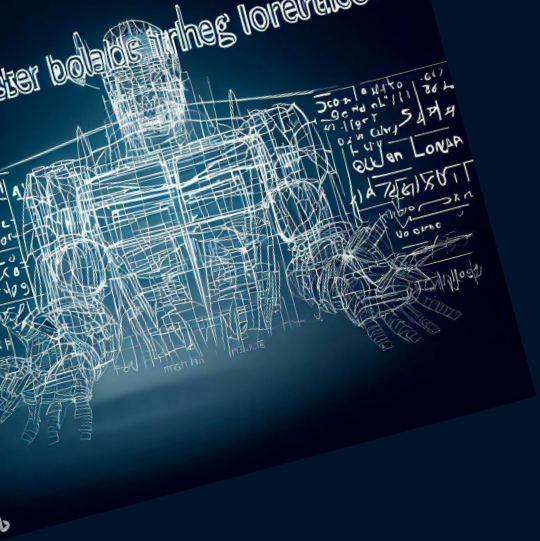


LARGE LANGUAGE MODELS

INTRODUCTION TO LANGUAGE MODELS

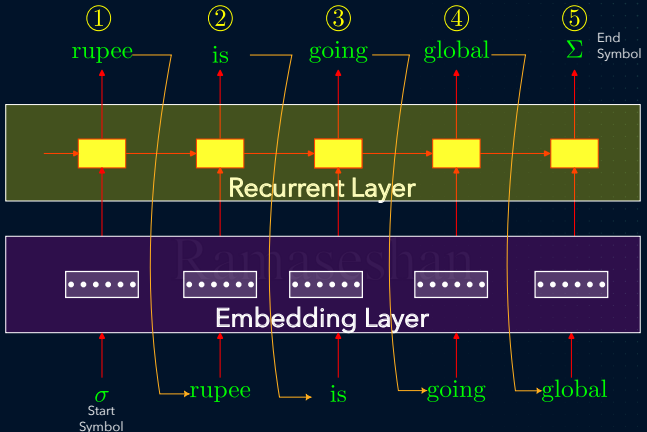
July 17-21, 2023

Ramaseshan Ramachandran

Machine Translation
Embeddings from Language Models
Static Representation of word vectors
Contextual Word Representation
ELMO

Large Language Models
Self-Attention
Multi-head Attention
LLaMA
Bias and Challenges
Responsible AI

RECURRENT TEXT GENERATOR/TRANSDUCER MODEL

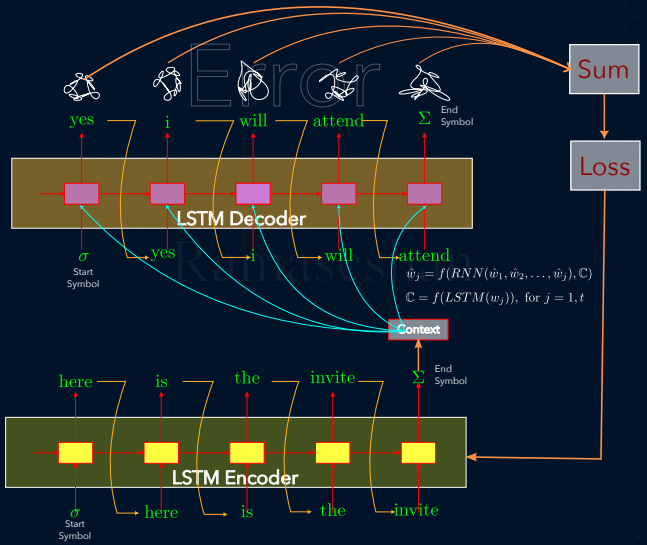


① $p(rupee|\sigma)$ ② $p(is|\sigma, rupee)$

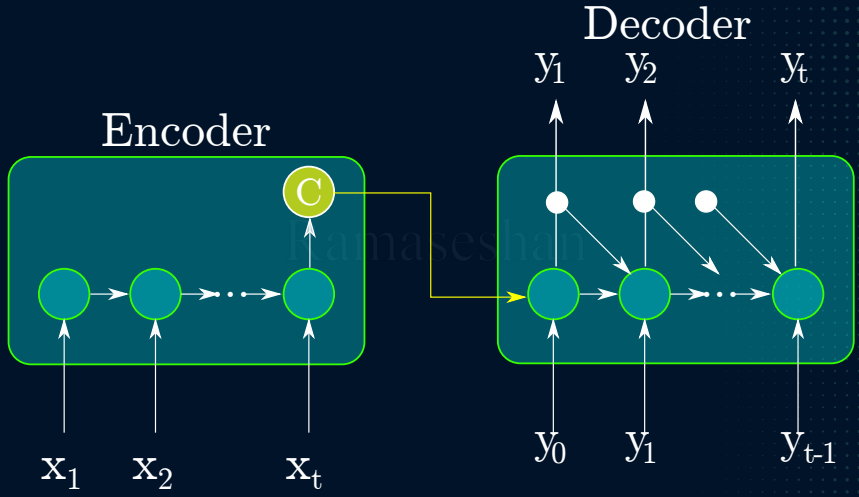
③ $p(going|\sigma, rupee, is)$ ④ $p(global|\sigma, rupee, is, going)$

⑤ $p(\Sigma|\sigma, rupee, is, going, global)$

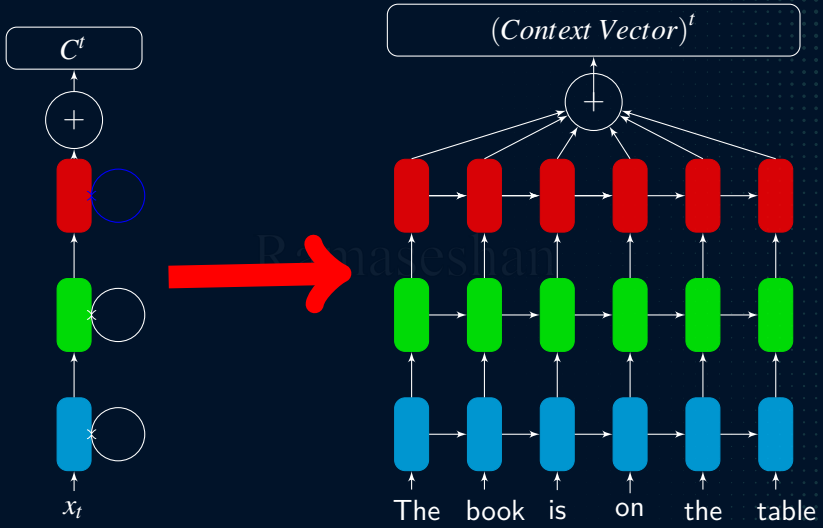
SEQUENCE TO SEQUENCE GENERATOR



RNN-BASED ENCODER-DECODER - SEQ2SEQ TRANSLATION



SEQ2SEQ ENCODER



BAHDANAU ET AL. NMT MODEL

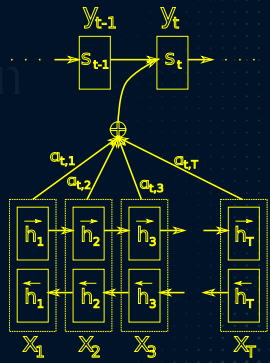
Input is a sequence of words $x_1, ..., x_n$
Target sentence is again a sequence of words $y_1, ..., y_m$

Encoder

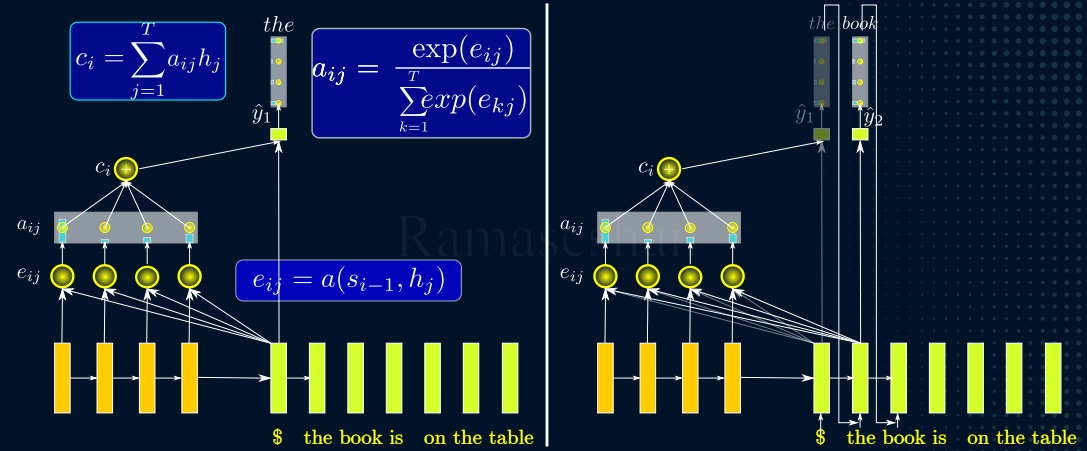
Let $(h_1, ..., h_n)$ be the hidden vectors representing the input sentence. These vectors are the output of a bi-LSTM/bi-GRU for instance, and capture contextual representation of each word in the sentence

Decoder

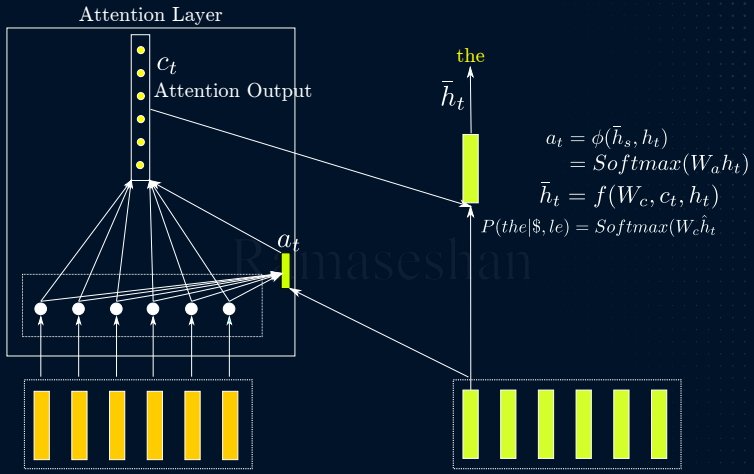
The hidden states s_i of the decoder are computed using a recursive formula of the form $s_i = f(s_{i-1}, y_{i-1}, c_i)$, where s_{i-1} is the previous hidden vector, y_{i-1} is the generated word at the previous step, and c_i is a context vector that capture the context from the original sentence that is relevant to the time step i of the decoder.



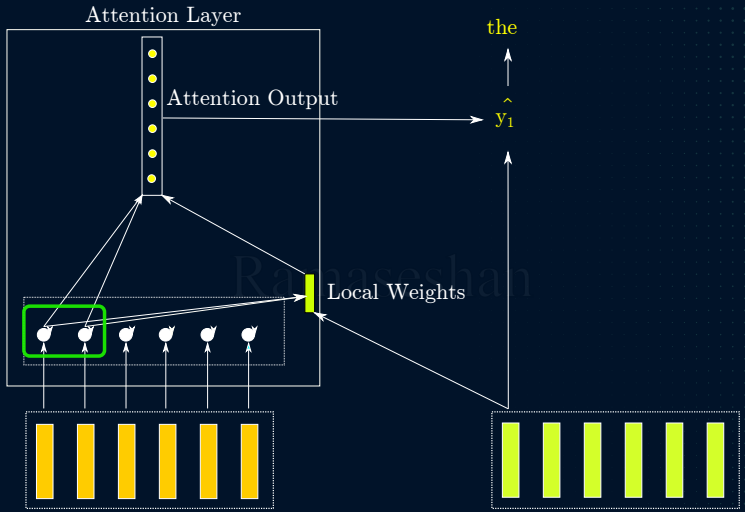
ATTENTION

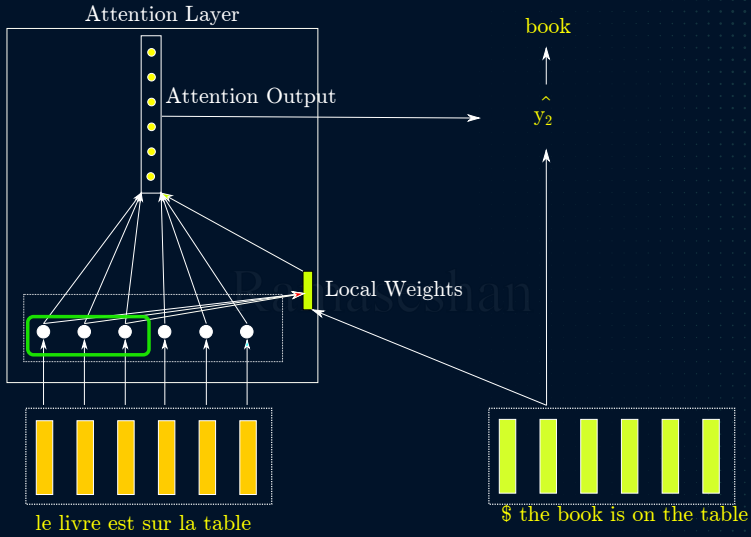

$c_i = \sum_{j=1}^{T} a_{ij} h_j$
$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} exp(e_{kj})}$
$e_{ij} = a(s_{i-1}, h_j)$
the
book
$\hat{y}_1$
$\hat{y}_2$
c_i
a_{ij}
e_{ij}
$ the book is on the table
$ the book is on the table

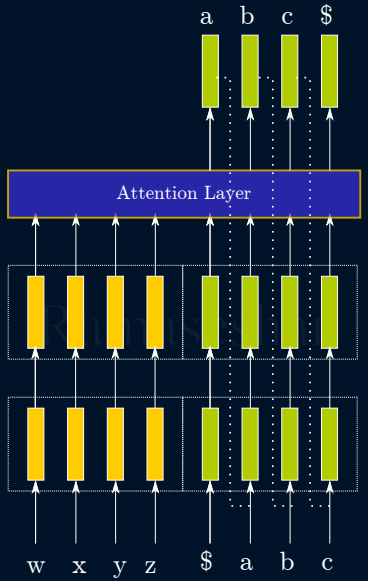
TRANSLATION WITH GLOBAL ATTENTION



TRANSLATION WITH LOCAL ATTENTION




Attention Layer
Attention Output
book
$\hat{y}_2$
Local Weights
le livre est sur la table
$ the book is on the table



Source: Minh-Thang Luong et al, Effective Approaches to AIttention-based Neural Machine Translation

TRAINING WITH CONTEXT

- The next token in the sequence gets the largest probability mass
- If there are t words in the sequence $(w_1, w_2, w_3, \ldots, w_t)$, the transducer will have $t+1$ input and output neurons

$$p(w_{j+1}|w_1, w_2 \ldots, w_j) = f(RNN(\theta, w_1, w_2, \ldots, w_j)), \text{ where } j \in [1,t]$$
$$= f(RNN(\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_j))$$

and $\hat{w}_j = p(w_j|\hat{w}_1, \hat{w}_2, \ldots, \hat{w}_{j-1})$

- It is possible to create the next token using a conditional context, such as a topic or class of the current content, active or passive voice, etc.
- Example sentence generated without a context
  - *determine the time it takes a piece of glass to hit the ground?* ***A car drives straight off the edge of a cliff***
- Example sentence with context
  - *determine the time it takes a piece of glass to hit the ground,* ***if a ball is dropped from a height of 45m****.*

A TYPICAL SETUP

Sentence pairs	3-5M
English words	110M
French words	116M
Vocabulary	≈50K (Source and Target)
Word Embedding size	1000
Hidden layer	1000 LSTM cells
Stacked Hidden Layer	4-8
Learning Rate	Initially as high as 1 and exponential reduction
Training	
Mini batch Gradient Descend size	128
Training Time	1 GPU - about 7-10 days
Evaluation	Bleu - scores ranging from 27-32

SEQ2SEQ-BASED APPLICATION

- **Next word/sentence prediction**
  Train pairs of consecutive sentences to generate a contenxt sensitive sentences in a sequence
- **Question-Anwering**
  Given a question (including word problems), find context sensitive answers
- **Auto-encoding**
  Building sentence vectors for identifying similar sentences - extending the distributional hypothsis of words into sentences
- **Machine Translation**
  Given pairs of language texts for training, trsnslate an unknown sentence usig the trained model
- **Precise writing**
  Given a long paragraph, compress into one two sentences
- **Title generation**
  Given abstract and title pairs, train and generate title for any new abstract
- **Chatbots**
- **Discourse Analysis**
  Analysis of written, spoken, sign language

ADVANTAGES OF ATTENTION

- Ability to focus on significant part of the sentence
- Ability to peek into source sentence
- Reduces the problem of vanishing gradient
- Alignments are found automatically - no need to train
- Improves NMT performance for alignment

SENSE DISAMBIGUATION OF THE WORD BANK

Synset('bank.n.01')	sloping land (especially the slope beside a body of water)
Synset('depository-financial-institution.n.01')	a financial institution that accepts deposits and channels the money into lending activities
Synset('bank.n.03')	a long ridge or pile
Synset('bank.n.10')	a flight maneuver; aircraft tips laterally about its longitudinal axis (especially in turning)
Synset('bank.v.02')	enclose with a bank
Synset('bank.v.03')	do business with a bank or keep an account at a bank
Synset('bank.v.04')	act as the banker in a game or in gambling
Synset('bank.v.05')	be in the banking business
Synset('deposit.v.02')	put into a bank account
Synset('trust.v.01')	have confidence or faith in

SENSE DISAMBIGUATION - THE WORD PROGRAM

Synset('plan.n.01')	a series of steps to be carried out or goals to be accomplished
Synset('program.n.02')	a system of projects or services intended to meet a public need
Synset('broadcast.n.02')	a radio or television show
Synset('platform.n.02')	a document stating the aims and principles of a political party
Synset('program.n.05')	an announcement of the events that will occur as part of a theatrical or sporting event
Synset('course_of_study.n.01')	an integrated course of academic studies
Synset('program.n.07')	(computer science) a sequence of instructions that a computer can interpret and execute
Synset('program.n.08')	a performance (or series of performances) at a public presentation
Synset('program.v.01')	arrange a program of or for
Synset('program.v.02')	write a computer program

STATIC WORD REPRESENTATION

- Static representation of words
- Irrespective of the context, a polysemous word has the same vector representation
- Insensitive to the context in which they appear
- The word representation of a polysemous word is a biased representation due to certain context/pattern that appear more than any other context in the given corpus
- **It represents syntactic and semantic representations**
- **It does not represent the same word appearing in different linguistic contexts**

LATENT RELATIONSHIP

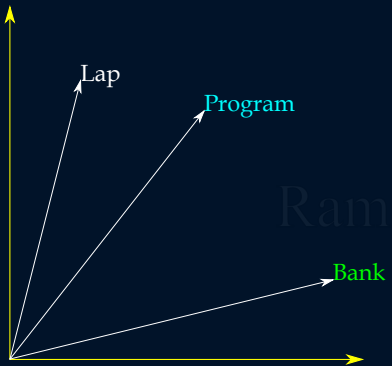
The pairs of words that co-occur in similar patterns tend to have semantic relations. Using this property every word in the vocabulary is encoded as dense vectors using vector space semantic models.

The individual dimension in the dense vectors is no longer intetpretable, except that its value may be proportional to the strength of the relationship it has with the rest of the words in the vocabulary

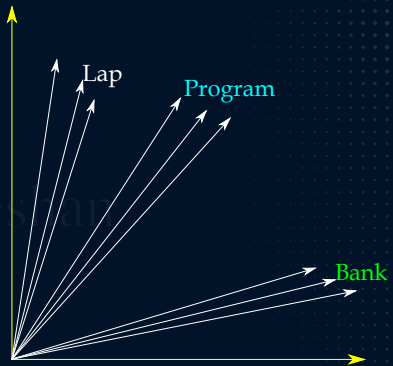
The probabilistic language models use near-by contexts and do not represent any contextual representation of nearby sentences present in a corpus

Is it possible to compute the contextual relationship amoung words, sentences and paragraphs?

CONTEXTUAL WORD REPRESENTATION



Static Word Representation



Contextual Word Representation

CONTEXTUAL WORD REPRESENTATION

Boys are playing cricket near the Indian **bank**

A fundamental aircraft motion is a **banking** turn

Boys are playing cricket on Cauvery river **bank**

What is your **program** today?

Have you completed the **programming** assignment today?

A cultural **program** was telecast on TV

- The RNN layer holds the context specific representation of the words
- Why not use them for contextual word vector generation
- Can we use it for NLP tasks?
- Each token is assigned a representation $w_i = f(w_1, w_2, w_3, ...w_n)$ where n is the number of tokens/words in a sentence

ELMO

1. A word vector is a function of other word vectors in an input sentence **[DBLP:journals/corr/abs-1802-05365]** - $w_i = f(w_1, w_2, w_3, \ldots, w_{t-1})$

   $FORWARD:\quad p(w_1, w_2, \ldots, w_T) = \prod_k^T p(w_k | w_1, w_2, \ldots\ldots, w_{t-1})$

   $BACKWARD:\quad p(w_1, w_2, \ldots, w_T) = \prod_k^T p(w_k | w_{k+1}, w_{k+2}, \ldots\ldots, w_T)$
2. If there are L layers in both forward and backward LSTMs, then the after L layers, $\overrightarrow{h}_{k,L}$ and $\overleftarrow{h}_{k,L}$ are computed
3. $\overrightarrow{h}_{k,L}$ and $\overleftarrow{h}_{k,L}$ are scaled (may use a layer norm)
4. $\overrightarrow{h}_{k,L}$ and $\overleftarrow{h}_{k,L}$ are concatenated for downstream applications
5. For every word in the vocabulary, a linear combination of the its representation across the L layers are obtained
6. There is no need for **unknown token** representation

LARGE LANGUAGE MODELS

- LLMs are typically trained on massive datasets of text and code
- Large language models transform natural language processing
- Capable of generating human-like text
- Learn patterns, context, and semantic relationships in data
- GPT/BARD are state-of-the-art models using LLMs
- Revolutionizing various applications across industries

TYPICAL APPLICATIONS OF LLMS

- Natural language understanding
- Natural language generation
- Machine translation
- Question answering
- Text summarization
- code generation

LANGUAGE UNDERSTANDING

- Comprehends and interprets text with high accuracy
- Captures meaning, context, and nuances in language
- Understands complex queries and generates relevant responses
- Enables chatbots and virtual assistants to communicate effectively
- Enhances machine translation and language comprehension tasks

CREATIVE CONTENT GENERATION

- Generates coherent and contextually appropriate text.
- Creates engaging and original stories, articles, and scripts.
- Assists with content creation for marketing and advertising.
- Provides innovative solutions for writing, gaming, and entertainment.
- Expands creative possibilities for artists and content creators.

KNOWLEDGE EXPANSION

- Accesses vast amounts of information from the Internet.
- Processes and summarizes complex texts and articles.
- Helps in research, data analysis, and decision-making.
- Provides insights and answers to specific questions.
- Enables continuous learning and knowledge sharing

.

REAL WORLD APPLICATIONS

- Improves customer service through chatbots and support systems.
- Personalizes user experiences in e-commerce and recommendation systems.
- Enhances medical diagnostics and healthcare decision-making.
- Powers virtual assistants and voice-activated technologies.
- Transforms education and online learning platforms.

HOW DO LARGE LANGUAGE MODELS WORK?

- Uses transformer architecture
- Learns long-range dependencies between words
- Generalize the statistical regularities of human language

DO I KNOW MY NEIGHBOURS?

Traditional domain in an windowed (ramp or otherwise) approach

- Frequency count - cooccurence
- Correlation

SVD Domain

- The transformation of the incidence matrix into SVD domain leads to the define the relationship between two words based on the magnitude of the diagonal matrix Σ

$$A = \sum_{j=1}^{r} \sigma_j u_j v_j^T \tag{1}$$

The low-rank approximation captures as much information/energy as possible with respect to the word relationships

GloVe Domain

- Ratio of the co-occuring word probabilities are transformed into domain

Useful in

- Auto-generation of sentences
- Auto-completion of sentences
- Creating contextualized word-vectors
- Machine Translation

PRE-TUNED AND FINE-TUNED

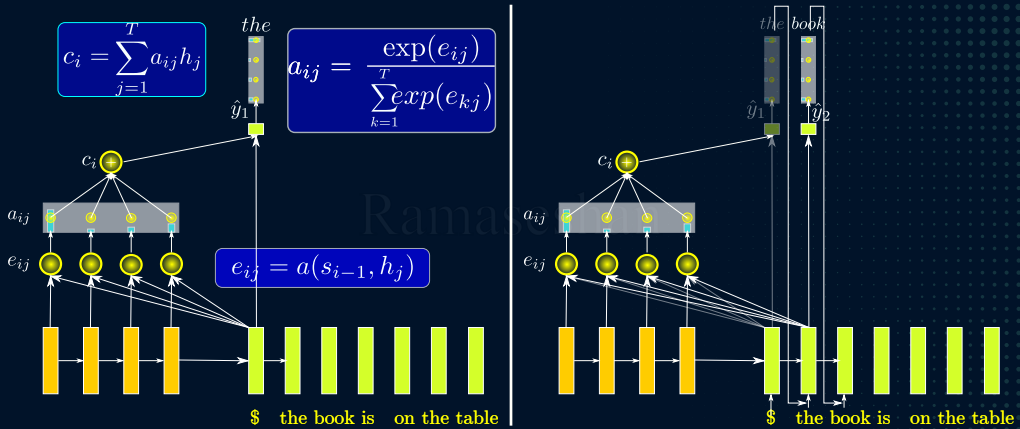
TRANSFER LEARNING

- Models are trained on a specific task - Word-embedding $\rightarrow$ build contextual embedding
- Use the learned weights for another task - attention in machine learning - transfer the hidden layer (contextual information) and use it for another task - mostly done using bidirectional LSTM - Embeddings from Language Model (ELMO)
  - Train using a sentence - forward and reverse Sequence
  - Concatenate hidden layers of (1)
  - Depending on the task on hand, multiply each (2) vector by a scale factor
  - Sum (2) to get the contextualized embedding

  $\rightarrow ELMO_k = \gamma \sum_{j=0}^{L} SoftMax_j h_{j,k}$

  where $h_{j,k} = BiLSTM(w_{1:T}, k), k$ is the index of the word and $1 \leq j \leq L$

ATTENTION

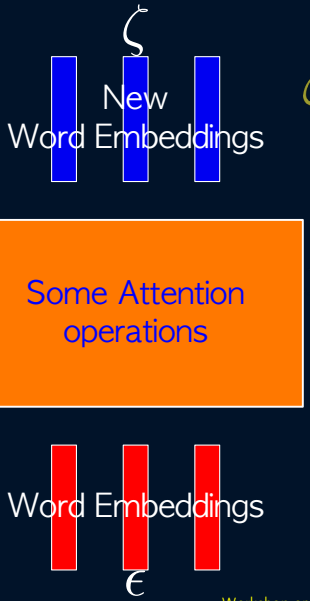

$c_i = \sum_{j=1}^{T} a_{ij} h_j$
$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} exp(e_{kj})}$
$e_{ij} = a(s_{i-1}, h_j)$
the
$\hat{y}_1$
c_i
a_{ij}
e_{ij}
$ the book is on the table
the book
$\hat{y}_2$

- How much should I be influenced by my neighbors?
- How I capture information about my neighbours?
- Ideally a model creates a word vector that has enough information about its neighbours irrespective of how well it is connected with them statistically

ANY CHANGE REQUIRED?

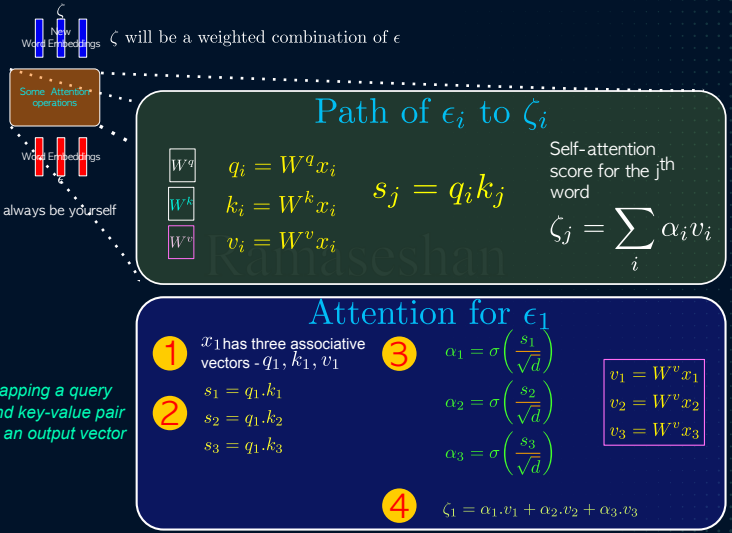
- Is it possible to encode the time-series differently?
- How do I reduce the complexity of recurrence?
- Is it possible to look at each embedding differently?

IDEAL ENCODING OF EMBEDDING



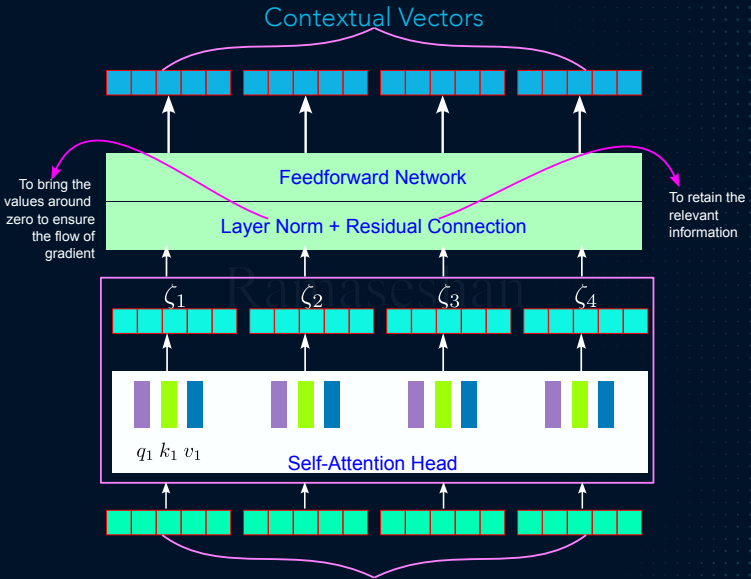
ζ will be the weighted combinations of ϵ

SELF-ATTENTION LAYER

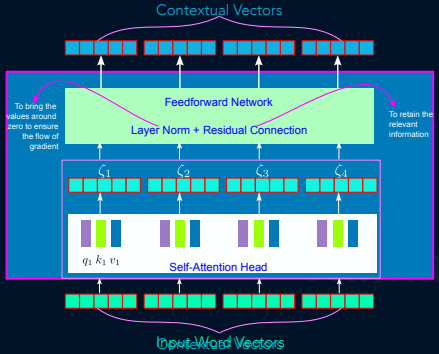


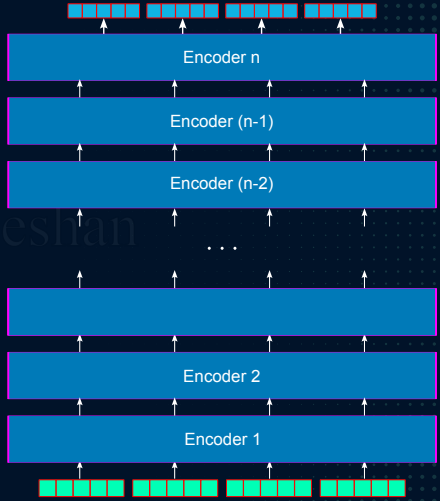
Mapping a query and key-value pair to an output vector

SINGLE TRANSFORMER LAYER



TRANSFORMER ENCODER





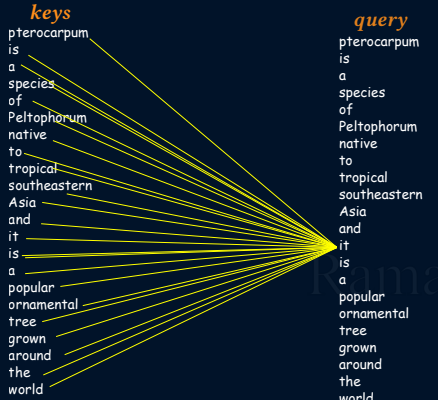
SELF-ATTENTION

Peltophorum pterocarpum is a species of **Peltophorum**, native to tropical southeastern Asia and **it** is a popular ornamental tree grown around the world

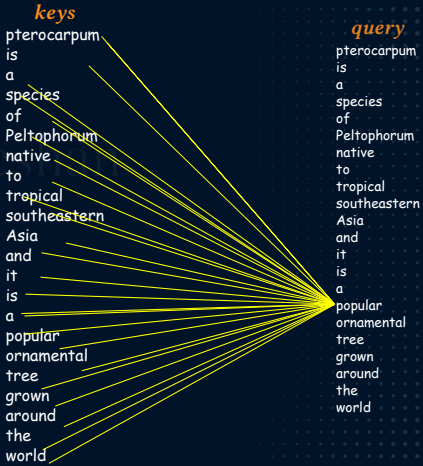
What does it refer to?

Self-attention allows us to associate *it* with Peltophorum pterocarpum

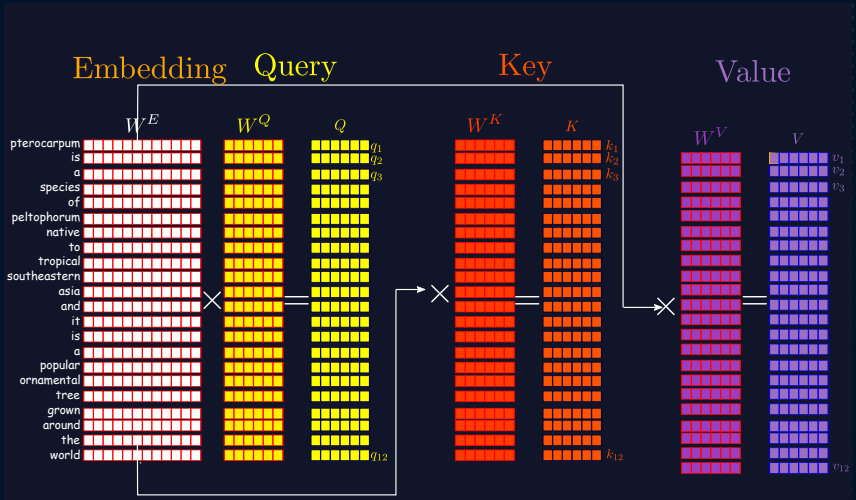
SELF-ATTENTION - WORD LEVEL



Self-attention allows each word to align itself to other words using their positions and looks for clues for a better contextul encoding

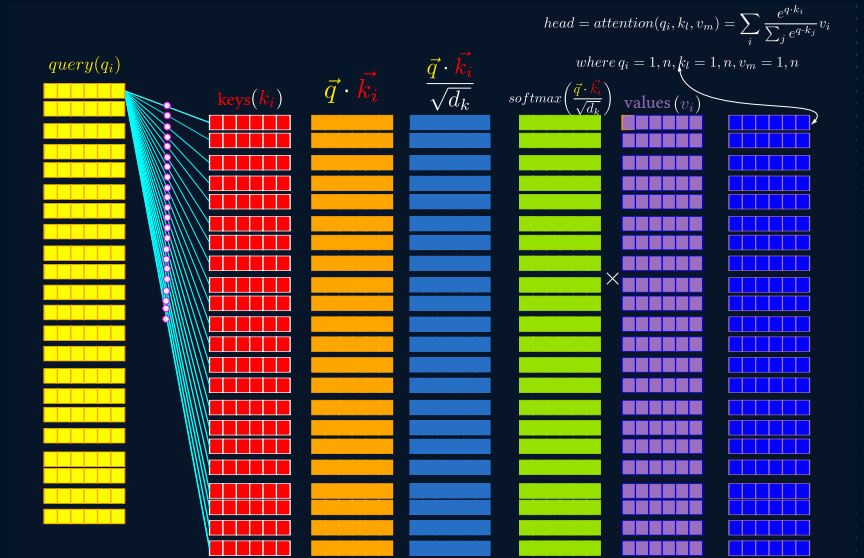


CREATING QUERY, KEY AND VALUES



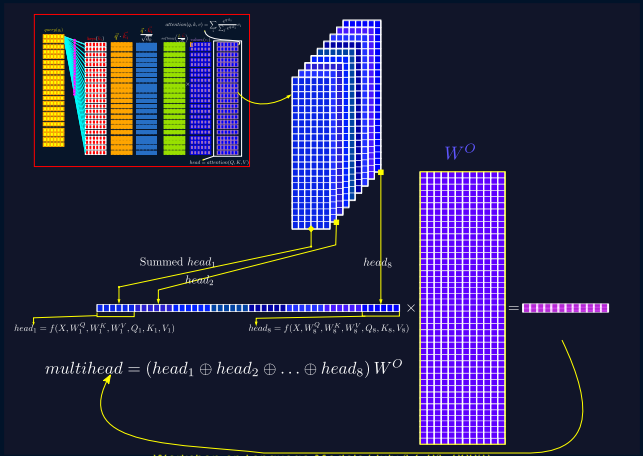
W^Q, W^K and W^V are learned during the training process

COMPUTING ATTENTION(Q,K,V)

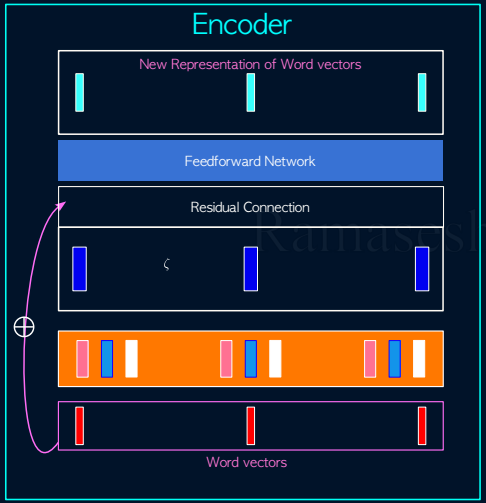


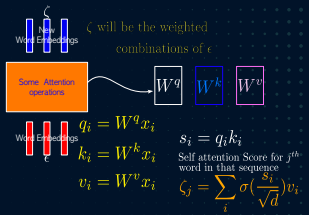
MULTI-HEAD ATTENTION

The summed attention score, $attention(q,k,v) = \sum_i \frac{e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}} v_i$, has information about all the word, but may have more information about words that have similarity scores higher than others in that context.

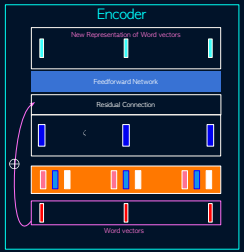


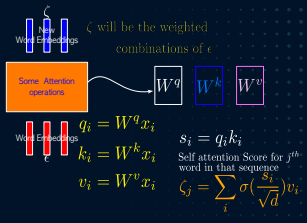
SELF-ATTENTION

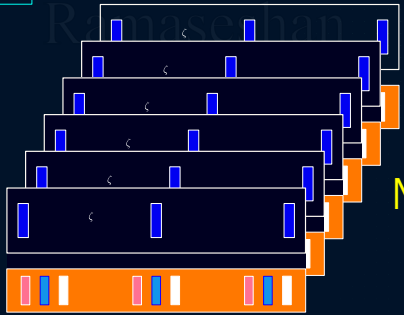




MULTIHEAD ATTENTION







Multihead

POSITIONAL ENCODING

- RNN models encode the time signal in a sequential manner
- Positional encoding is important for contextual learning
- Transformers a separate positional vector is added to the embedding
- Even and odd positional encoding using sin and cos is given as below:

$$PE_{pos,2i} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{pos,2i+1} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

- Another approach - Sinusoidal encoding

$PE_t = [sin(\omega_1 t); sin(\omega_2 t); \ldots sin(\omega_d)t]$, where $\omega_i = \frac{1}{10000^{i/d}}$

POSITIONAL ENCODING

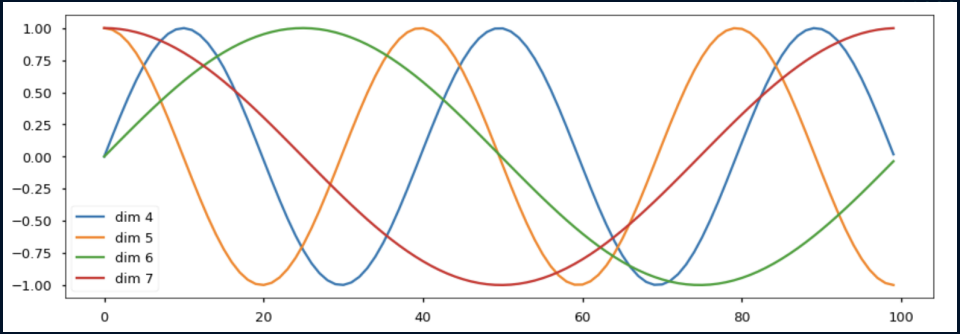


Figure: The frequency and offset of the wave is different for each dimension [1]

[1] Positional Encoding

WORD EMBEDDING

1. Word embedding is fundamental to Deep Learning of NLP
2. Contextual word embedding is now used in most of the downstream applications

CLOZE TEST

According to a report in yesterday's newspaper (1)——— police dog was taken to Raj Bhavan (2)——— Monday. This was to trace the (3)——— of the "very important horse" which (4)——— reported missing on Sunday. The dog picked (5)——— the scent on some traces of (6)——— and ran a few yards before losing the (7)———. The police have launched a vigorous (8)——— into the whole affair. They have (9)——— the services of a forensic expert, (10)——— fingerprint expert and a photographer. (11)——— are now fourteen horses at Raj Bhavan (12)——— are kept in a large shed near the gate.

1	once	a	new
2	at	next	on
3	police	killers	dogs
...	...	...	...
11	There	We	So
12	who	were	which

The purpose of the Cloze test is to measure the reading comprehension of a student with respect to grammar, usage, vocabulary, and contextual understanding.

MASKED LM

Objectives

- Predict the masked word using the context
- Combine left and right contexts to predict the masked word
- Use sentence pairs to predict the next sentence
- Use the trained model for token-level and sentence-level tasks
- Mask some of the tokens from the input
- Predict the original token using its context
- Use Bidirectional deep transformer model fuse the left and right context and develop a contextual representation

HOW DO WE MASK?

- 15% of the words are randomly masked
- Perform the following operation on the the i^{th} token
  1. Replace with a $<MASK>$ 80% of the time
  2. Replace with a random token 10% of the time
  3. Retain the original 10% of the time

BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT) ARCHITECTURE

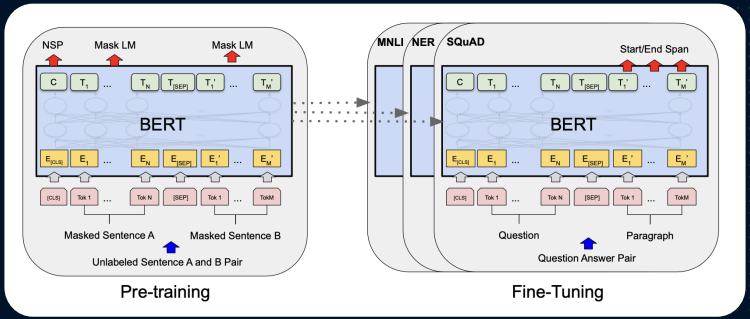


Figure: Overall pre-training and fine-tuning procedures for BERT. Apart from output layers, the same architec- tures are used in both pre-training and fine-tuning. The same pre-trained model parameters are used to initialize models for different down-stream tasks. During fine-tuning, all parameters are fine-tuned. [CLS] is a special symbol added in front of every input example, and [SEP] is a special separator token (e.g. separating ques- tions/answers).

GENERAL PURPOSE TRANSFORMER - GPT

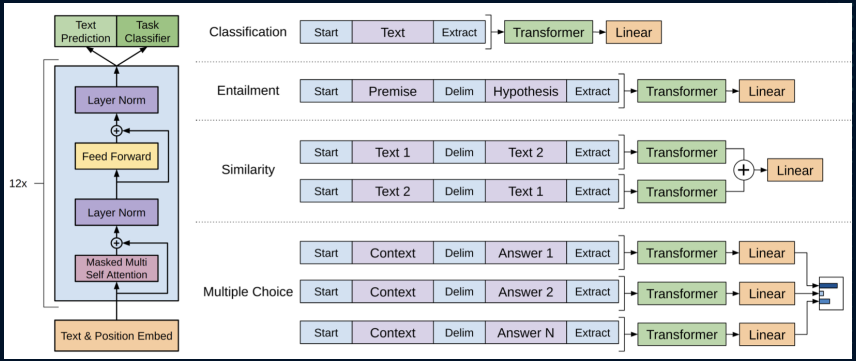
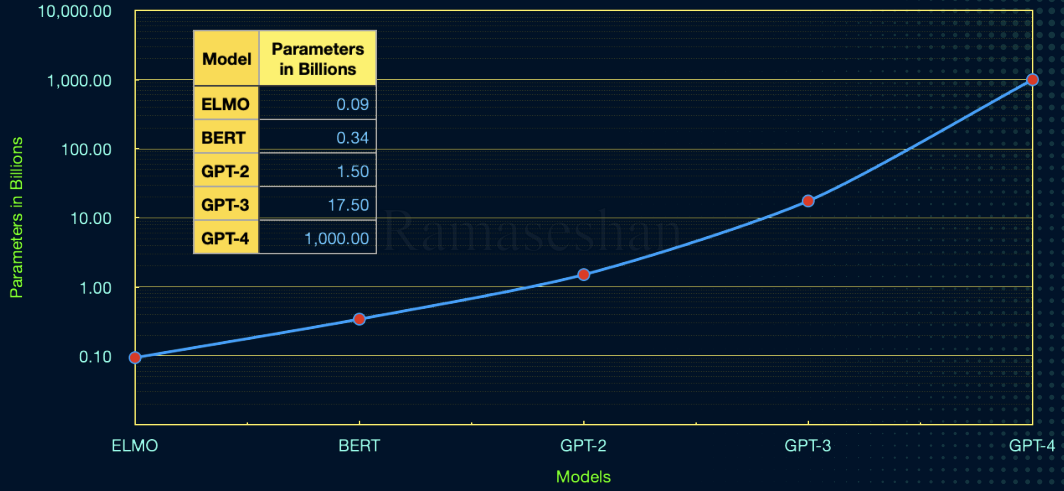


Figure: (left) Transformer architecture and training objectives used in this work. (right) Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer[2]

[2] GPT

PARAMETERS OF LARGE LANGUAGE MODEL



Model	Parameters in Billions
ELMO	0.09
BERT	0.34
GPT-2	1.50
GPT-3	17.50
GPT-4	1,000.00

LLAMA

Llama-2 was pretrained on publicly available online data sources. The fine-tuned model, Llama-2-chat, leverages publicly available instruction datasets and over 1 million human annotations.

- LLaMA stands for Large Language Model Meta AI.
- Llama 2 pretrained models are trained on 2 trillion tokens

Here is the link to the paper

MODEL DETAILS

	Training Data	Parameters	Content Length	GQA[3]	Tokens	Learning Rate
Llama 2	Publicly available online data	7B	4k	NA	2.0T	3.0×10^4
Llama 2	”	13B	4k	NA	2.0T	3.0×10^4
Llama 2	”	70B	4k	YES	2.0T	1.5×10^4

TODELETE GQA

[3] Grouped-Query Attention

Ramaseshan

CHALLENGES

- Computationally expensive to train and deploy
  - Facebook's 65B LLaMA model - Trained for 21 days on 2048 Nvidia A100 GPUs, at a cost of $\approx$ 4 million USD
  - Google's 540B PaLM was trained on 6144 v4 TPUs for 1200hrs, at a cost of $\approx$ $27 million USD
  - Cost of training ChatGPT 4 is estimated to be over $100 million USD
    - Costs include data preparation (usually around $10M), cost of computing power and cost of engineering staff & administration expenses)
    - Power used to train ChatGPT 4 is $\approx$ 1.287 gWH. Average Indian house-hold consumtion per year is 1200 kWH.
- Can be biased, reflecting the biases of the data they are trained on
- May generate harmful or offensive content.

FAIRNESS AND INCLUSION

Everyone should be treated fairly when using our products and they should work equally well for all people.

ROBUSTNESS AND SAFETY

- Should meet high performance standards
- Should be robust and safe, with rigorous testing and validation to avoid unintended negative consequences

TRANSPARENCY AND ACCOUNTABILITY

- Developers and users of the system should be responsible for its actions
- Rules and regulations governing the use of AI systems should be transparent,
- Mechanisms should be in place to investigate and punish misuse of the system

EXPLAINABILITY

- Should enable users to understand the underlying processes to build trust
- System should explain its decisions in a clear and understandable way
- Open for scrutiny


THANK YOU