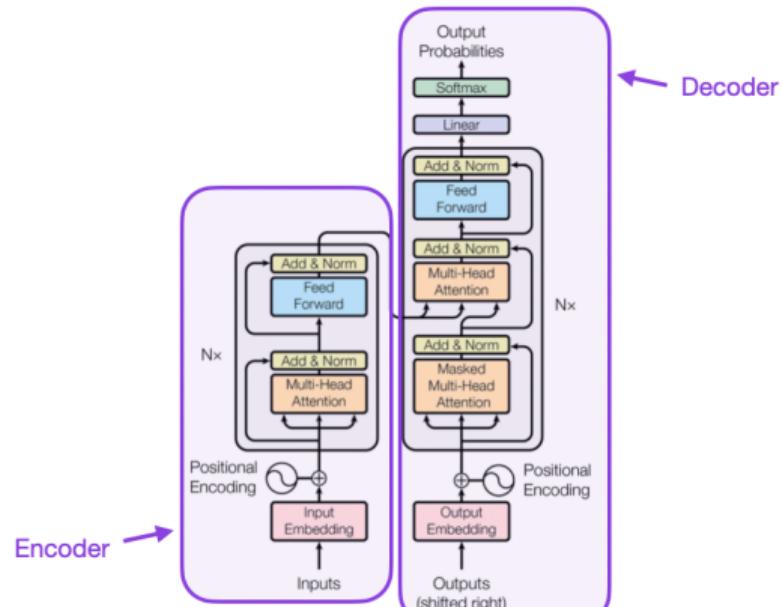


Figure 1: The Transformer - model architecture.

- Transformers are a newer variant
- Perform parallel computations instead of sequential computations
- Faster training and good performance over long sequences
- Maximum token size for responses is fixed during training
- Backbone of all Large Language Models that we see today

Image: <https://jinglescode.github.io/2020/05/27/illustrated-guide-transformer/>

RNN vs. Transformers

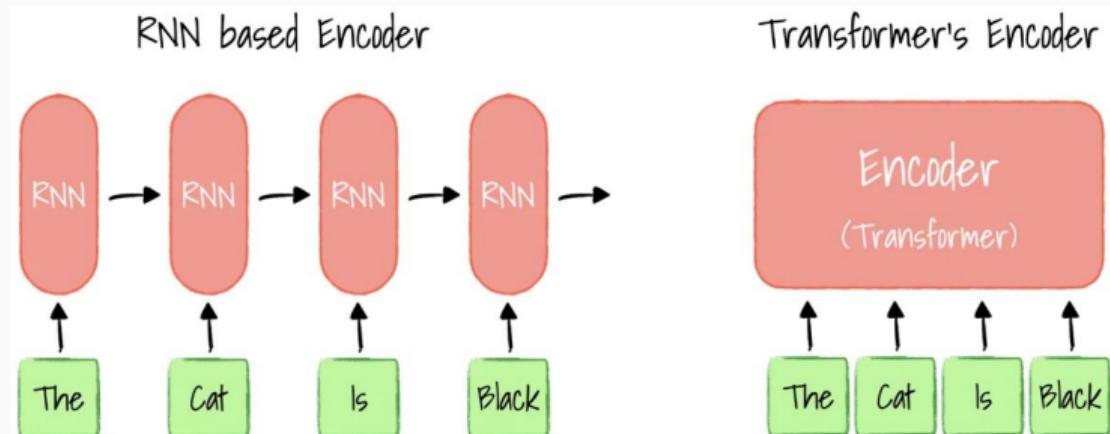


Image: <https://jinglescode.github.io/2020/05/27/illustrated-guide-transformer/>

Large Language Models

Large Language Models



Large Language Models



- Large Language Models have billions of parameters

Large Language Models



- Large Language Models have billions of parameters
- Trained on a wide variety of NLP tasks

Large Language Models



- Large Language Models have billions of parameters
- Trained on a wide variety of NLP tasks
- Auto Encoding models for Sentiment Analysis etc.

Large Language Models



- Large Language Models have billions of parameters
- Trained on a wide variety of NLP tasks
- Auto Encoding models for Sentiment Analysis etc.
- Auto Regressive models for Text Generation

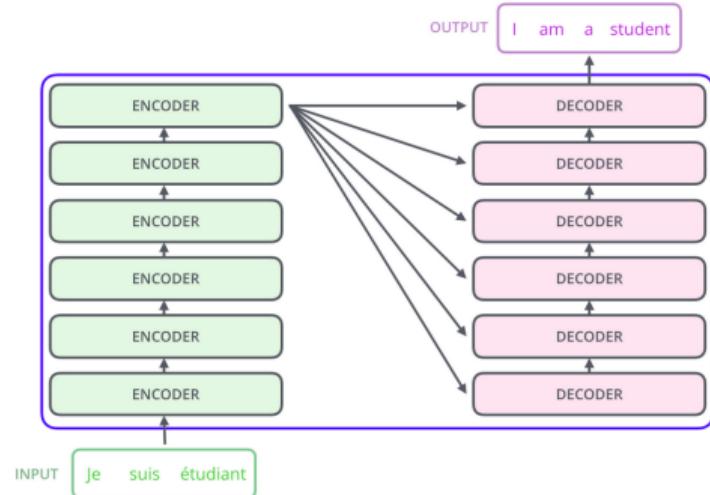
Large Language Models



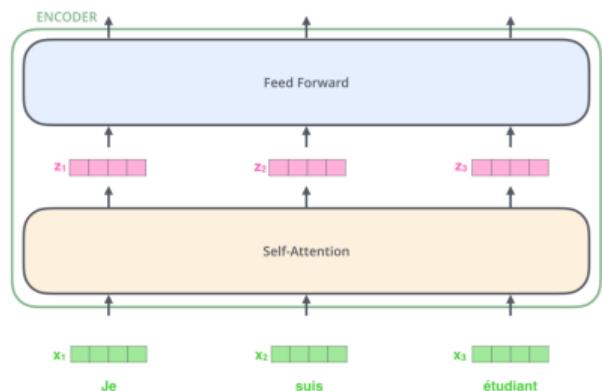
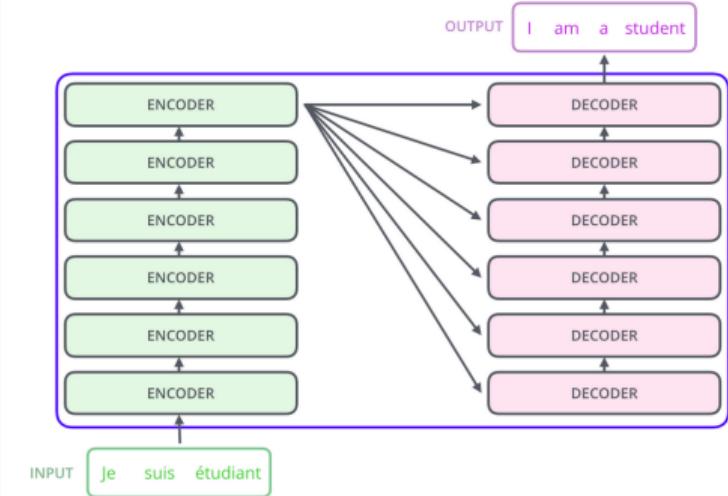
- Large Language Models have billions of parameters
- Trained on a wide variety of NLP tasks
- Auto Encoding models for Sentiment Analysis etc.
- Auto Regressive models for Text Generation
- Sequence to Sequence models for Question Answering

Image: <https://images.app.goo.gl/uLGT8z3WkbpvXxPz9>

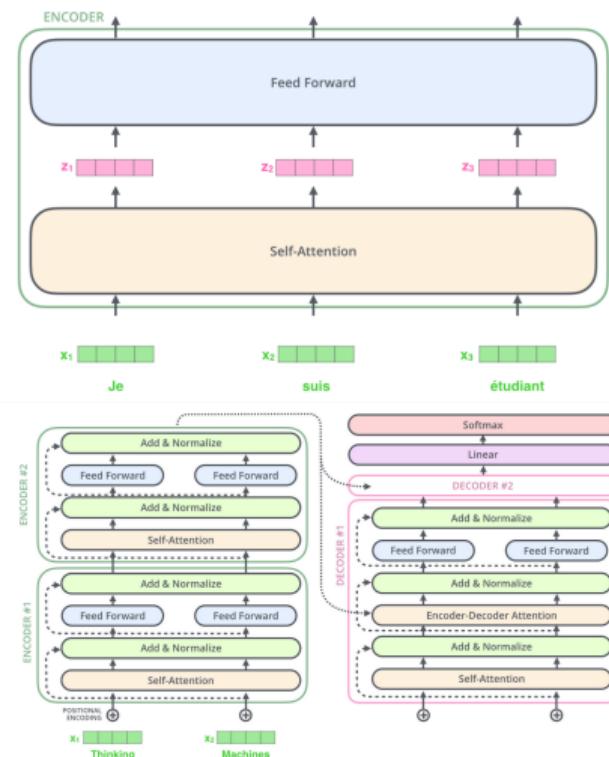
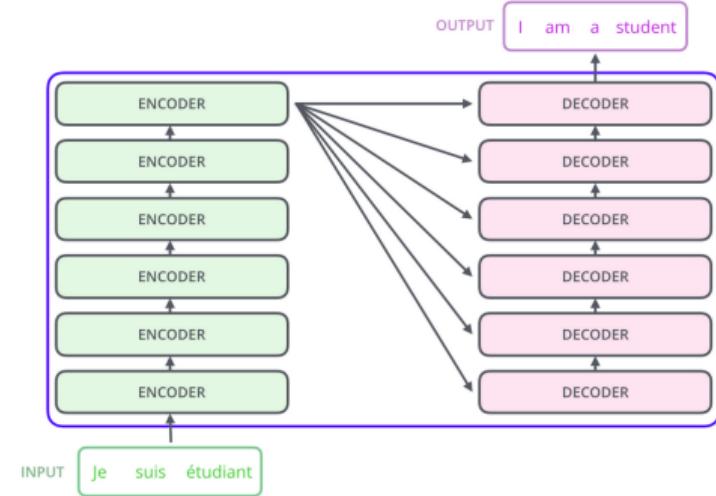
Typical LLM architecture



Typical LLM architecture



Typical LLM architecture



Types of LLM models

- Auto encoding models: In a sentence, one text token in middle is hidden, but tokens on either side are available for prediction

Types of LLM models

- Auto encoding models: In a sentence, one text token in middle is hidden, but tokens on either side are available for prediction
- Auto regressive models: Given k tokens, model is trained to predict the $(k+1)$ th token

Types of LLM models

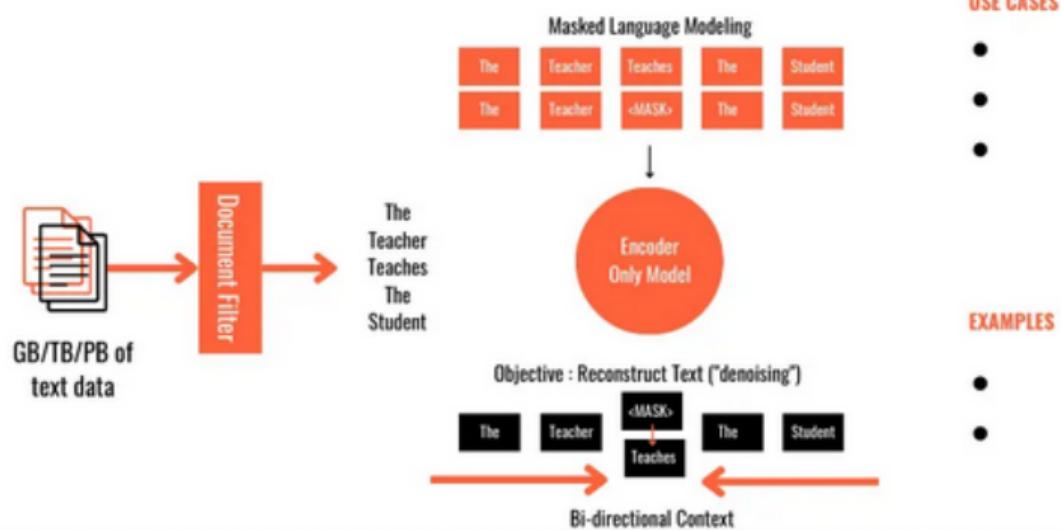
- Auto encoding models: In a sentence, one text token in middle is hidden, but tokens on either side are available for prediction
- Auto regressive models: Given k tokens, model is trained to predict the $(k+1)$ th token
- Sequence to sequence models: Training to predict the missing k tokens in a long string

Types of LLM models

- Auto encoding models: In a sentence, one text token in middle is hidden, but tokens on either side are available for prediction
- Auto regressive models: Given k tokens, model is trained to predict the $(k+1)$ th token
- Sequence to sequence models: Training to predict the missing k tokens in a long string
- Bidirectional vs Unidirectional context

Auto encoding model

AutoEncoding Models (Encoder Only)



USE CASES

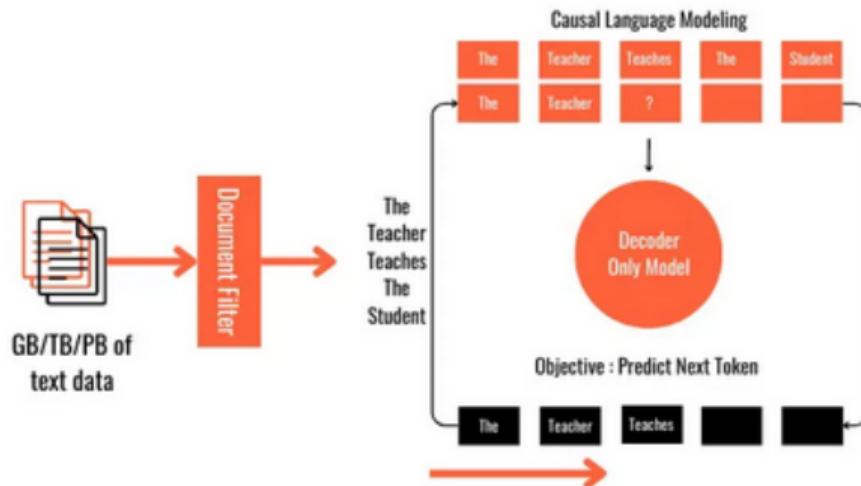
- Sentiment Analysis
- Named Entity Recognition
- Word Classification

EXAMPLES

- BERT
- ROBERTA

Image: <https://medium.com/mlearning-ai/3-llm-architectures-f527ed781ba9>

AutoRegressive Models (Decoder Only)



USE CASES

-

Text Generation

(Most common architecture now and larger models can perform a variety of tasks)

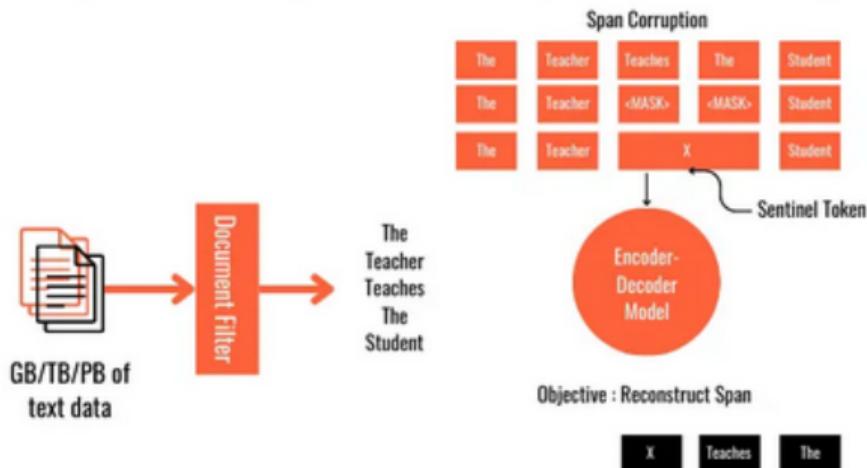
EXAMPLES

-
-

GPT
BLOOM

Sequence to Sequence models

Sequence-to-Sequence Models (Encoder-Decoder)



USE CASES

-
-
-

Translation
Text Summarisation
Question Answering

EXAMPLES

-
-

T5
BART

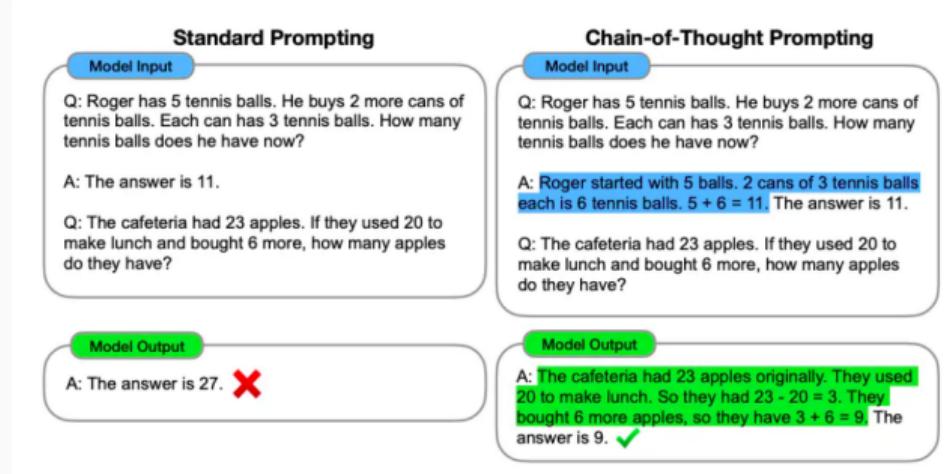
- In context learning, zero shot learning: Model able to answer directly, without illustrative examples

- In context learning, zero shot learning: Model able to answer directly, without illustrative examples
- One shot learning: One example is provided to model as context, and model answers newer questions based on this example

- In context learning, zero shot learning: Model able to answer directly, without illustrative examples
- One shot learning: One example is provided to model as context, and model answers newer questions based on this example
- Few shot learning: Model looks at few examples in context and learns to answer new questions

- In context learning, zero shot learning: Model able to answer directly, without illustrative examples
- One shot learning: One example is provided to model as context, and model answers newer questions based on this example
- Few shot learning: Model looks at few examples in context and learns to answer new questions
- All these do not alter any model weights. In the traditional sense, model is not trained

Prompt Engineering



Designing and optimising prompts for NLP models to respond accurately

Image: <https://images.app.goo.gl/nCGMuhzYwaU38Kzz8>

LLM model fine tuning

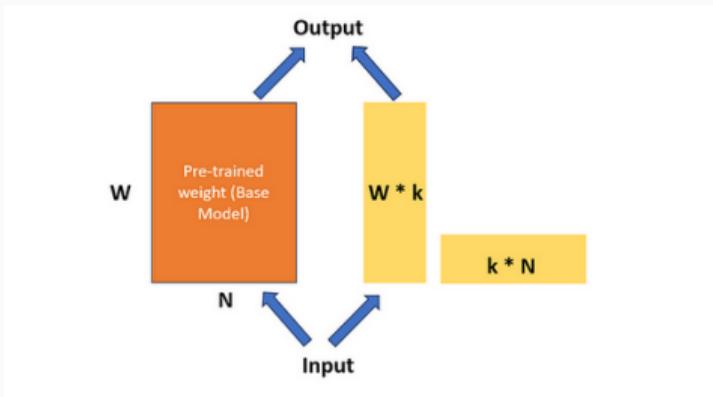


Image: <https://towardsai.net/p/1/parameter-efficient-fine-tuning-peft-inference-and-evaluation-of-llm-model-using-lora>

LLM model fine tuning

- Training an LLM from scratch is a resource intensive task

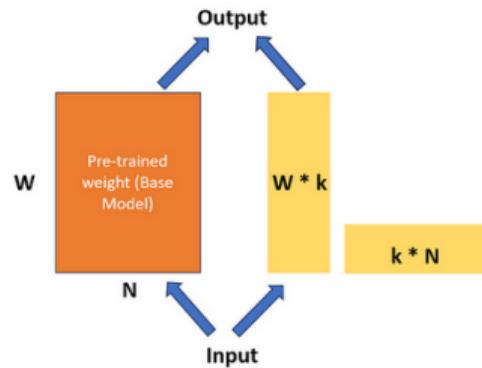
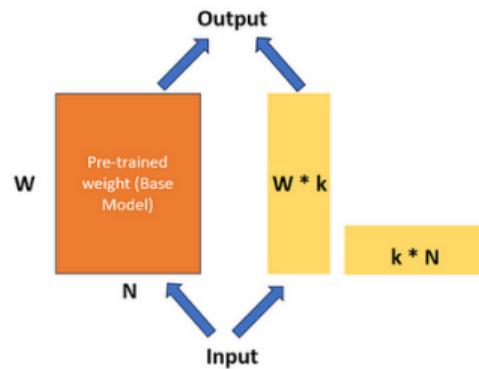


Image: <https://towardsai.net/p/1/parameter-efficient-fine-tuning-peft-inference-and-evaluation-of-llm-model-using-lora>

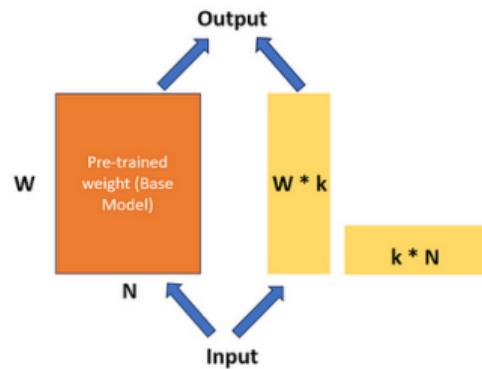
LLM model fine tuning



- Training an LLM from scratch is a resource intensive task
- Fine tuning: Training LLM on specific tasks

Image: <https://towardsai.net/p/1/parameter-efficient-fine-tuning-peft-inference-and-evaluation-of-llm-model-using-lora>

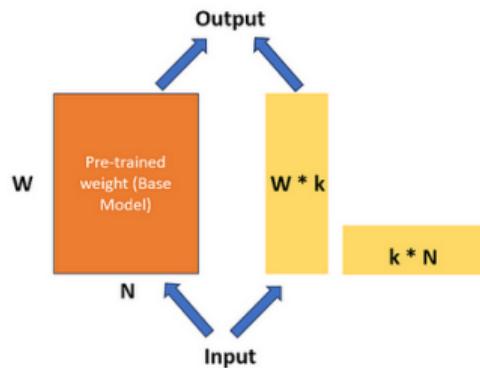
LLM model fine tuning



- Training an LLM from scratch is a resource intensive task
- Fine tuning: Training LLM on specific tasks
- Multi task fine tuning

Image: <https://towardsai.net/p/1/parameter-efficient-fine-tuning-peft-inference-and-evaluation-of-llm-model-using-lora>

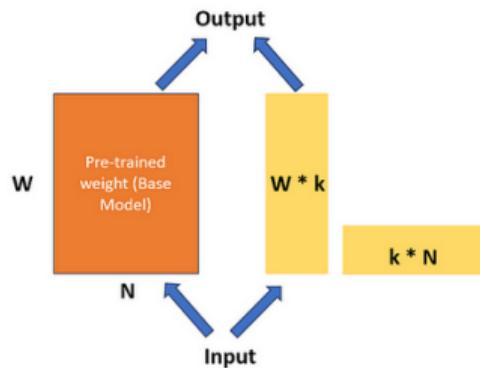
LLM model fine tuning



- Training an LLM from scratch is a resource intensive task
- Fine tuning: Training LLM on specific tasks
- Multi task fine tuning
- PEFT: Parameter efficient fine tuning

Image: <https://towardsai.net/p/1/parameter-efficient-fine-tuning-peft-inference-and-evaluation-of-llm-model-using-lora>

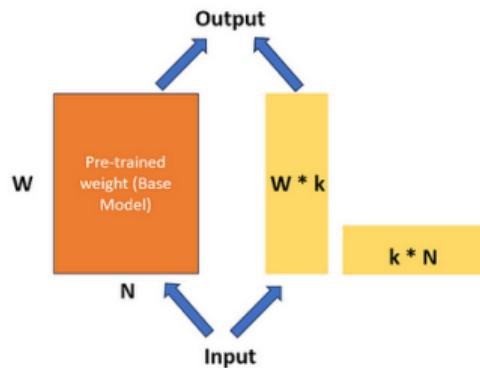
LLM model fine tuning



- Training an LLM from scratch is a resource intensive task
- Fine tuning: Training LLM on specific tasks
- Multi task fine tuning
- PEFT: Parameter efficient fine tuning
- LoRA: Low Rank Adaptation of LLMs

Image: <https://towardsai.net/p/1/parameter-efficient-fine-tuning-peft-inference-and-evaluation-of-llm-model-using-lora>

LLM model fine tuning



- Training an LLM from scratch is a resource intensive task
- Fine tuning: Training LLM on specific tasks
- Multi task fine tuning
- PEFT: Parameter efficient fine tuning
- LoRA: Low Rank Adaptation of LLMs
- **Prompt Tuning**

Image: <https://towardsai.net/p/1/parameter-efficient-fine-tuning-peft-inference-and-evaluation-of-llm-model-using-lora>

LLM model evaluation

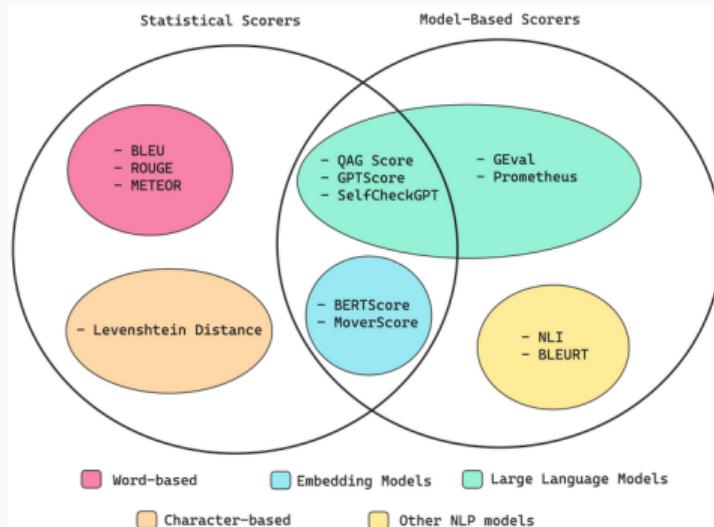
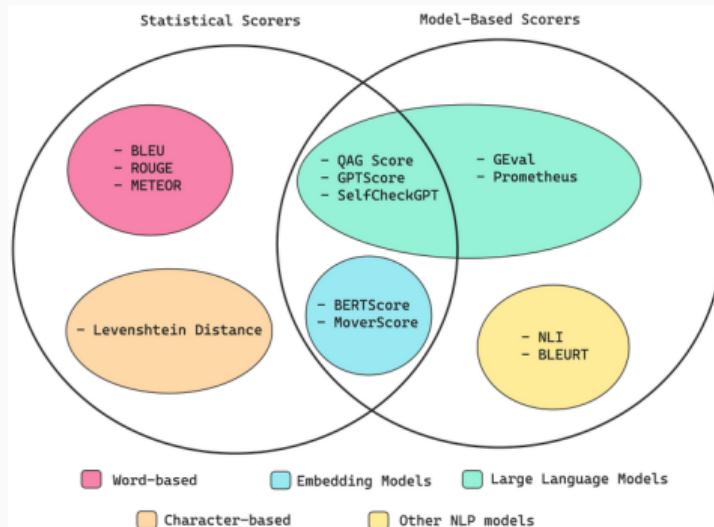


Image: <https://www.confident-ai.com/blog/llm-evaluation-metrics-everything-you-need-for-llm-evaluation>

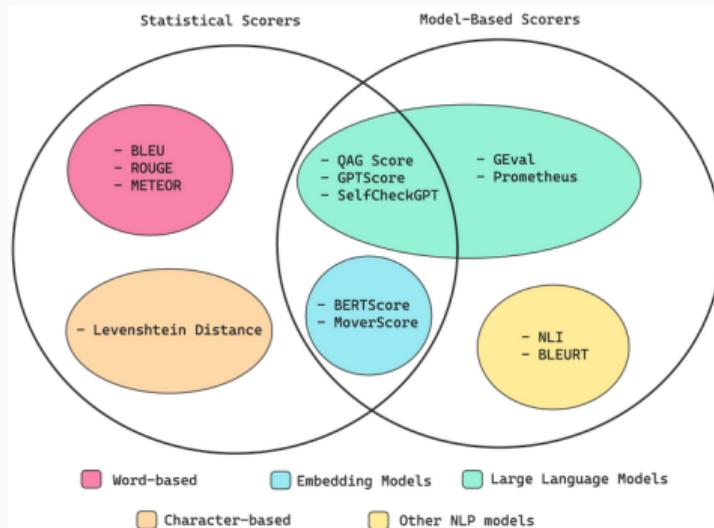
LLM model evaluation



- Relevancy - Context and Answer

Image: <https://www.confident-ai.com/blog/llm-evaluation-metrics-everything-you-need-for-llm-evaluation>

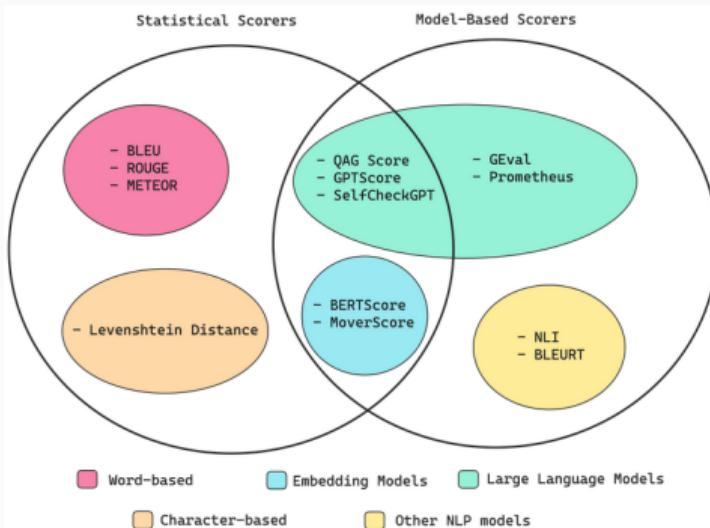
LLM model evaluation



- Relevancy - Context and Answer
- Hallucination

Image: <https://www.confident-ai.com/blog/llm-evaluation-metrics-everything-you-need-for-llm-evaluation>

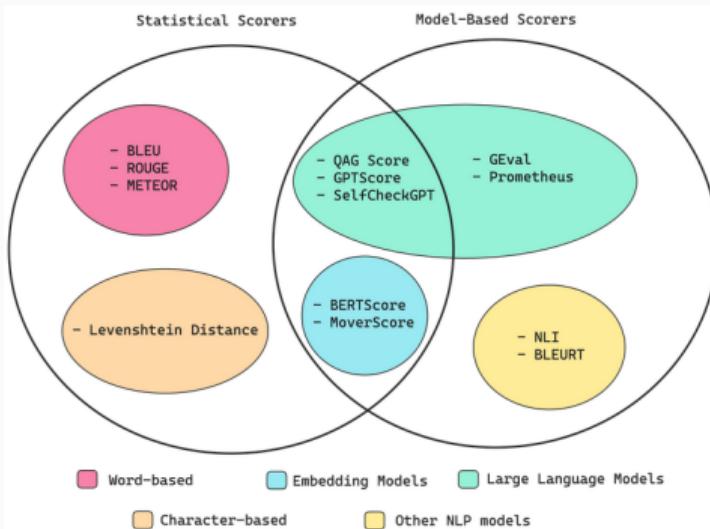
LLM model evaluation



- Relevancy - Context and Answer
- Hallucination
- Correctness and Bias

Image: <https://www.confident-ai.com/blog/llm-evaluation-metrics-everything-you-need-for-llm-evaluation>

LLM model evaluation



- Relevancy - Context and Answer
- Hallucination
- Correctness and Bias
- Metrics - Task specific

Image: <https://www.confident-ai.com/blog/llm-evaluation-metrics-everything-you-need-for-llm-evaluation>

Inference

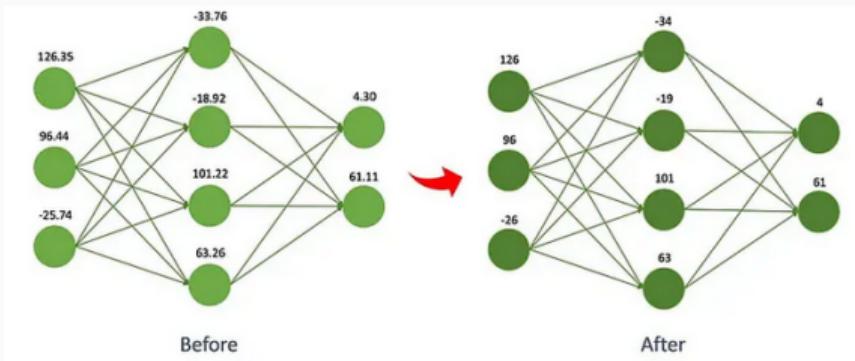
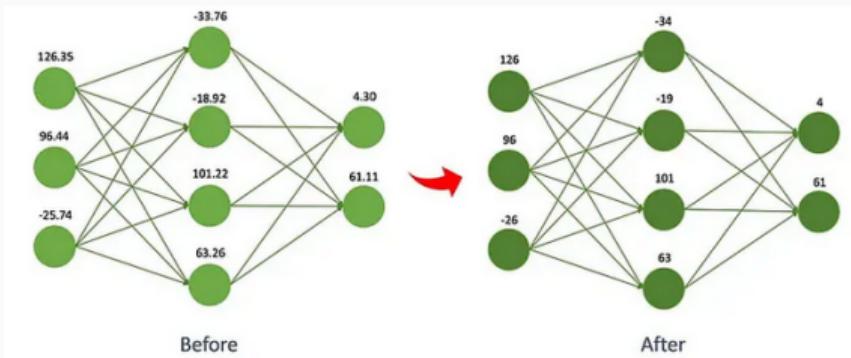


Image: <https://blog.gopenai.com/the-llm-revolution-boosting-computing-capacity-with-quantization-methods-b8666cdb4b6a>

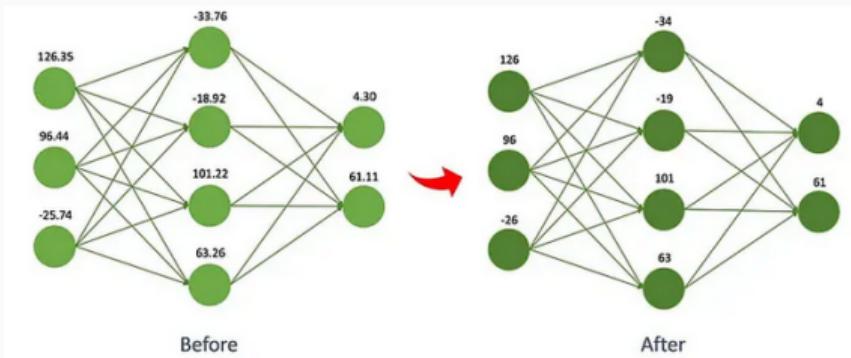
Inference



- Quantization of model weights

Image: <https://blog.gopenai.com/the-llm-revolution-boosting-computing-capacity-with-quantization-methods-b8666cdb4b6a>

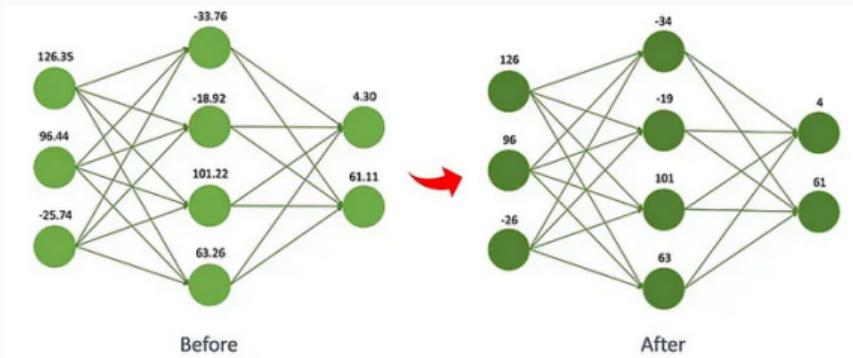
Inference



- Quantization of model weights
- Training parallelization

Image: <https://blog.gopenai.com/the-llm-revolution-boosting-computing-capacity-with-quantization-methods-b8666cdb4b6a>

Inference



- Quantization of model weights
- Training parallelization
- QLoRA: Quantized Low Rank Adaptation

Image: <https://blog.gopenai.com/the-llm-revolution-boosting-computing-capacity-with-quantization-methods-b8666cdb4b6a>

NLP Applications

Sentiment Analysis



My experience so far has been fantastic!

POSITIVE



The product is ok I guess

NEUTRAL

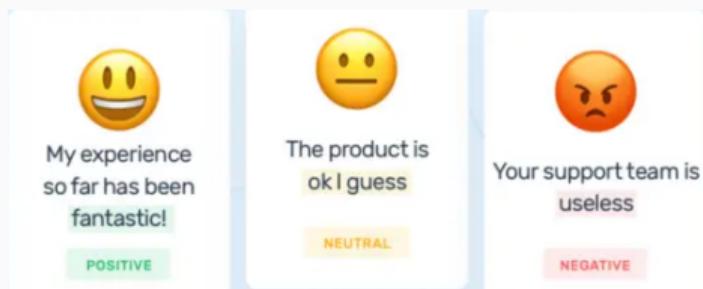


Your support team is useless

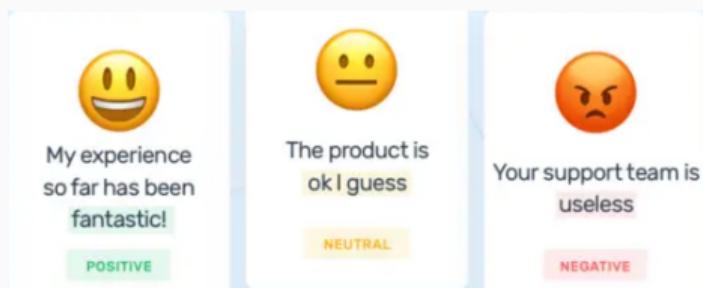
NEGATIVE

Sentiment Analysis

- Uses any one of embedding methods to convert sentence to a vector

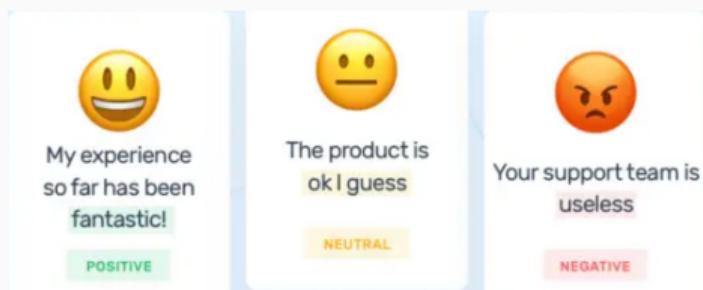


Sentiment Analysis



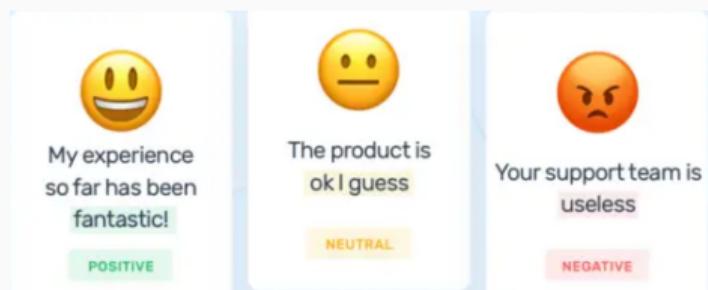
- Uses any one of embedding methods to convert sentence to a vector
- Trains a one class classifier on the resultant vector to predict sentiment

Sentiment Analysis



- Uses any one of embedding methods to convert sentence to a vector
- Trains a one class classifier on the resultant vector to predict sentiment
- Better embeddings lead to improved predictions for the same SVM model

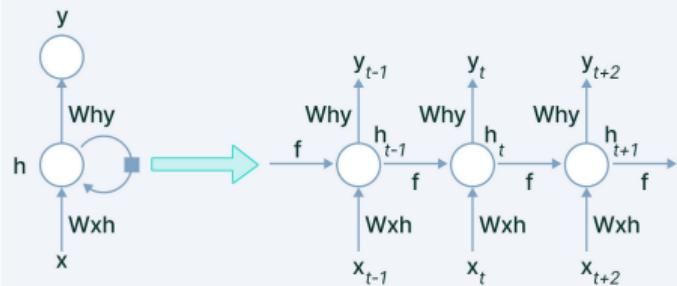
Sentiment Analysis



- Uses any one of embedding methods to convert sentence to a vector
- Trains a one class classifier on the resultant vector to predict sentiment
- Better embeddings lead to improved predictions for the same SVM model
- Denser embeddings give better answer than sparse embeddings

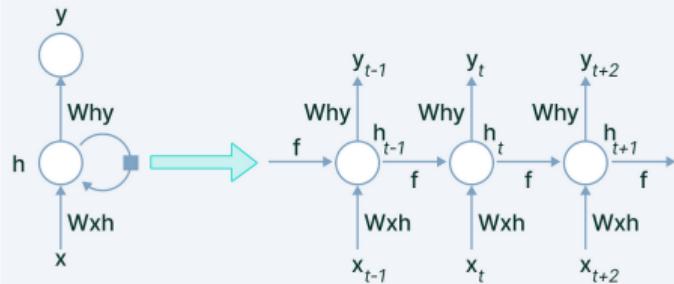
Machine Translation

The Recurrent Neural Networks (RNN)



Machine Translation

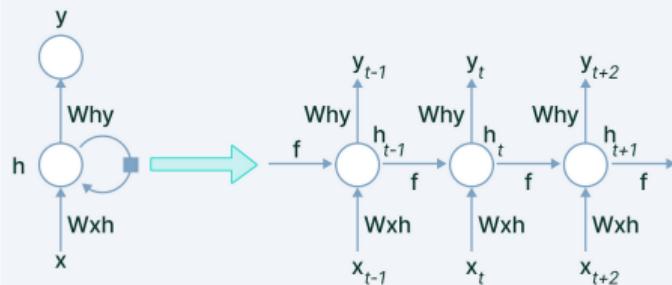
The Recurrent Neural Networks (RNN)



- Converts a string of arbitrary length to a fixed length sequence of numbers, called context vector

Machine Translation

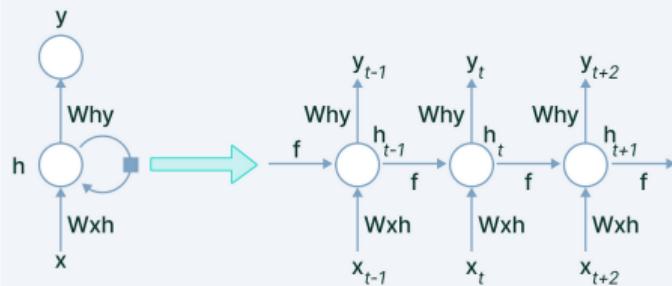
The Recurrent Neural Networks (RNN)



- Converts a string of arbitrary length to a fixed length sequence of numbers, called context vector
- Once encoded, different systems can be used to decode to different languages

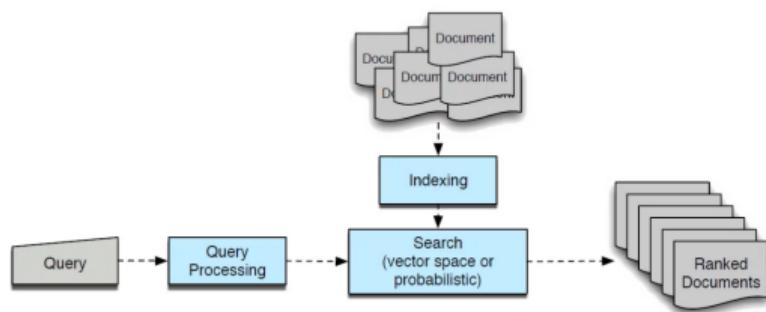
Machine Translation

The Recurrent Neural Networks (RNN)



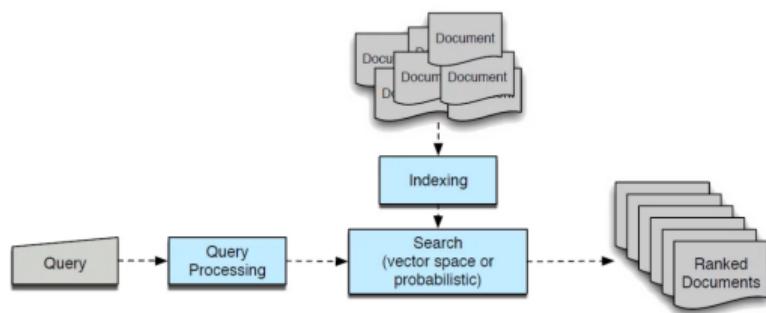
- Converts a string of arbitrary length to a fixed length sequence of numbers, called context vector
- Once encoded, different systems can be used to decode to different languages
- Decoder generates an output string of variable length in the target language

Information Retrieval

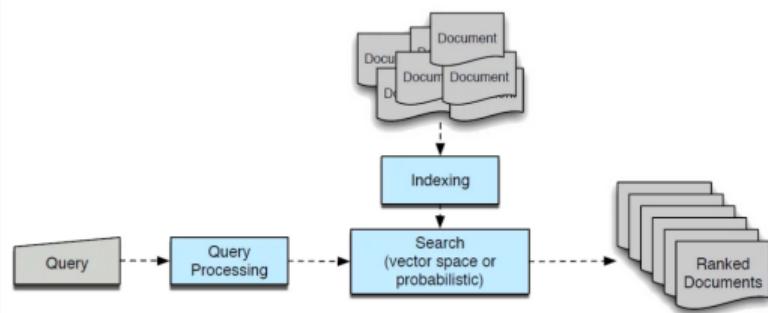


Information Retrieval

- Corpus of documents are processed to be represented as vectors

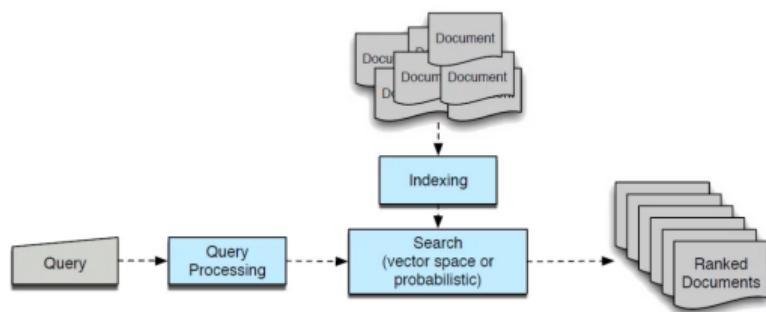


Information Retrieval

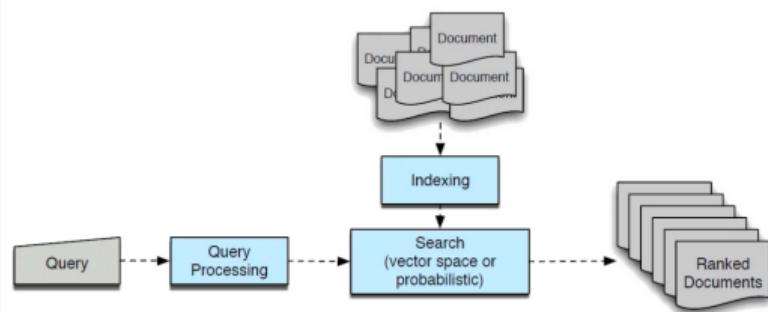


- Corpus of documents are processed to be represented as vectors
- Stored in a vector data base

Information Retrieval



- Corpus of documents are processed to be represented as vectors
- Stored in a vector data base
- User query is converted into vector using same processing



- Corpus of documents are processed to be represented as vectors
- Stored in a vector data base
- User query is converted into vector using same processing
- Vector search to find nearest documents are retrieve them

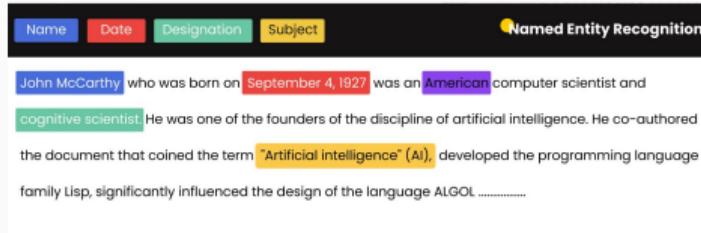
Information Extraction

Name Date Designation Subject Named Entity Recognition

John McCarthy, who was born on September 4, 1927, was an American computer scientist and cognitive scientist. He was one of the founders of the discipline of artificial intelligence. He co-authored the document that coined the term "Artificial intelligence" (AI), developed the programming language family Lisp, significantly influenced the design of the language ALGOL

Information Extraction

- Extracting structured information from unstructured text



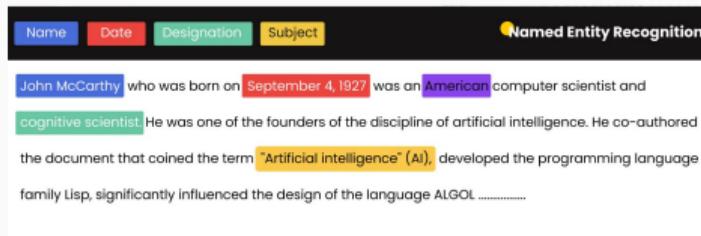
Information Extraction

Name Date Designation Subject **Named Entity Recognition**

John McCarthy, who was born on September 4, 1927, was an American computer scientist and cognitive scientist. He was one of the founders of the discipline of artificial intelligence. He co-authored the document that coined the term "Artificial intelligence" (AI), developed the programming language family Lisp, significantly influenced the design of the language ALGOL

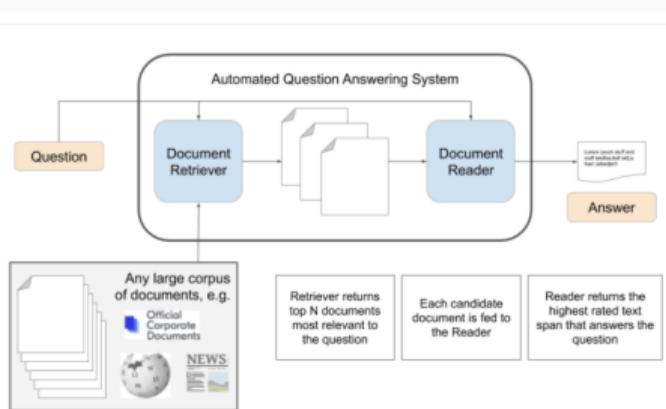
- Extracting structured information from unstructured text
- Uses techniques like Named Entity Recognition (NER)

Information Extraction



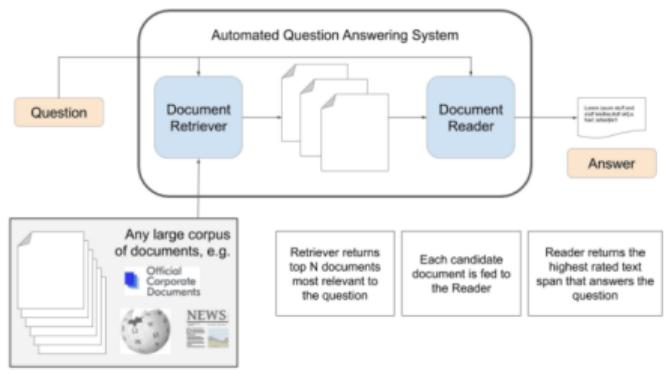
- Extracting structured information from unstructured text
- Uses techniques like Named Entity Recognition (NER)
- Extracted data stored in data base for analysis and insights

Question Answering / Retrieval Augmented Generation

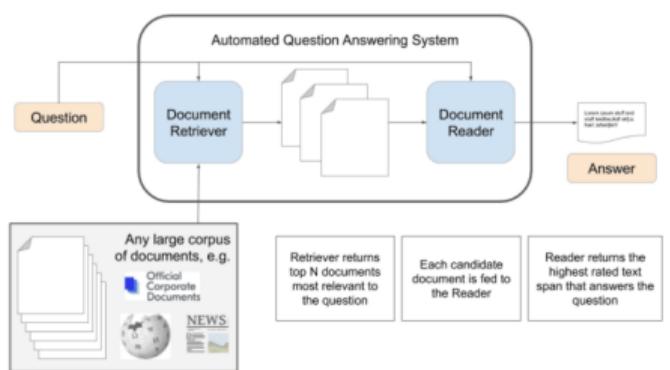


Question Answering / Retrieval Augmented Generation

- A corpus of documents is processed and indexed

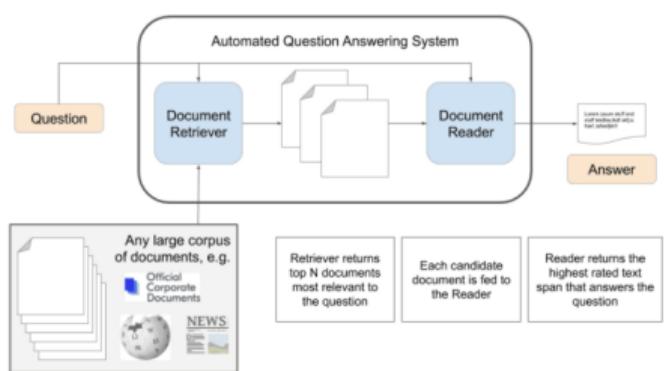


Question Answering / Retrieval Augmented Generation



- A corpus of documents is processed and indexed
- User query is processed to find nearest document and chunks of text closest to query

Question Answering / Retrieval Augmented Generation



- A corpus of documents is processed and indexed
- User query is processed to find nearest document and chunks of text closest to query
- LLM / NLP model generates response to the question based on identified chunk of text

LLM Applications

Question:

What is the nucleus of a cell and what function does it perform in living organisms?



Cell Biology

Difficulty Easy

76% Confidence

Answer:

A nucleus is the central core of an atom, containing protons and neutrons, which are held together by the strong nuclear force. An array of holes, or pores, in the nuclear membrane allows for the selective passage of certain molecules (such as proteins and nucleic acids) into and out of the nucleus.

[Report Answer](#)

Long Explanation

A nucleus, as related to [genomics](#), is the membrane-enclosed organelle within a [cell](#) that contains the [chromosomes](#). An array of holes, or pores, in the nuclear membrane allows for the selective passage of certain molecules (such as proteins and nucleic acids) into and out of nucleus.

The nucleus is one of the most obvious parts of the cell when you look at a picture of the cell. It's in the middle of the cell, and the nucleus contains all of the cell's chromosomes, which encode the genetic material. So this is really an important part of the cell to protect. The nucleus has a membrane around it that keeps all the chromosomes inside and makes the distinction between the chromosomes being inside the nucleus and the other organelles and components of the cell staying outside.

Question Answering

Image: <https://towardsdatascience.com/rag-vs-finetuning-which-is-the-best-tool-to-boost-your-llm-application-94654bieaba7>

LLM Applications

Question:

What is the nucleus of a cell and what function does it perform in living organisms?

Cell Biology

Difficulty: Easy



76% Confidence

Answer:

A nucleus is the central core of an atom, containing protons and neutrons, which are held together by the strong nuclear force. An array of holes, or pores, in the nuclear membrane allows for the selective passage of certain molecules (such as proteins and nucleic acids) into and out of the nucleus.

[Report Answer](#)

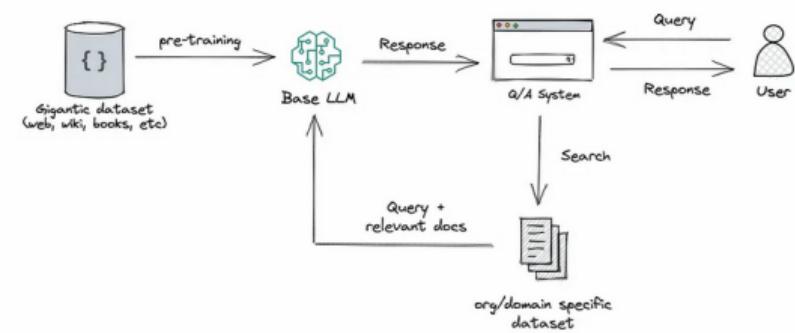
Long Explanation

A nucleus, as related to [genomics](#), is the membrane-enclosed organelle within a [cell](#) that contains the [chromosomes](#). An array of holes, or pores, in the nuclear membrane allows for the selective passage of certain molecules (such as proteins and nucleic acids) into and out of nucleus.

The nucleus is one of the most obvious parts of the cell when you look at a picture of the cell. It's in the middle of the cell, and the nucleus contains all of the cell's chromosomes, which encode the genetic material. So this is really an important part of the cell to protect. The nucleus has a membrane around it that keeps all the chromosomes inside and makes the distinction between the chromosomes being inside the nucleus and the other organelles and components of the cell staying outside.

Question Answering

Image: <https://towardsdatascience.com/rag-vs-finetuning-which-is-the-best-tool-to-boost-your-llm-application-94654bieaba7>



Intelligent Chatbots

LLM Applications

K Generate a Python function that takes a product review and its corresponding sentiment as inputs, and appends these as a new row to a specified CSV file.

Q Sure, I can provide a Python function for that. We will use Python's built-in `csv` library to read and write CSV files. The function will append a new row to the CSV file with the review and its sentiment.

Here's the function:

```
python
Copy code

import csv

def append_review_to_csv(filename, review, sentiment):
    # Open the file in append mode
    with open(filename, 'a', newline='', encoding='utf-8') as csvfile:
        # Create a writer object
        writer = csv.writer(csvfile)
        # Write the review and sentiment as a new row
        writer.writerow([review, sentiment])
```

Code Generation

Image: <https://towardsdatascience.com/rag-vs-finetuning-which-is-the-best-tool-to-boost-your-llm-application-94654bieaba7>

LLM Applications

K Generate a Python function that takes a product review and its corresponding sentiment as inputs, and appends these as a new row to a specified CSV file.

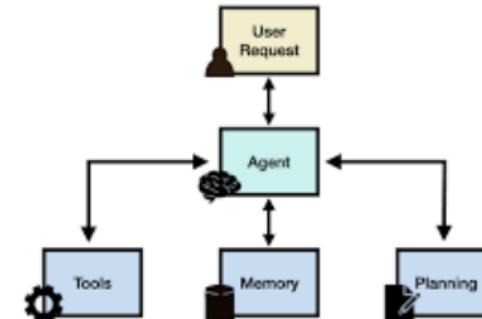
Q Sure, I can provide a Python function for that. We will use Python's built-in 'csv' library to read and write CSV files. The function will append a new row to the CSV file with the review and its sentiment.

Here's the function:

```
python
import csv

def append_review_to_csv(filename, review, sentiment):
    # Open the file in append mode
    with open(filename, 'a', newline='', encoding='utf-8') as csvfile:
        # Create a writer object
        writer = csv.writer(csvfile)
        # Write the review and sentiment as a new row
        writer.writerow([review, sentiment])
```

Code Generation



Agent based applications

Image: <https://towardsdatascience.com/rag-vs-finetuning-which-is-the-best-tool-to-boost-your-llm-application-94654bieaba7>

- Lot of work going on in the LLMs for Indian language space
- Sarvam.ai - <https://www.sarvam.ai/>
- Bharath GPT - <https://bharatgpt.ai/>
- Krutrim - <https://www.olakrutrim.com/>
- And many more

Summary

- Variety of NLP methods - RNNs to LLMs

Summary

- Variety of NLP methods - RNNs to LLMs
- Each method has its abilities and computational requirements

Summary

- Variety of NLP methods - RNNs to LLMs
- Each method has its abilities and computational requirements
- For high volume usage, LLMs may be an overkill for most of the tasks

Summary

- Variety of NLP methods - RNNs to LLMs
- Each method has its abilities and computational requirements
- For high volume usage, LLMs may be an overkill for most of the tasks
- Active areas of research

Summary

- Variety of NLP methods - RNNs to LLMs
- Each method has its abilities and computational requirements
- For high volume usage, LLMs may be an overkill for most of the tasks
- Active areas of research
 - Edge deployment (Mobile/Laptop)

Summary

- Variety of NLP methods - RNNs to LLMs
- Each method has its abilities and computational requirements
- For high volume usage, LLMs may be an overkill for most of the tasks
- Active areas of research
 - Edge deployment (Mobile/Laptop)
 - Indic Languages

Summary

- Variety of NLP methods - RNNs to LLMs
- Each method has its abilities and computational requirements
- For high volume usage, LLMs may be an overkill for most of the tasks
- Active areas of research
 - Edge deployment (Mobile/Laptop)
 - Indic Languages
 - Application specific - Law, Agriculture

Thank you.