EEP 596: Adv Intro ML || Lecture 17

Dr. Karthik Mohan

Univ. of Washington, Seattle

March 2, 2023

Last Time

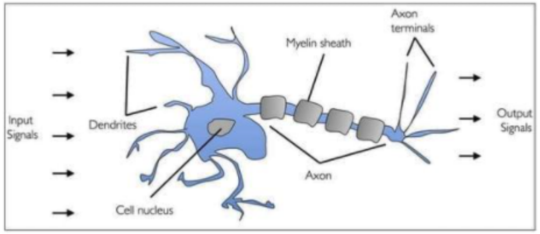
a. Applications in NLP
b. State of the art models in NLP → Transformers

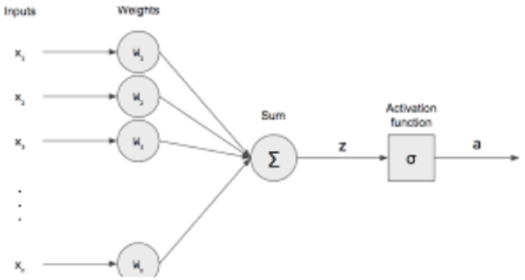
Today

- Attention and Transformers
- Transformer Demo

Evolution of DNN architectures for NLP!

Perceptron

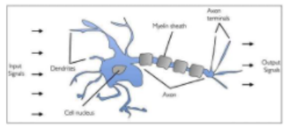


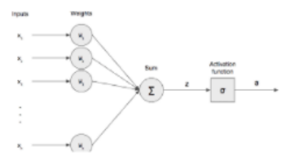


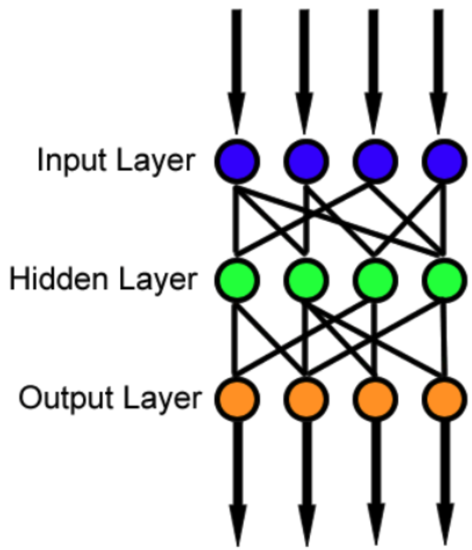
Evolution of DNN architectures for NLP!

Perceptron

Feed Forward NN

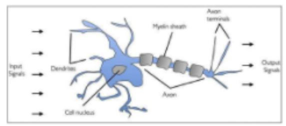


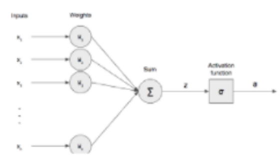




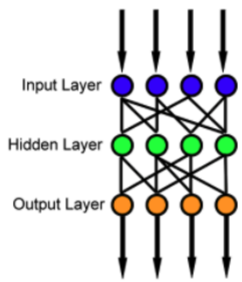
Evolution of DNN architectures for NLP!

Perceptron





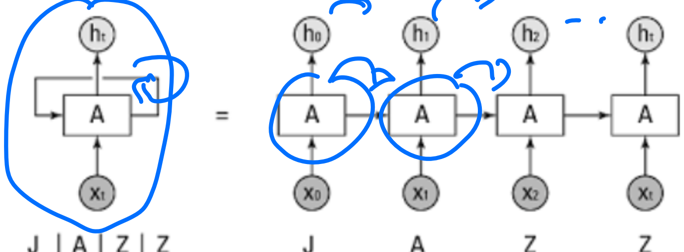
Feed Forward NN



chaining FFNs

RNN

W H



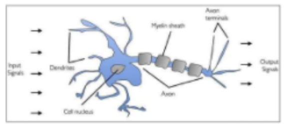
Sequence

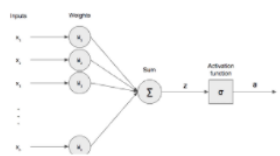
Language

- Vanishing Grad
- Explod Grad

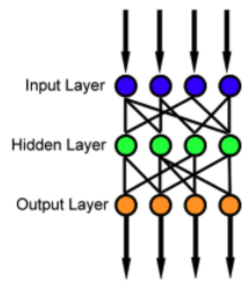
Evolution of DNN architectures for NLP!

Perceptron

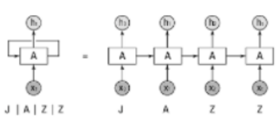




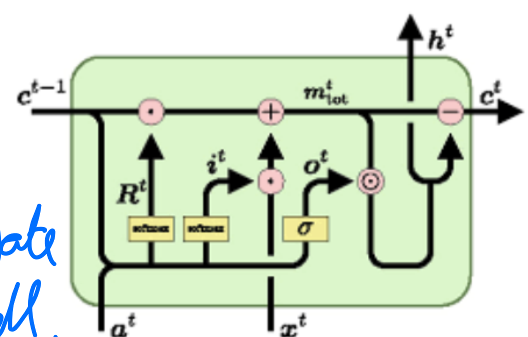
Feed Forward NN



RNN



LSTM

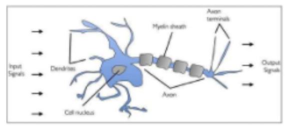


c^t – Long term Cell State

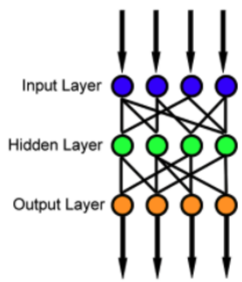
h^t – Short term Cell State

Evolution of DNN architectures for NLP!

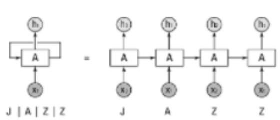
Perceptron



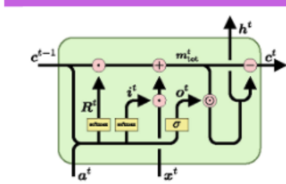
Feed Forward NN



RNN



LSTM



LSTM with Attention

Input: question
Output: Answer

Input:- Documents
Output:- Summary

Input tokens (Do you like NLP?)

Attention

Output:- Yes I do like NLP

Evolution of DNN architectures for NLP!

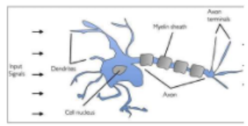
Sentence 1:- "I like this place. It's really cool" → "coolness factor"

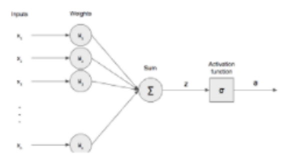
Sentence 2:- "Can you turn on the heat. It's really cool here" → temperature

Contextual Embeddings

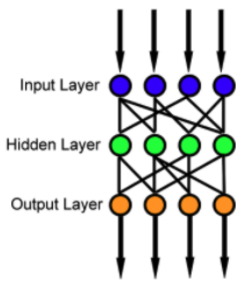
LLMs → Large Language Models

Perceptron





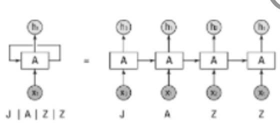
Feed Forward NN



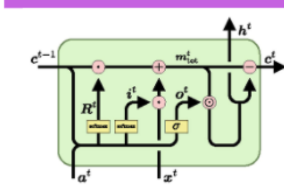
RNN

Hallucination Issues

"I like like this place place"



LSTM



LSTM with Attention

Transformer

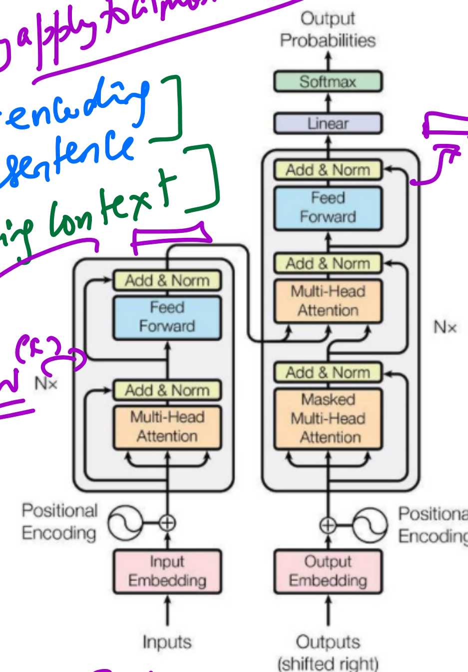
c) They apply to almost every NLP task:

a) Better encoding of sentence

b) Preserving context

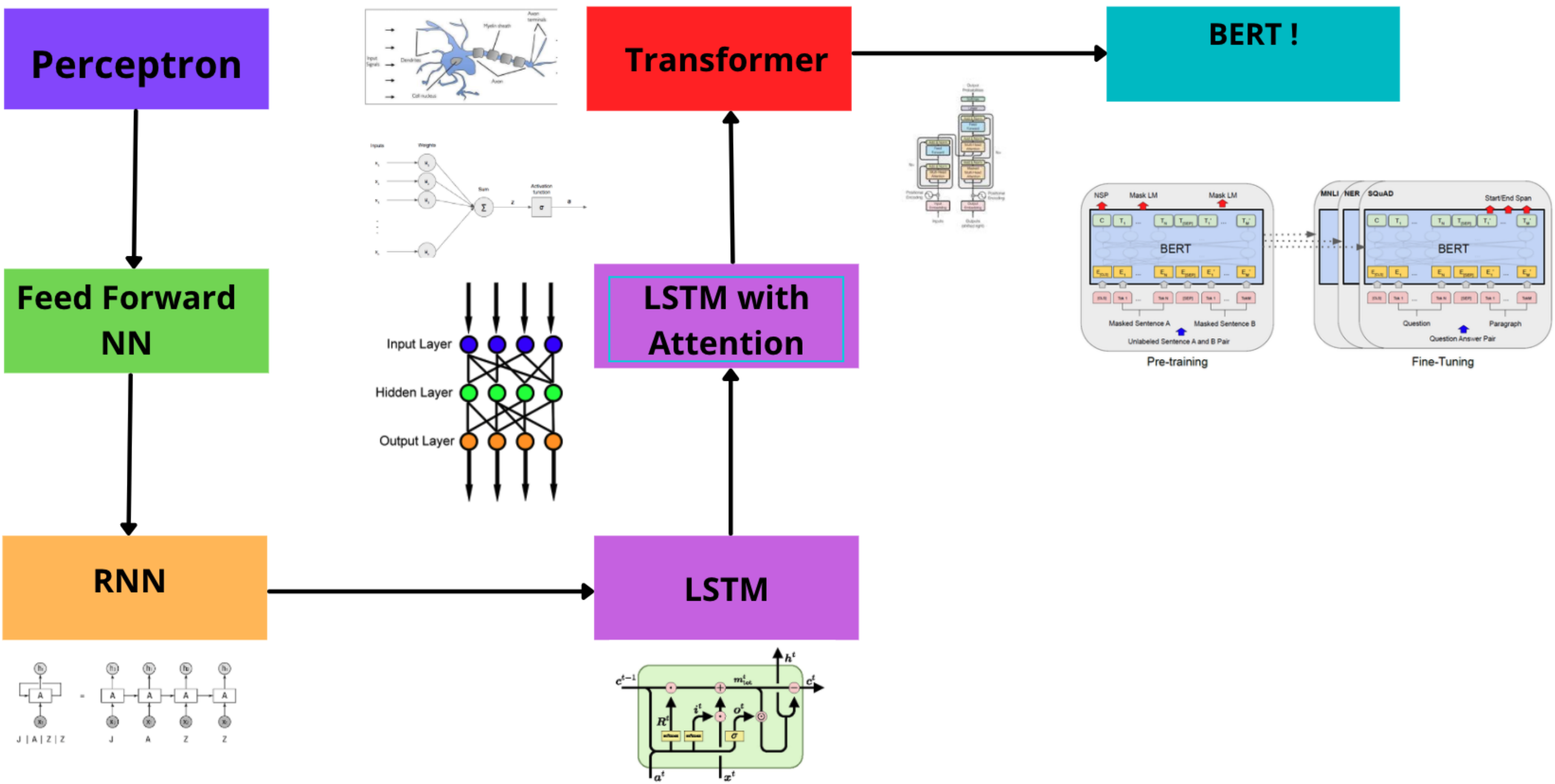
$W^{(?)}$

Sentence Encoding

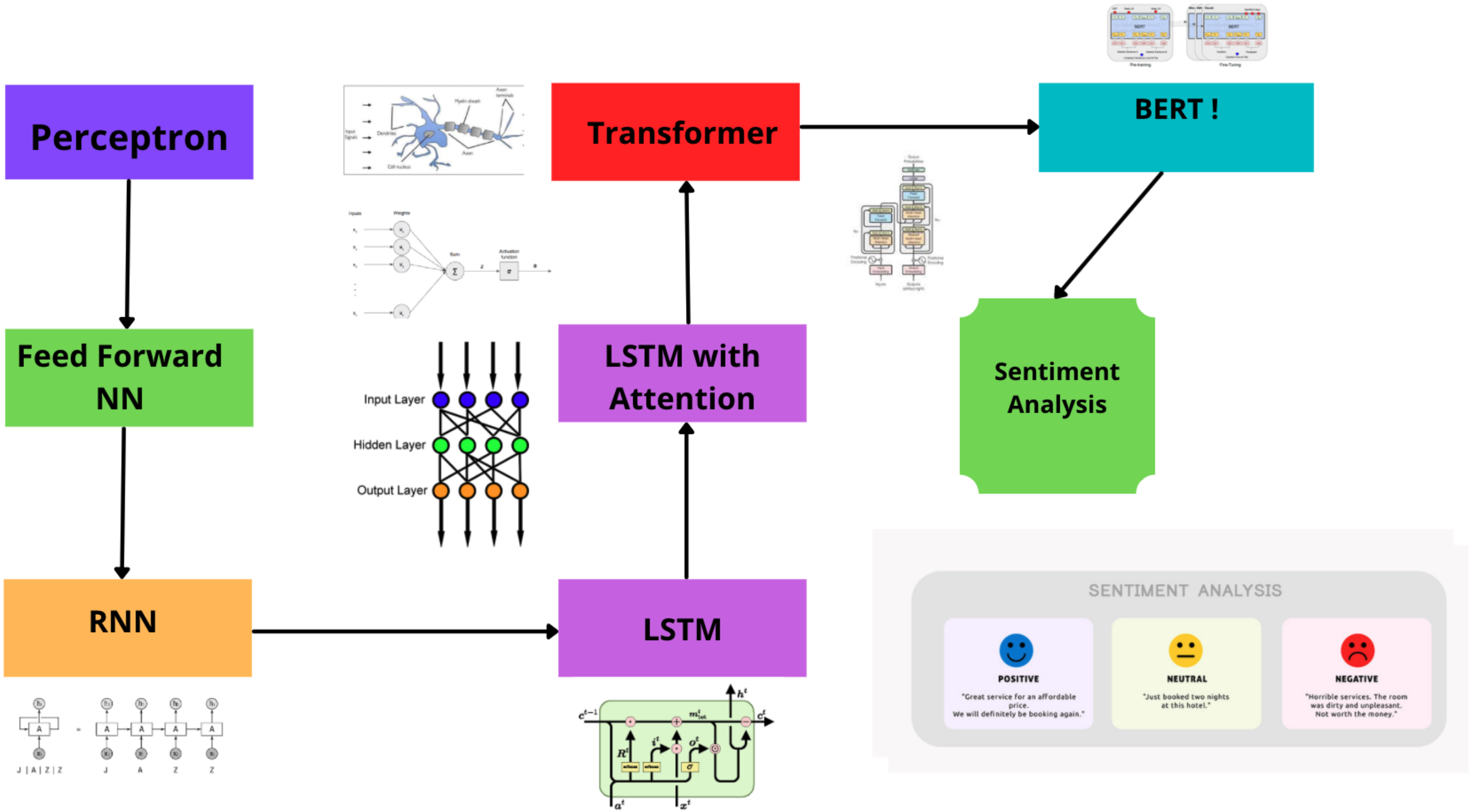


Sentence

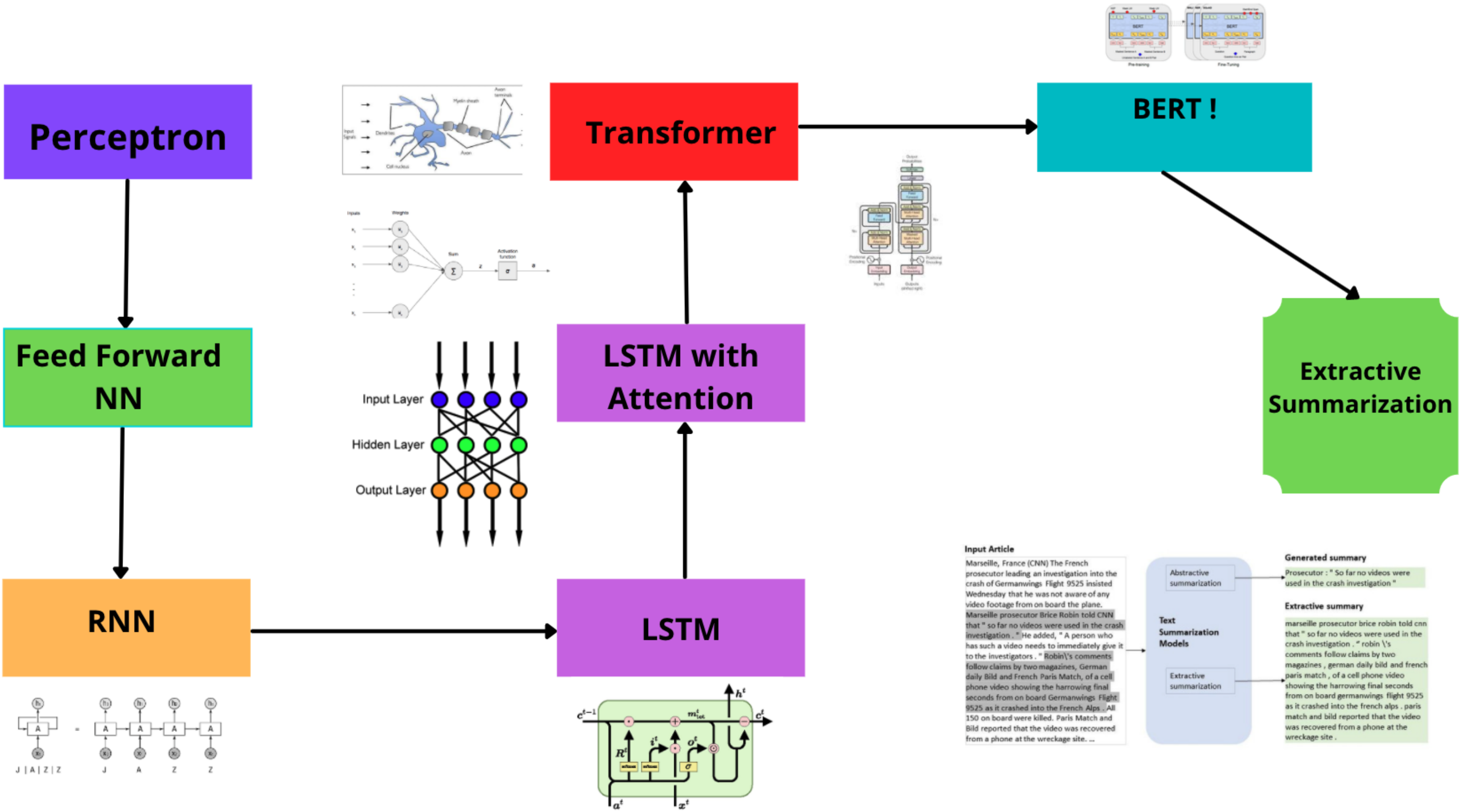
Evolution of DNN architectures for NLP!



Evolution of DNN architectures for NLP!



Evolution of DNN architectures for NLP!



Transformer Archtiecture

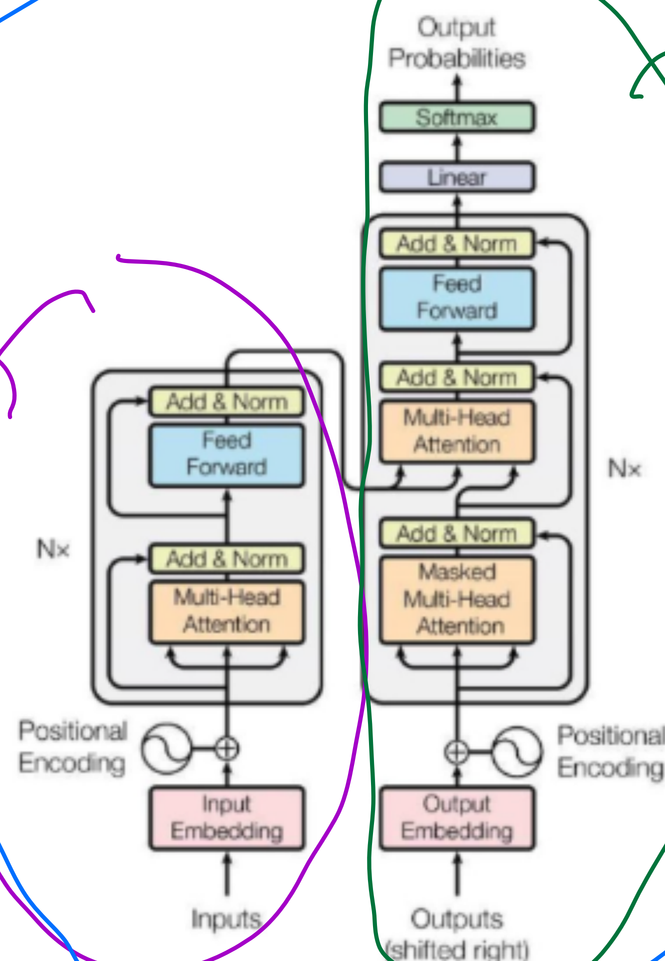
Encoder+Decoder: BART, T5, ..

(only Encoder)

BERT + its variants

Bi-directional Encoding

Good Rep. for Downstream Learning



Encoder

Decoder

only Decoder

GPT & its variants

very Good Language Modeling & Generation

"Auto-Regressive"

Transformer

Reference: Attention is all you need!

2 Attentions
1. Self Attention
2. Cross Attention

Cross Attention → Input-output Attention

Encoding of token "dog" → Rich, Contextual & deep encoding of "dog"

N = 6

FFN + Attention

"dog"

the dog is

Layer = 1 → Attention

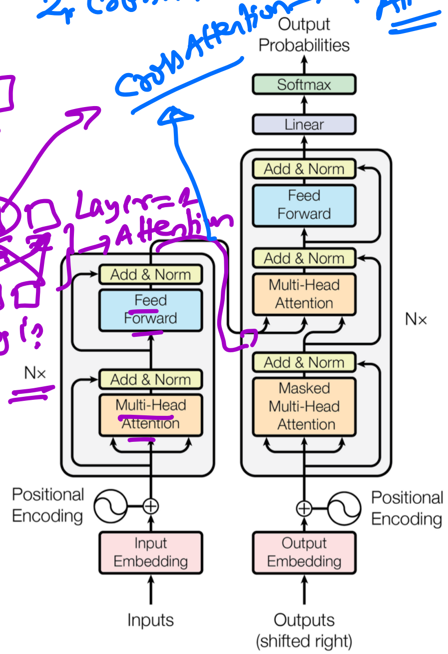
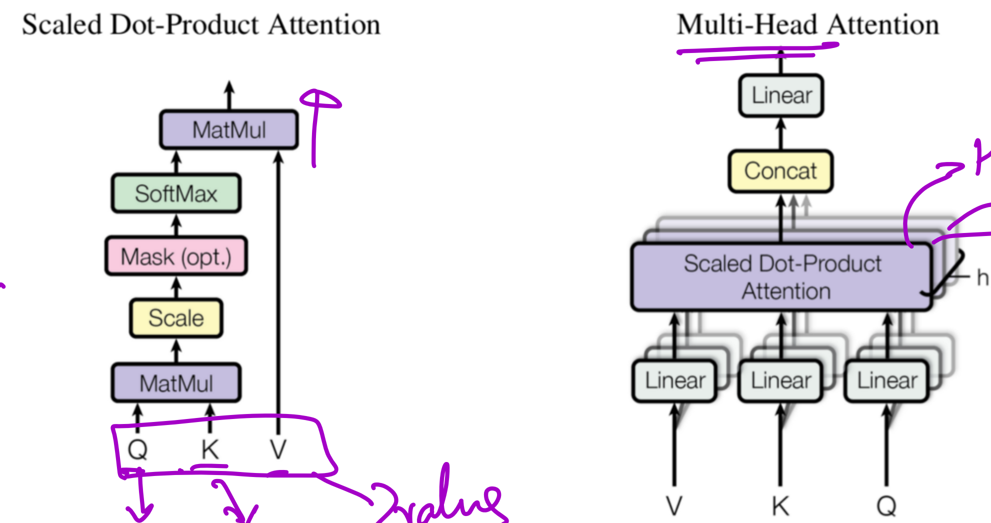


Figure 1: The Transformer - model architecture.



Head 1
Head 3
Head h

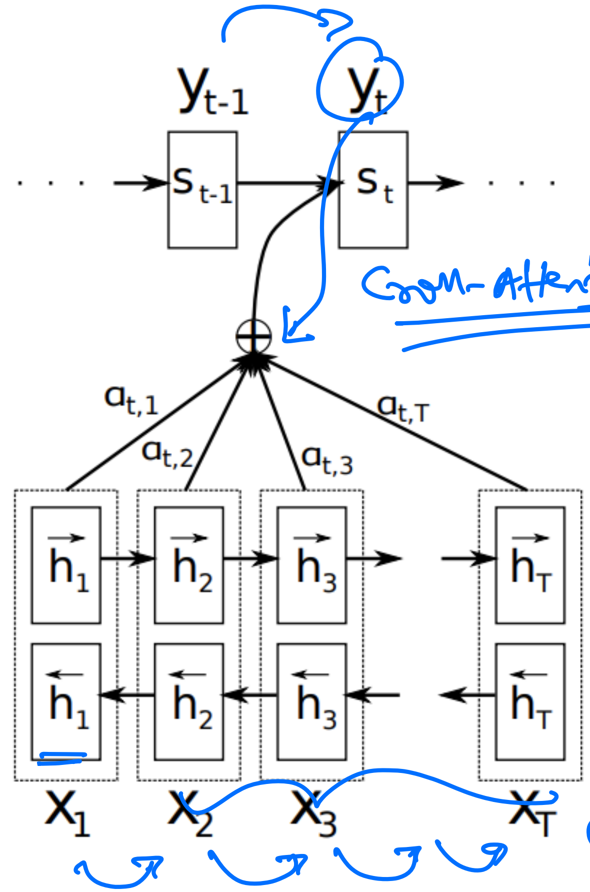
Query Key Value

Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

First Attention Models

Reference paper

LSTM attention

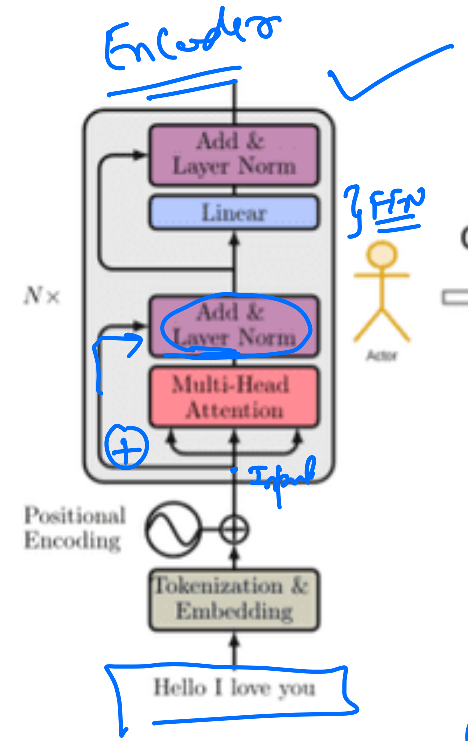


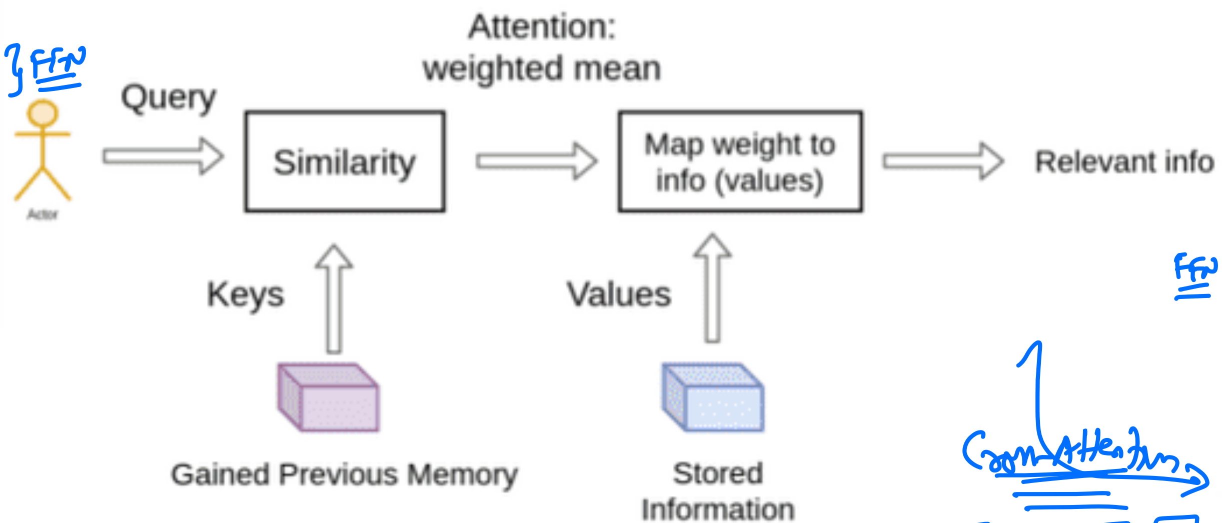
Cross-Attention !!

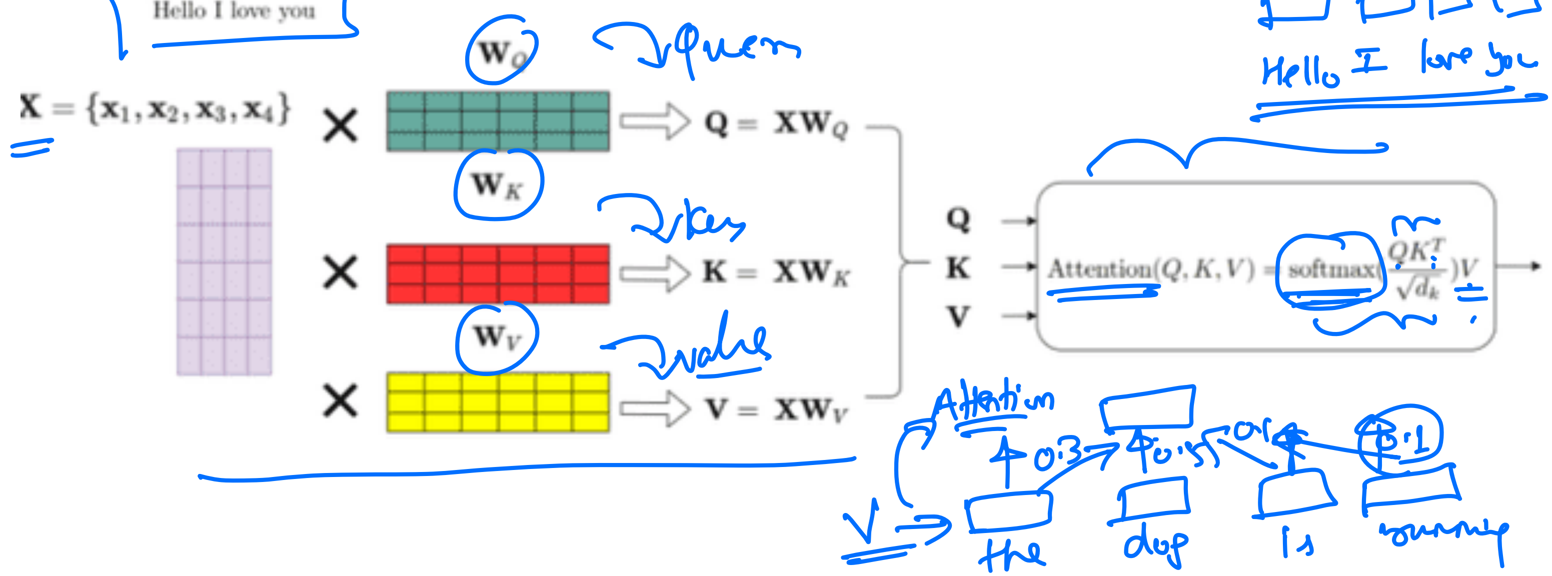
← Inputs

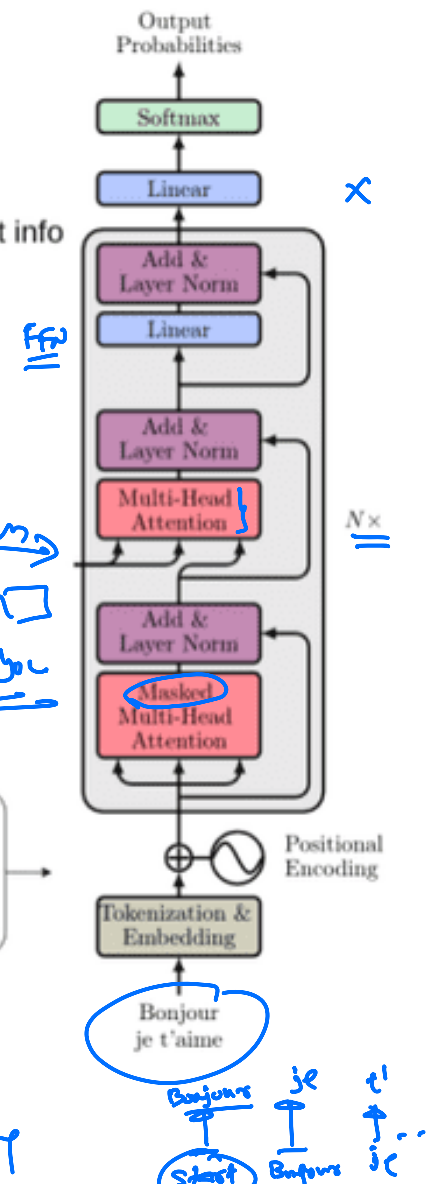
Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Transformers Architecture







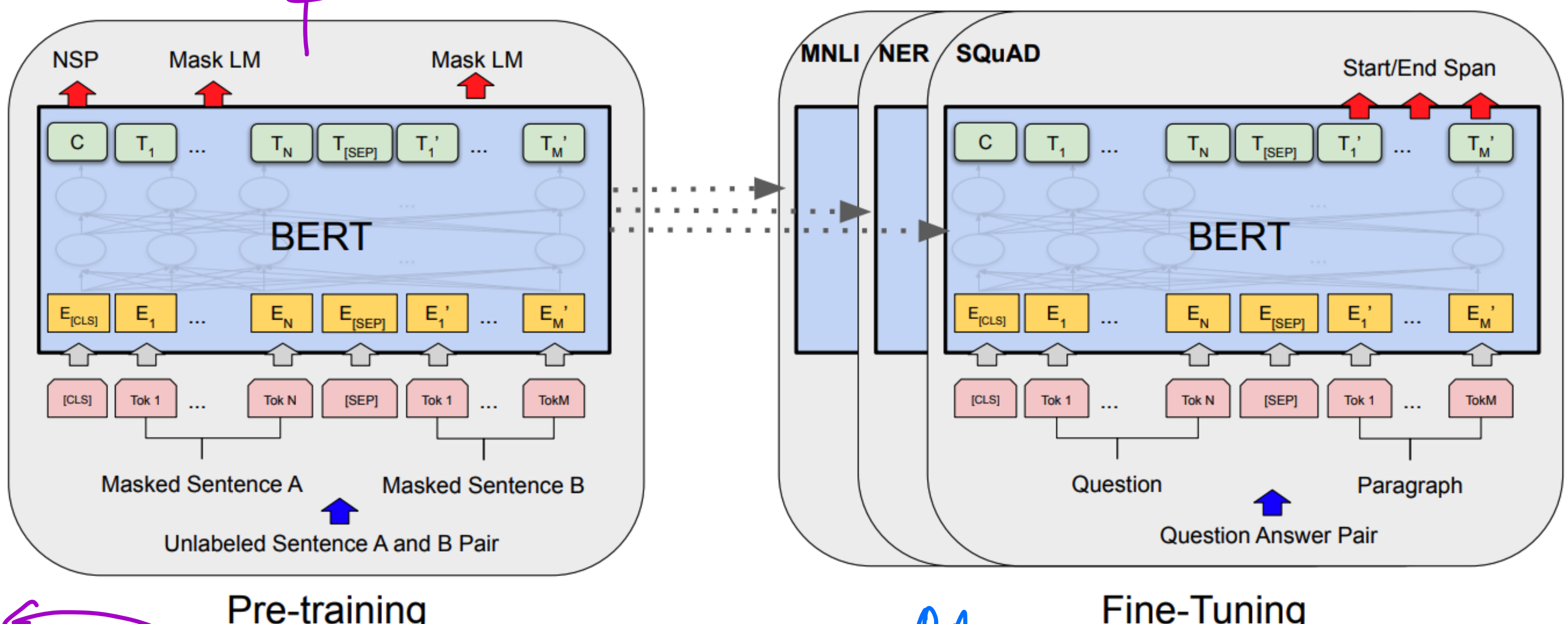


BERT - Bi-directional Encoders from Transformers

verb noun

classifier

Token Rep

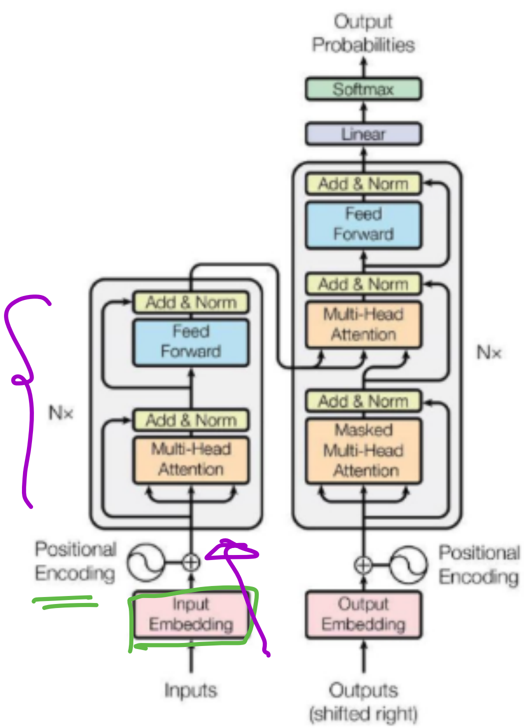


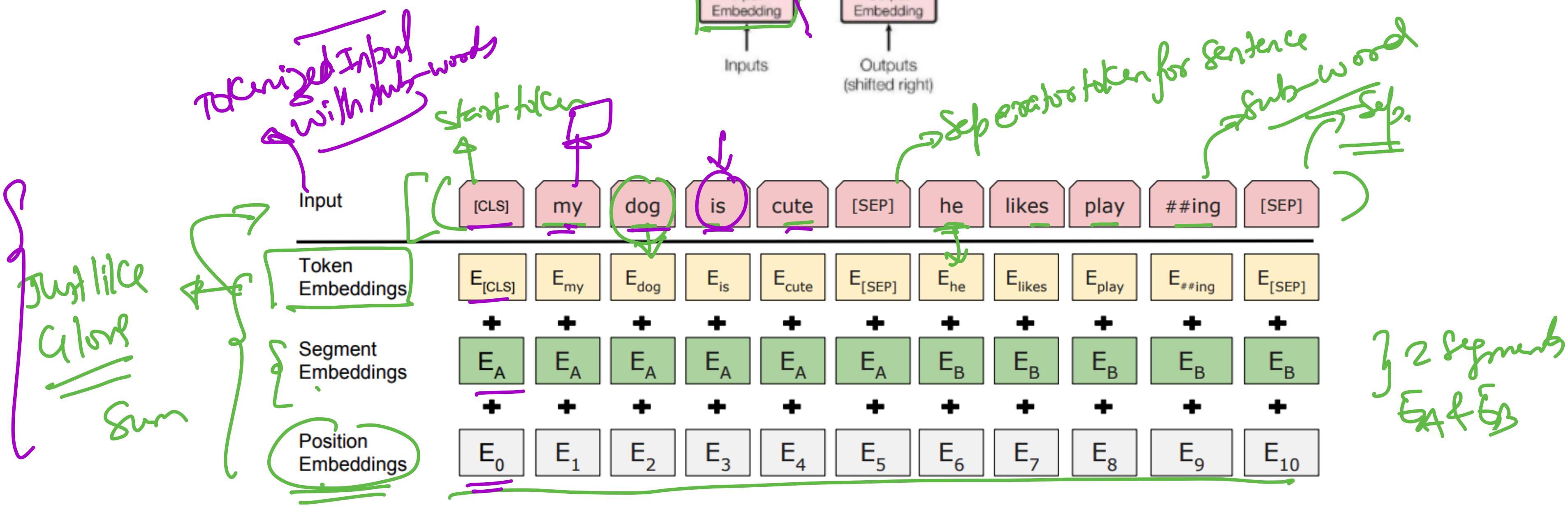
PoS
NER
Emotion Detection

BERT → Extractive → QA
→ Summarization

BART → Abstractive/Generative → QA
→ Summarization

BERT Embeddings





BERT pre-training

Two Tasks

1. **Masked LM Model:** Mask a word in the middle of a sentence and have BERT predict the masked word
2. **Next-sentence prediction:** Predict the next sentence - Use both positive and negative labels. How are these generated?

BERT pre-training

Two Tasks

1. **Masked LM Model:** Mask a word in the middle of a sentence and have BERT predict the masked word
2. **Next-sentence prediction:** Predict the next sentence - Use both positive and negative labels. How are these generated?

ICE: Supervised or Un-supervised?

Self-supervised

1. Are the above two tasks supervised or un-supervised?

BERT pre-training

Two Tasks

1. **Masked LM Model:** Mask a word in the middle of a sentence and have BERT predict the masked word
2. **Next-sentence prediction:** Predict the next sentence - Use both positive and negative labels. How are these generated?

ICE: Supervised or Un-supervised?

1. Are the above two tasks supervised or un-supervised?

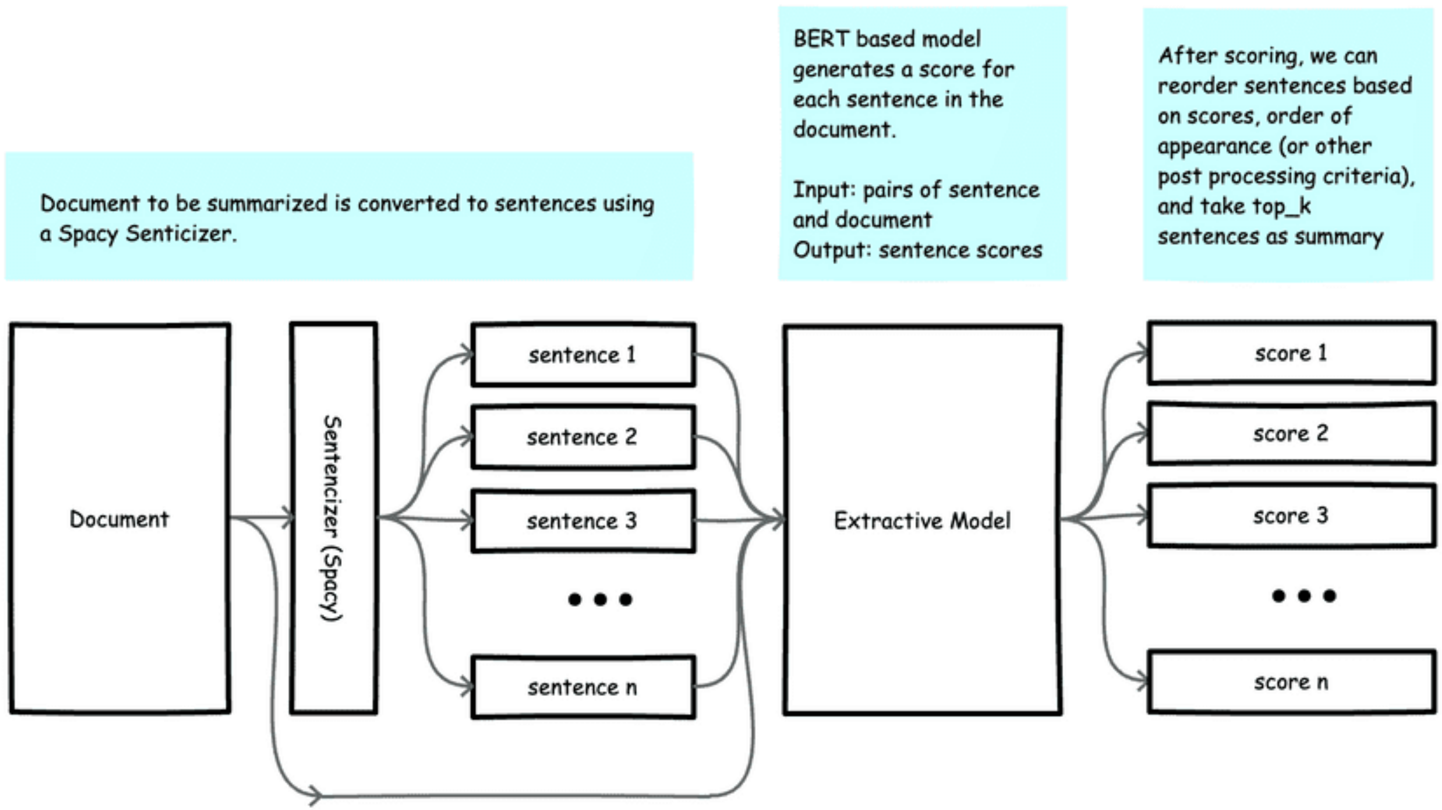
Data set!

English Wikipedia and book corpus documents!

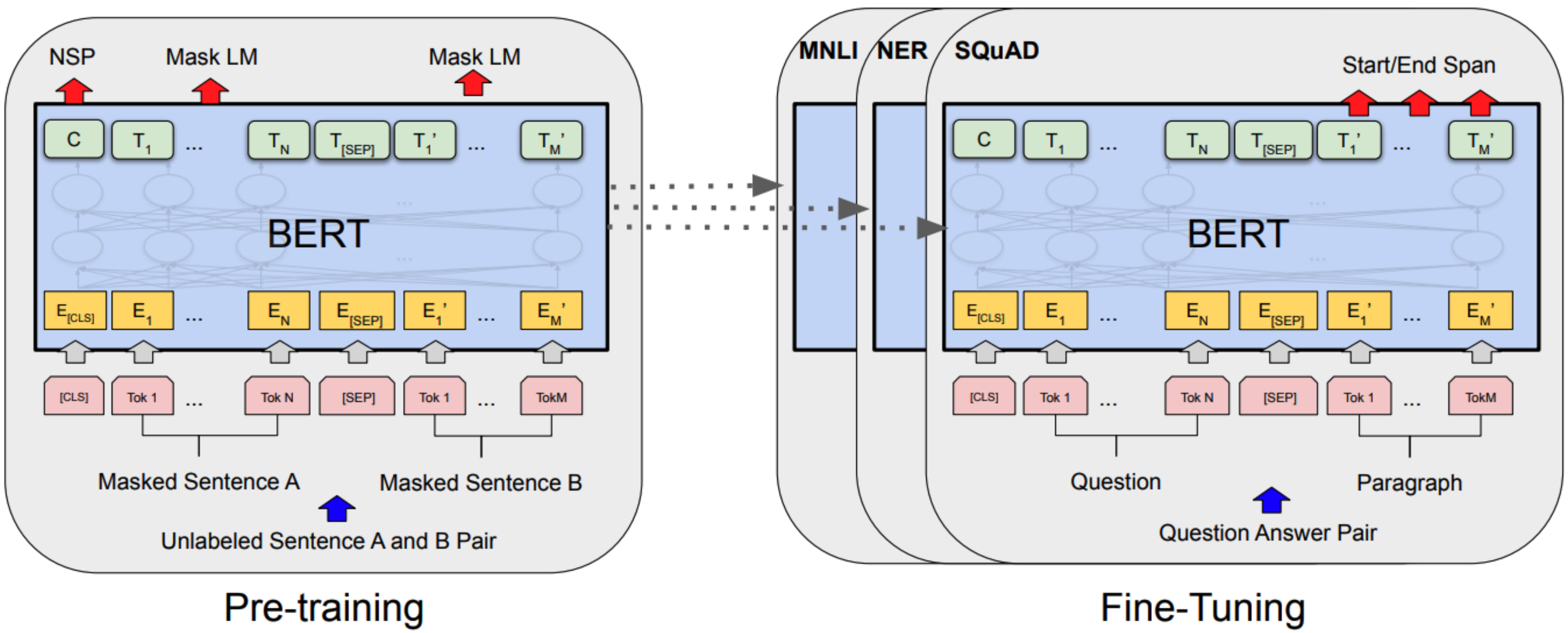
BERT - Bi-directional Encoders from Transformers

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

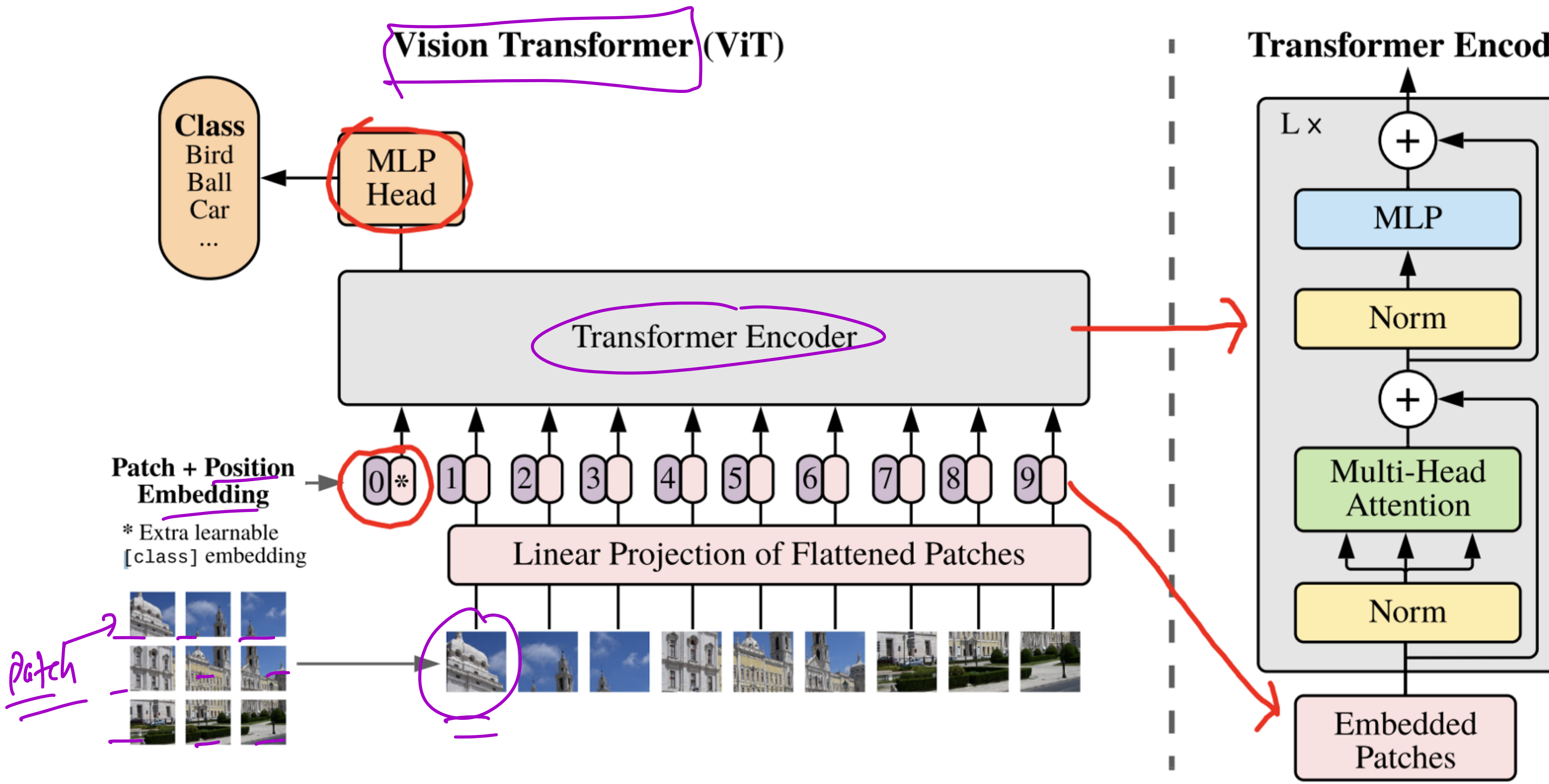
Document Summarization — BERT Based Extractive Model



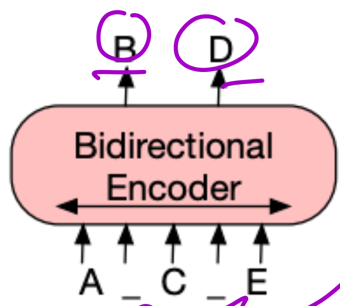
Question Answering — BERT Based Extractive Model



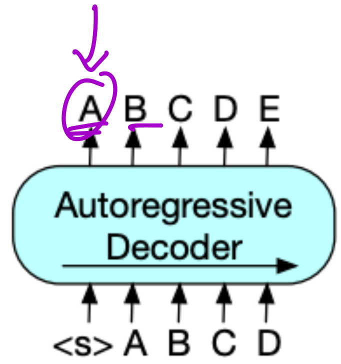
Vision Transformers: Transformers Architecture for Vision



BERT, BART and GPT archs and tasks



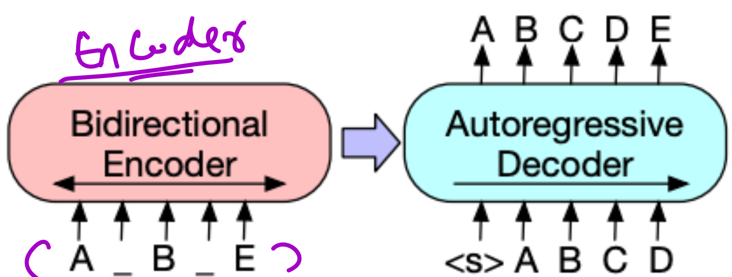
Masked



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.

(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.

Encoder



Decoder

(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Figure 1: A schematic comparison of BART with BERT (Devlin et al., 2019) and GPT (Radford et al., 2018).

BART Paper

BART

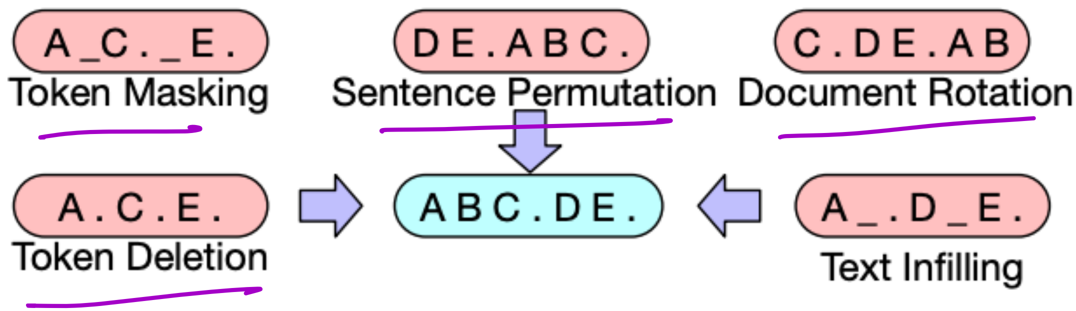
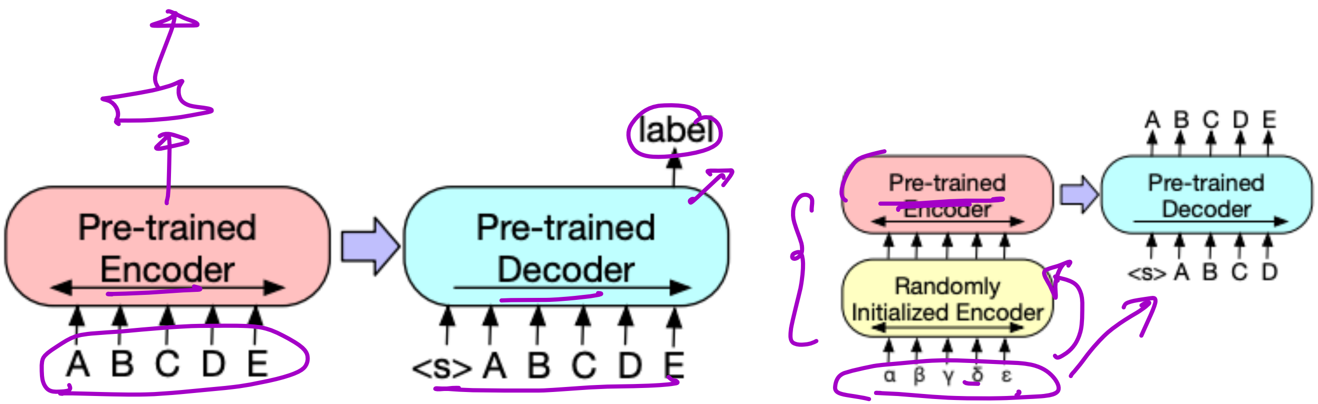


Figure 2: Transformations for noising the input that we experiment with. These transformations can be composed.

BART Paper

BART



(a) To use BART for classification problems, the same input is fed into the encoder and decoder, and the representation from the final output is used.

(b) For machine translation, we learn a small additional encoder that replaces the word embeddings in BART. The new encoder can use a disjoint vocabulary.

Figure 3: Fine tuning BART for classification and translation.

BART Paper

Breakouts Time #1

Queries in search → Head → Simple Retrieval / Simple Model
→ Tail → Generative ML model

Auto-complete — 5 mins

Let's say you are tasked with building an in-email auto-completion application, which can help complete partial sentences into full sentences through suggestions (auto-complete). Auto-complete is also called **type ahead** for query completion in the context of search. What's a traditional non-Machine learning way of doing auto-complete/query completion? How would you use what we have learned so far to use ML to model this? What architecture would you use? What would be your data? And what are some pitfalls or pain-points your model should address?

Can you tell me about ___?

Transformers Demo on Paraphrasing Task

1. **Pre-Training:** We don't do pre-training as that's expensive, requires lots of compute over many days, models have already been optimized and leaves a huge carbon footprint.

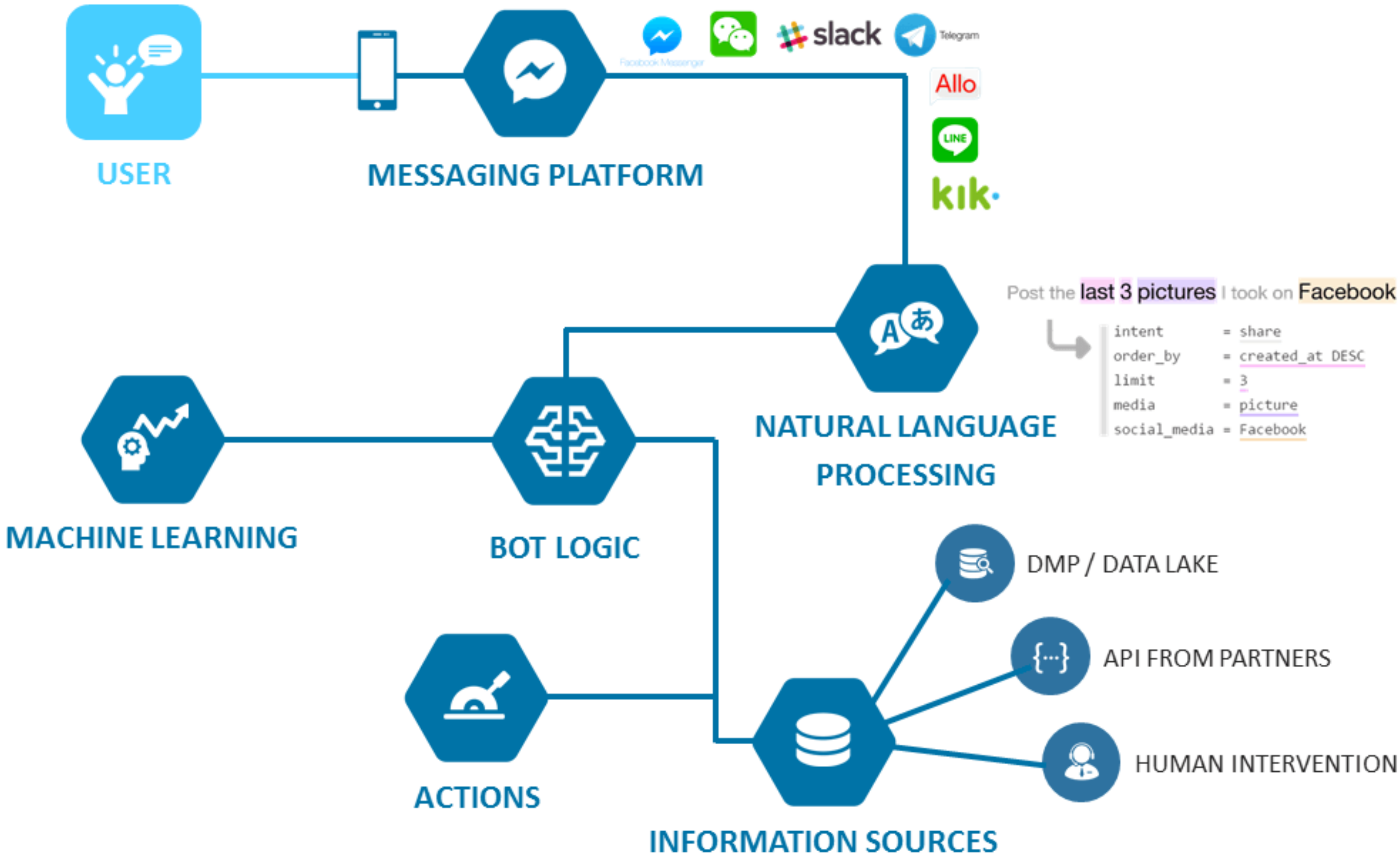
Transformers Demo on Paraphrasing Task

1. **Pre-Training:** We don't do pre-training as that's expensive, requires lots of compute over many days, models have already been optimized and leaves a huge carbon footprint.
2. **Fine-Tuning:** But we can **leverage** pre-training so we don't have to build a model that understands language from sractch. For instance BERT or ALBERT will do it for us. But needs to be fine-tuned to get good performance on our task of interest.

Transformers Demo on Paraphrasing Task

1. **Pre-Training:** We don't do pre-training as that's expensive, requires lots of compute over many days, models have already been optimized and leaves a huge carbon footprint.
2. **Fine-Tuning:** But we can **leverage** pre-training so we don't have to build a model that understands language from sractch. For instance BERT or ALBERT will do it for us. But needs to be fine-tuned to get good performance on our task of interest.
3. **Notebook Demo:** Let's take a look at how fine-tuning can be done using Hugging Face Libraries.

Chat Bots


USER
MESSAGING PLATFORM
slack
Telegram
Allo
LINE
kik·
Post the last 3 pictures I took on Facebook
intent = share
order_by = created_at DESC
limit = 3
media = picture
social_media = Facebook
NATURAL LANGUAGE PROCESSING
MACHINE LEARNING
BOT LOGIC
DMP / DATA LAKE
API FROM PARTNERS
HUMAN INTERVENTION
ACTIONS
INFORMATION SOURCES

Breakouts Time #2

Retrieving Tables with Chat bots — 7 mins

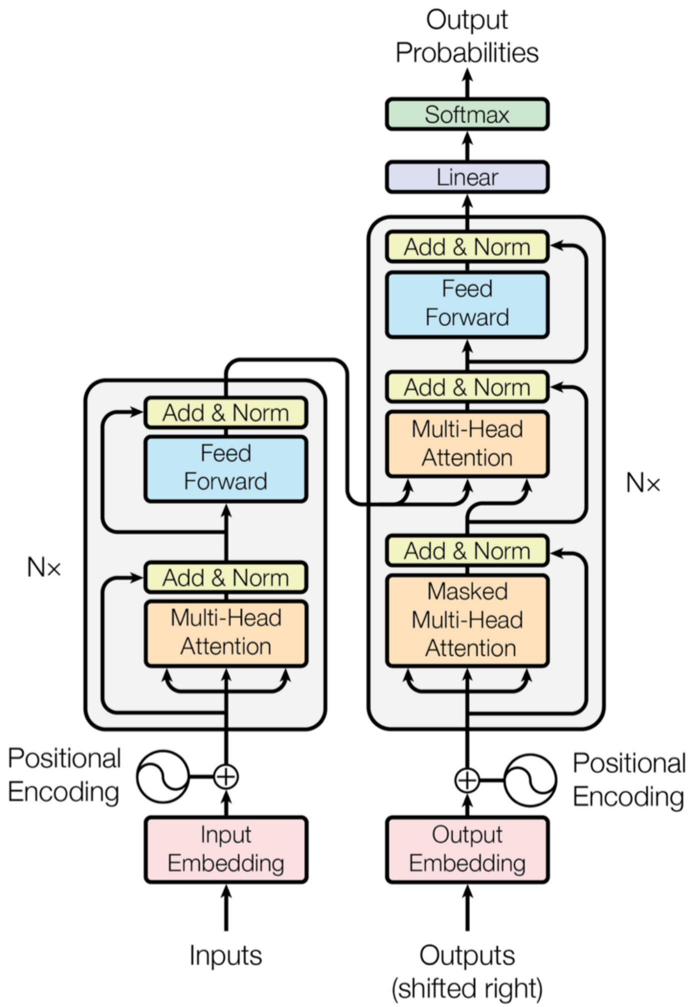
You are building a chat-bot product at your company where queries come in from customers that own data in your company's cloud service. Your chat-bot responds retrieves the right table or combination of tables (through merge/filter operations) that contains this information or returns back with follow up questions to get more precise information or get back with a "Sorry, I don't have that information" response. How would you go about building a chat-bot like this? What data would you use? What ML models would you use, would it be supervised or un-supervised learning? What would be your evaluation metric? How would you test if your chat bot is accurate in its responses?

Additional Slides

Attention Motivation

Transformer

Reference: Attention is all you need!



Transformers Architecture

Transformer

Reference: Attention is all you need!

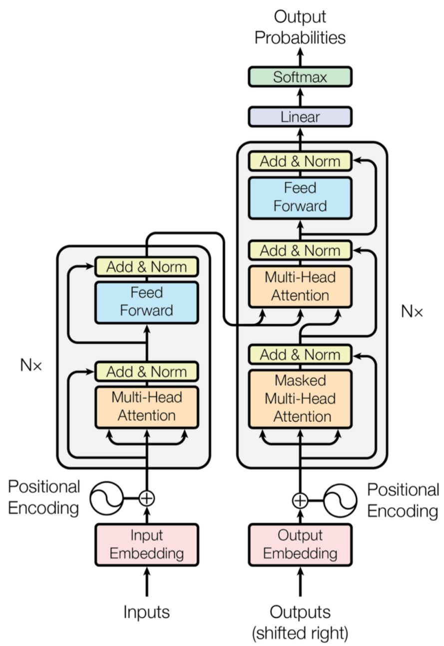


Figure 1: The Transformer - model architecture.

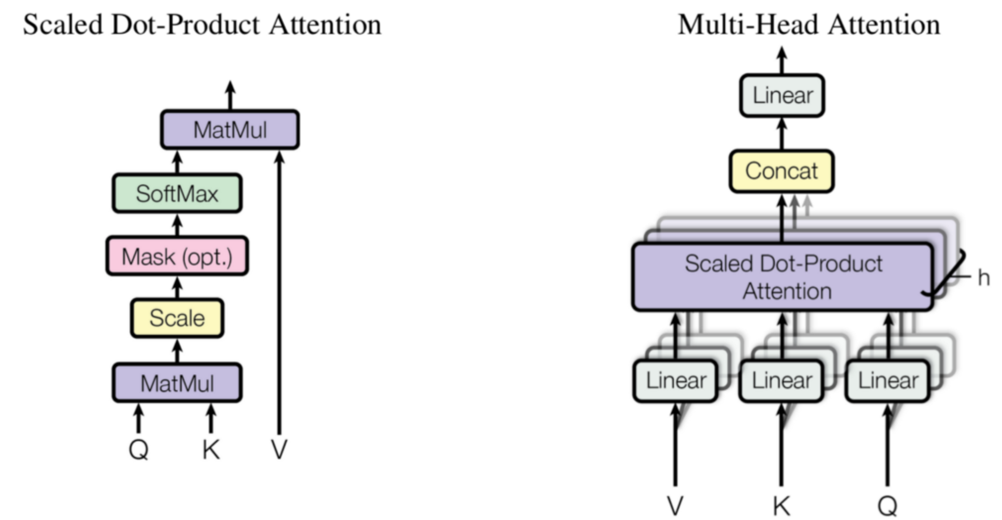


Figure 2: (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel.

Retrieving Tables from queries

Context

Many a times, we have a Natural Language Query - E.g. "Which quarter in the past 5 years had the most amount of sales for fashion products". From this natural language query, we want to retrieve a data table that is perhaps the most similar to the query and helps answer the query.

Retrieving Tables from queries

Context

Many a times, we have a Natural Language Query - E.g. "Which quarter in the past 5 years had the most amount of sales for fashion products". From this natural language query, we want to retrieve a data table that is perhaps the most similar to the query and helps answer the query.

SQL queries vs Natural Language queries

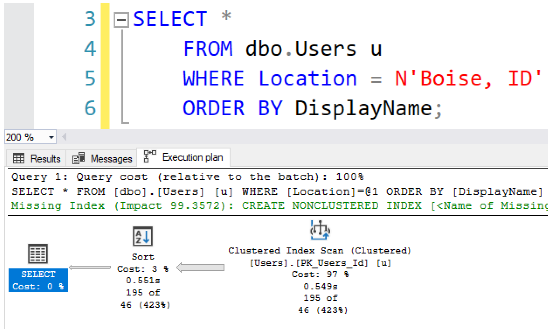


Table2Vec

Region	Release Date	Label	Release Format
United Kingdom	22 September 2008	Super Records	DVD
Ireland		cords	DVD
Japan		rax	DVD
Argentina	18 May 2009	EMI Music	Digital Download
Singapore	12 June 2009	Warner Music	DVD
Spain	1 December 2009	EMI Music Spain	Digital Download

pgTitle: Radio:Active
secondTitle: Release history
caption: Release history

Table2Vec

Embedding a Table?

1. Identify key entities in a table - E.g. headers and key words
2. Approach 1: Take a weighted average of these entity embeddings and call it the Table embedding
3. Approach 2: Pass the key entities in the table through a sequence model and generate a Table embedding.
4. Other approaches?

Query to a Table

Given a Natural Language query, how could you fuzzy match tables to a query?

1. Get a query embedding

Query to a Table

Given a Natural Language query, how could you fuzzy match tables to a query?

1. Get a query embedding
2. Get a table embedding

Query to a Table

Given a Natural Language query, how could you fuzzy match tables to a query?

1. Get a query embedding
2. Get a table embedding
3. Use an appropriate metric to do the matching!

ICE #5

What similarity metric would be appropriate to match a query with a table, given embeddings for both that are constructed out of word/entity embeddings?

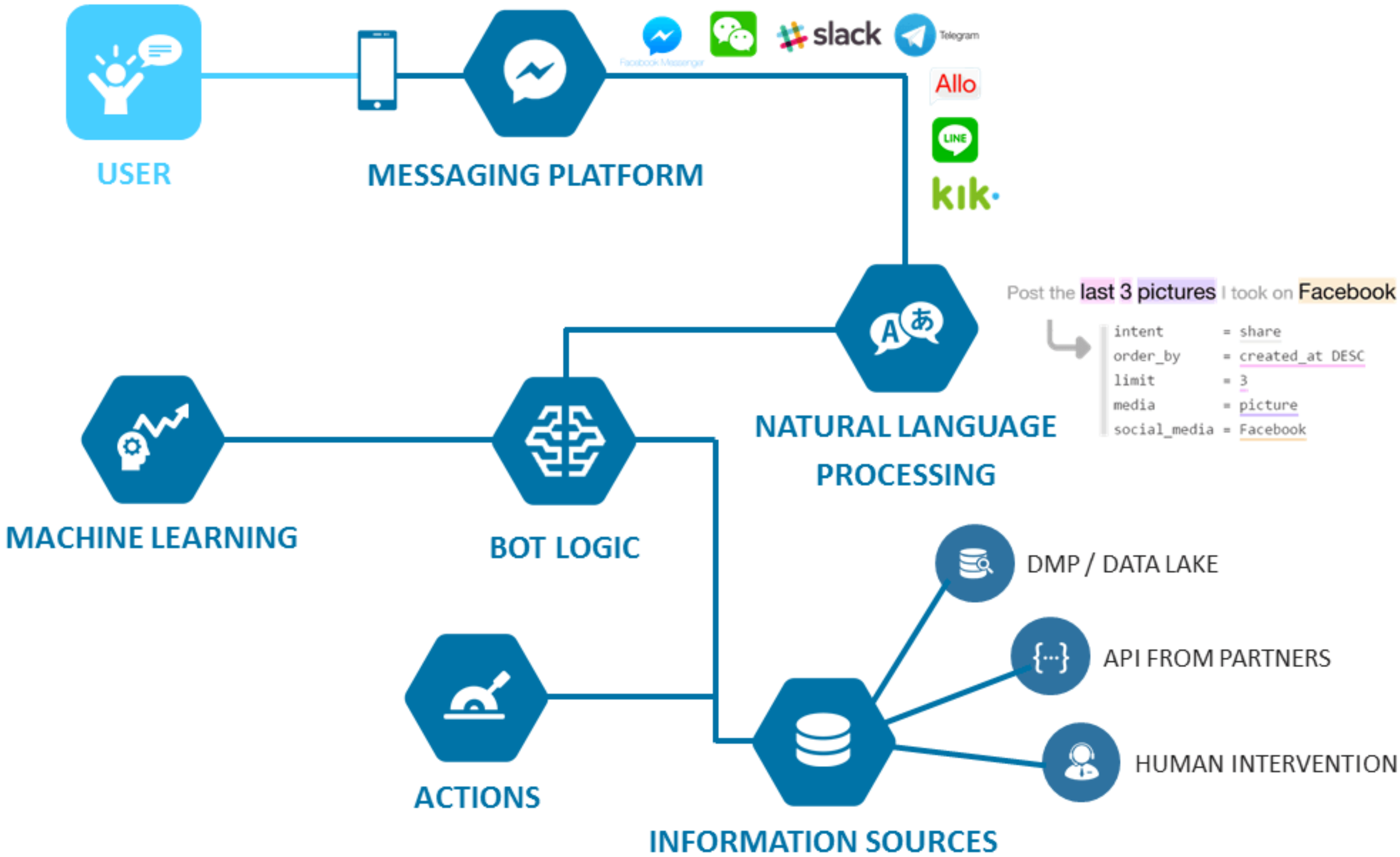
1. Jaccard Similarity
2. Ranking Similarity
3. Cosine Similarity
4. Sentence Similarity

ICE #6

Let's say we want to automatically convert a **Natural Language Query** to a **SQL** query. E.g. "Which quarter in the past 5 years had the most amount of sales for fashion products" to "SELECT ... FROM ... WHERE ..." What kind of deep learning architecture would support this problem?

1. Siamese Network
2. LSTM to LSTM sequence model
3. BERT model
4. Feed Forward Neural Network

Chat Bots



USER
MESSAGING PLATFORM
slack
Telegram
Allo
kik
Post the last 3 pictures I took on Facebook
intent = share
order_by = created_at DESC
limit = 3
media = picture
social_media = Facebook
NATURAL LANGUAGE PROCESSING
MACHINE LEARNING
BOT LOGIC
DMP / DATA LAKE
API FROM PARTNERS
HUMAN INTERVENTION
ACTIONS
INFORMATION SOURCES

Identifying bad actors from social media messages

Context

When messages on social media can spew hate or be inappropriate - Can a model be learned to classify them as inappropriate? E.g.

1. "You are f**** annoying me right now."

Identifying bad actors from social media messages

Context

When messages on social media can spew hate or be inappropriate - Can a model be learned to classify them as inappropriate? E.g.

1. “You are f**** annoying me right now.”
2. “If you don’t follow up on what we discussed, then things may not look so good for you.”

Breakouts Time #3

Identifying inappropriate speech (7 mins)

Think of a simple baseline model that can help you identify a message/sentence on social media as inappropriate. When would this baseline model work? When would it fail? What deep learning architecture can help you fix the baseline model? What data would you use for your model? How would you gather the data for training? What do the inputs and labels look like? What are some evaluation metrics that can be used to measure the success of your models?

Extra Slides

Breakouts Time 1

5 mins

Discuss in your groups what are some real-world applications of any or many of the Auto Encoder Architectures we discussed so far you can think of in your area of work or in a standard context e.g. images.