

# Deep Sequence Models for Search Ranking - Session II

Tutorial Link: <https://dlranking.github.io/dlrr/>

Data source: <https://huggingface.co/datasets/xglue>

XGLUE: A New Benchmark Dataset for Cross-lingual Pre-training, Understanding and Generation)

Presenters: Moumita Bhattacharya, Abby Liu, Linsey Pang

Notebooks: Abby Liu, Linsey Pang

Date: August 14th, 2022

# Agenda

- Brief History of Deep Learning Models For Search Ranking
- Sequence Models
  - RNN, LSTM
  - Attention, Self Attention
- Transformer Architecture
- Deep Classifier
- Deep Siamese Network
- Hands-on Session

# **Brief History of Deep Neural Network For Search Ranking**

## Key responsibilities of a search engine

**Indexing** — Ingesting and storing data efficiently so that it can be retrieved quickly

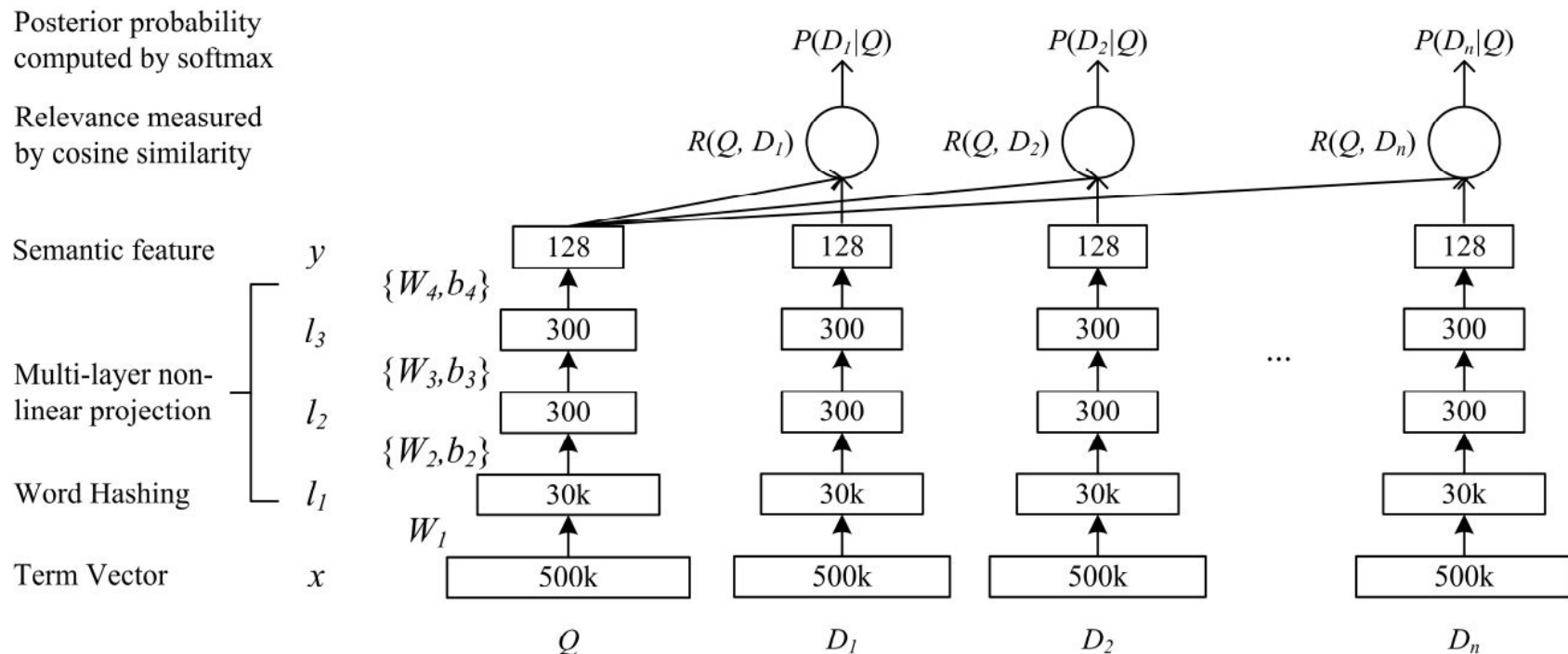
**Querying** — Providing retrieval functionality so that search can be performed by an end user

**Ranking** — Presenting and ranking the results according to certain metrics to best satisfy users' information needs

## Brief History of Neural IR

- Neural IR resort to deep learning for tackling the feature engineering problem of learning to rank, by directly using only automatically learned features from raw text of query and document. □
- There have been some pioneer work, including DSSM [[ref](#)], and CDSSM [[ref](#)].
- Both DSSM and CDSSM directly apply deep neural networks to obtain the semantic representations of query and document, and the ranking score is produced by computing their cosine similarity.
- Studies such as DRMM [[ref](#)] and DeepRank [[ref](#)] argued that DSSM and CDSSM only consider the semantic matching between query and document, but ignored relevance matching as well as modeling the relevance generation process.

# Learning Deep Structured Semantic Models for Web Search using Clickthrough Data (DSMM: 2013)



**Figure 1: Illustration of the DSMM.** It uses a DNN to map high-dimensional sparse text features into low-dimensional dense features in a semantic space. The first hidden layer, with 30k units, accomplishes word hashing. The word-hashed features are then projected through multiple layers of non-linear projections. The final layer's neural activities in this DNN form the feature in the semantic space.

# A Deep Relevance Matching Model for Ad-hoc Retrieval (DRMM: 2017)

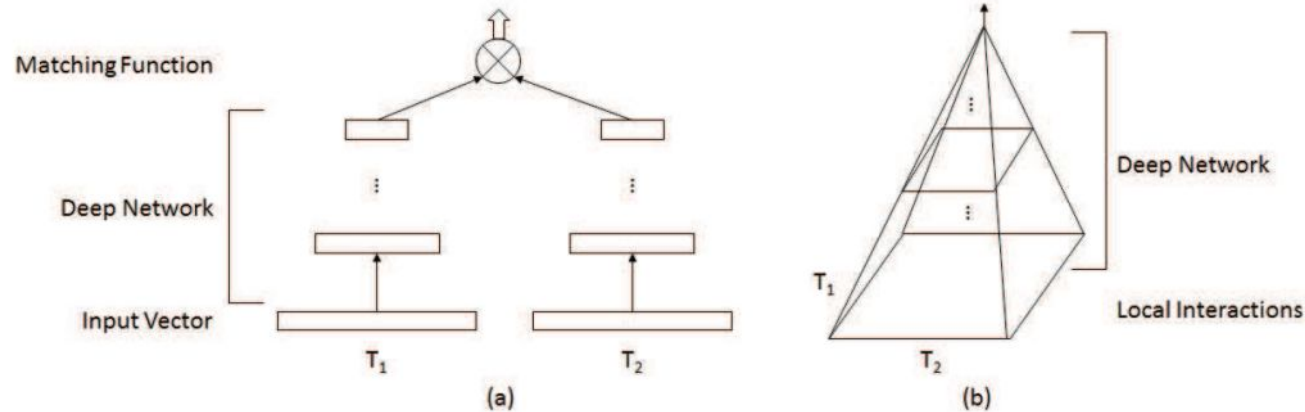


Figure 1: Two types of deep matching models: (a) Representation-focused models employ a Siamese (symmetric) architecture over the text inputs; (b) Interaction-focused models employ a hierarchical deep architecture over the local interaction matrix.

(Ref: [link](#))

# DeepRank: A New Deep Architecture for Relevance Ranking in Information Retrieval (2019)

Ref: ([link](#))

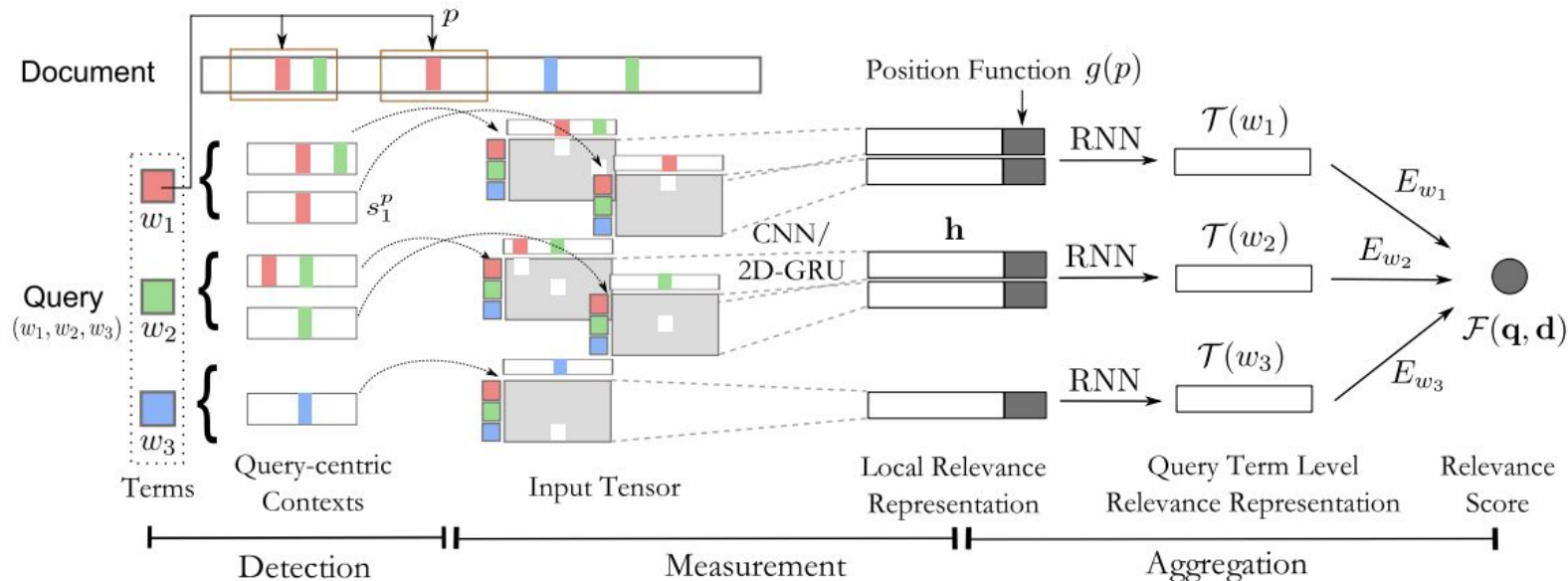


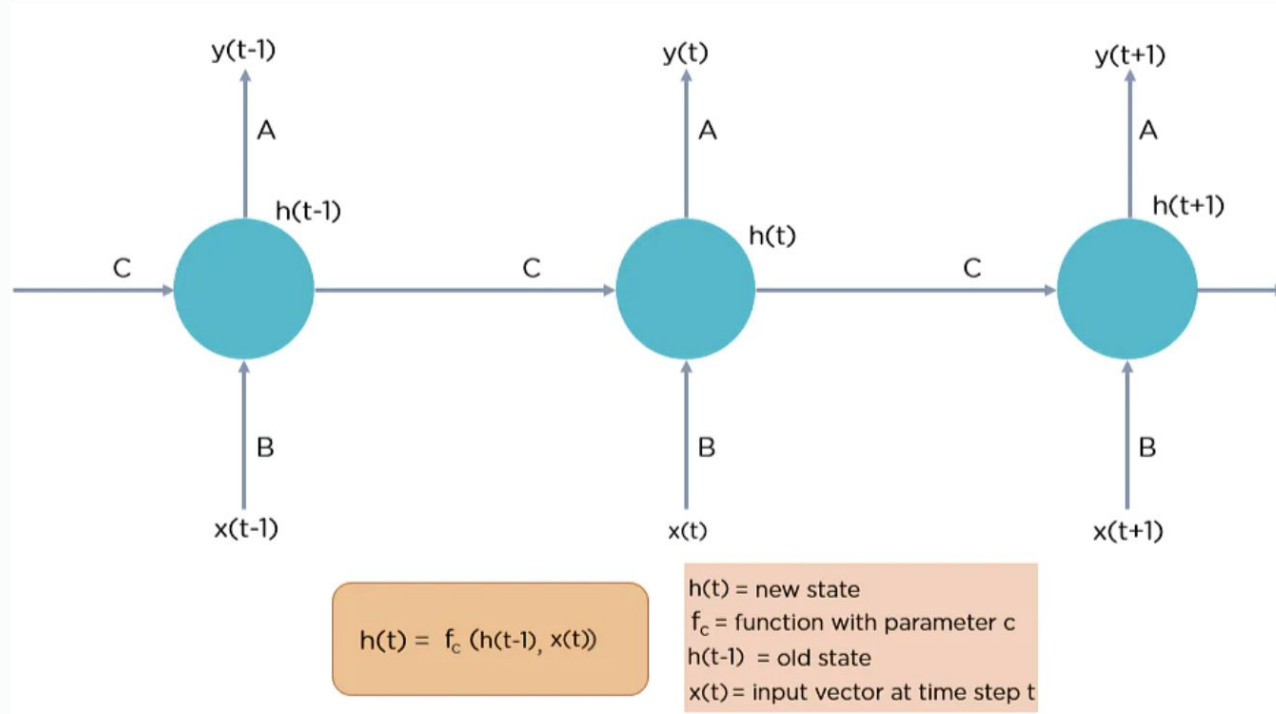
Figure 1: An illustration of DeepRank.



# Sequence Models

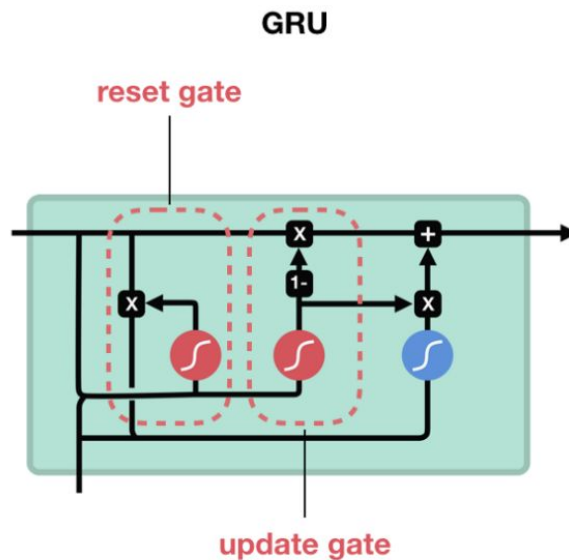
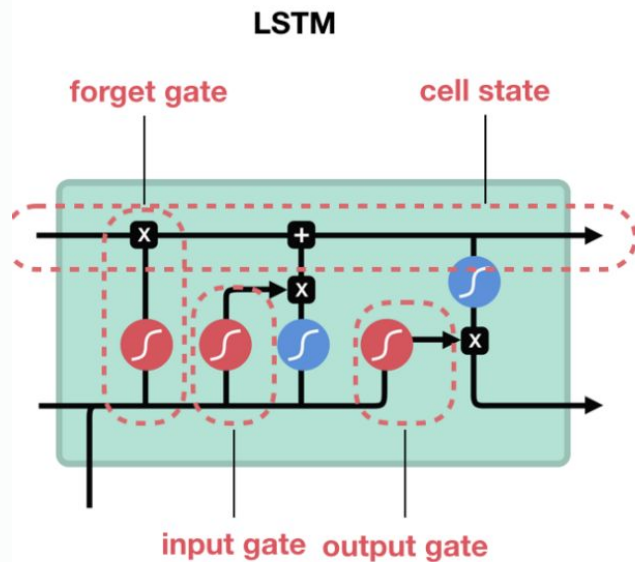
- RNN
- GRU
- LSTM
- Attention
- Transformer

# Recurrent Neural Network



```
torch.nn.RNN(input_size, hidden_layer, num_layer, bias=True,  
batch_first=False, dropout = 0, bidirectional = False)
```

# GRU and LSTM



sigmoid



tanh



pointwise  
multiplication



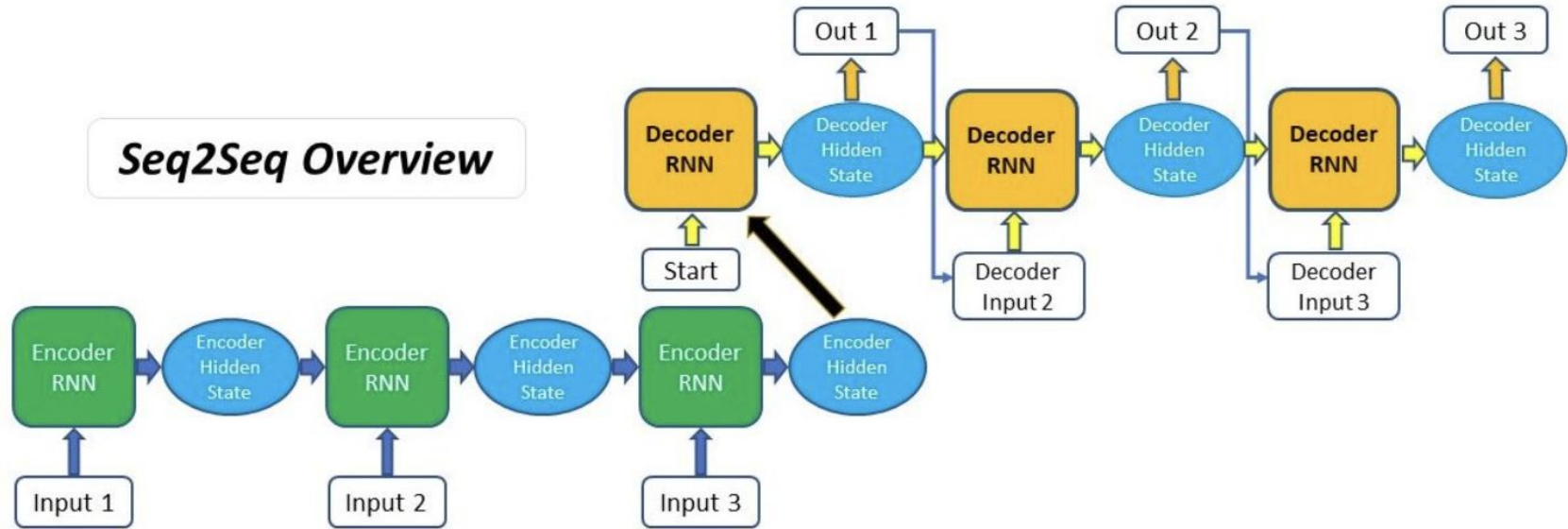
pointwise  
addition



vector  
concatenation

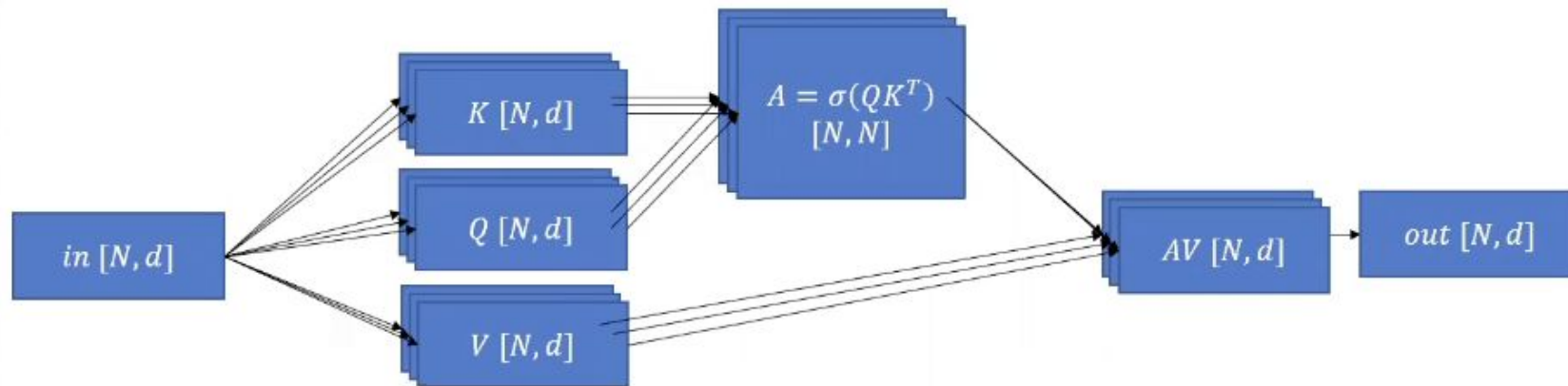
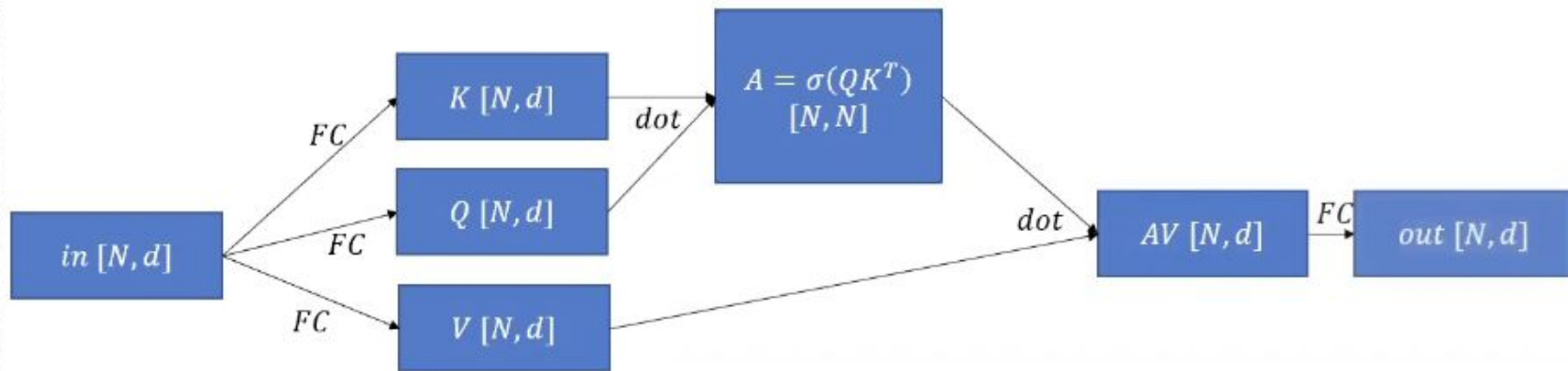
Ref: [blog](#)

# Encoder and Decoder for Seq2Seq

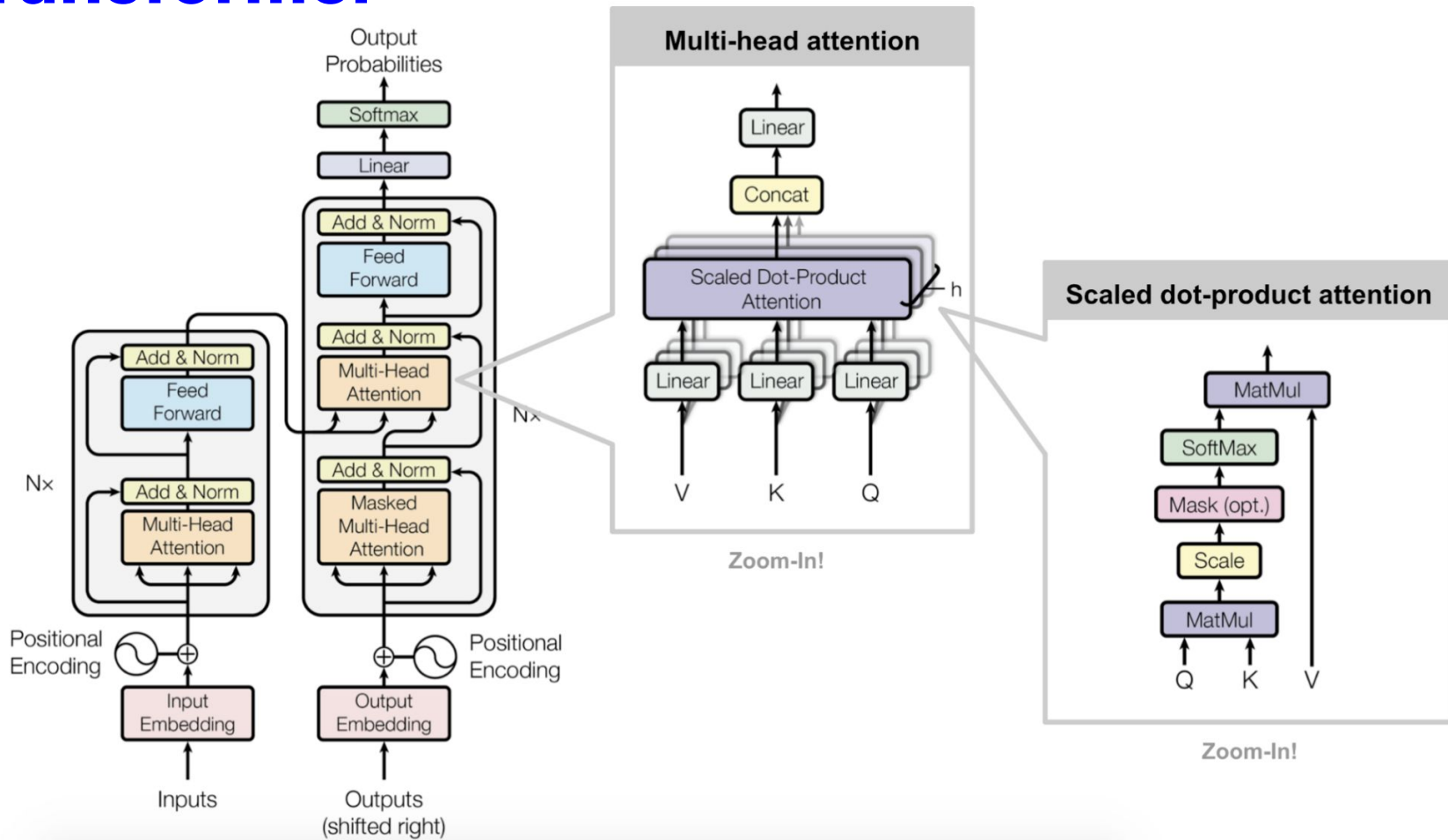


# Attention and Self Attention

# Single and Multi-Head Attention



# Transformer



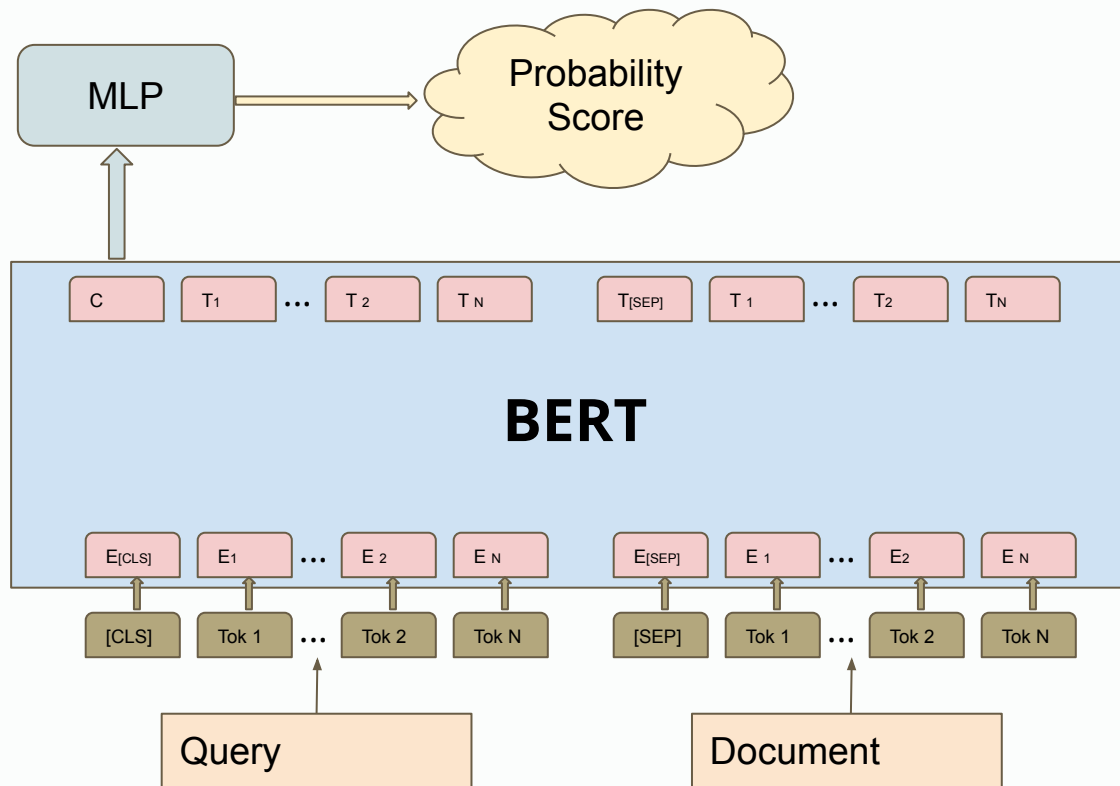
# Deep Classifier Network



# Deep Classifier Network

- The goal is to rank the documents based on the query document similarity.
- Use the BERT model to generate the embeddings of the queries and embeddings of the documents that capture the semantic meanings of them from the text.
- Concatenate the query and document embeddings, and feed into a classification network to calculate the probability of the document related to the query
- Rank the documents based on their probability score to the query

# Deep Classifier Network



# Deep Classifier Network

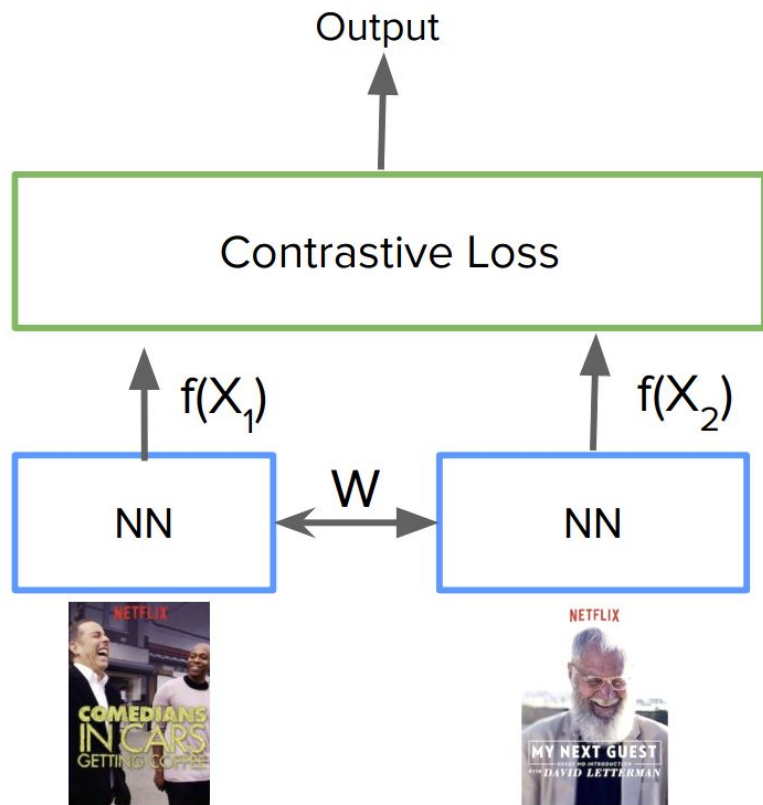
[Colab Demo](#)

# Deep Siamese Network

# Deep Metric Learning

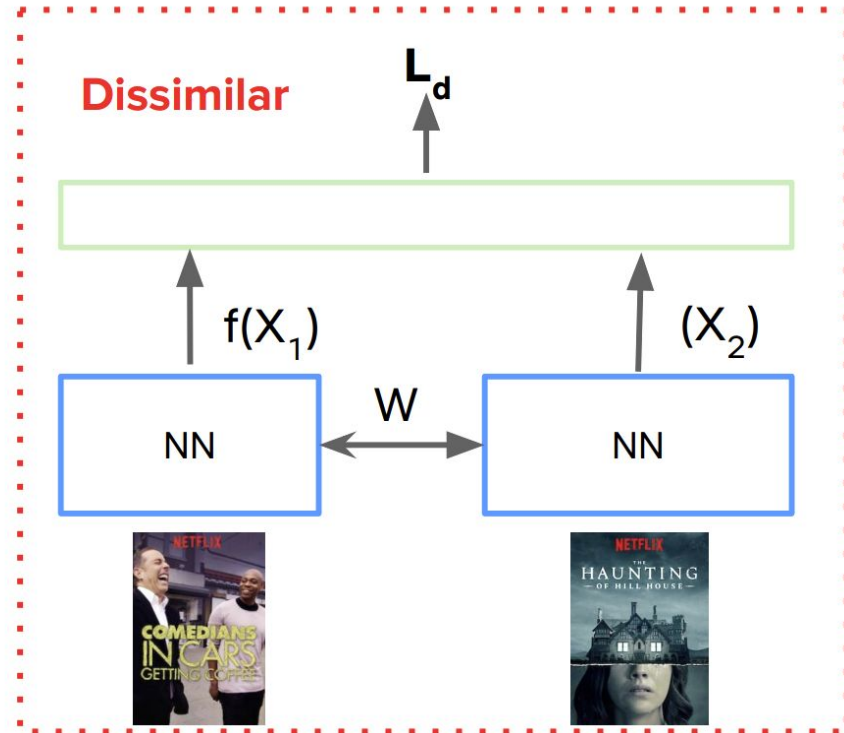
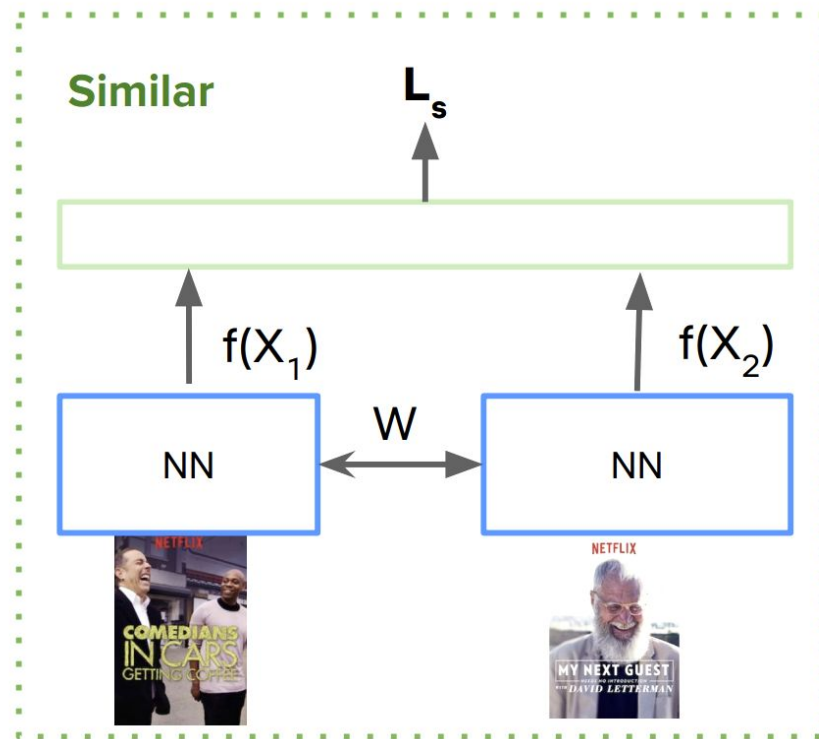
- The goal is to capture similarity between embeddings, such that the projected distance of similar items in the embedding space is smaller than the dissimilar items.
- Compared to the standard distance metric learning, it uses deep neural networks to learn a nonlinear mapping to the embedding space
- Helps with extreme classification settings with huge number classes, not many examples per class

# Siamese Networks



- Left and right legs of the network have identical structures (siamese)
- Weights are shared between the siamese networks during training
- Networks are optimized with a loss function, such as contrastive loss

# Siamese Networks



$$L = \sum_s L_s + \sum_d L_d$$

Total loss is the sum of losses on similar pairs and dissimilar pairs

# Contrastive Loss

The goal is to minimize  $L$  with respect to  $W$ , such that  $D_W$  is small for similar pairs, and large for dissimilar pairs

$$L^i = (1 - Y)L_s(D_W^i) + YL_d(D_W^i)$$

$Y$  is set to 0 if pairs are similar otherwise 1

Exact loss is defined as:

$$L = (1 - Y)\frac{1}{2}(D_W)^2 + Y\frac{1}{2}\{\max(0, m - D_W)\}^2$$

$m > 0$  is the margin



# Triplet Loss

The goal is the same, the distance between similar items must be low, and dissimilar items must be high.

$$L = \max(D(a, p) - D(a, n) + \alpha, 0)$$

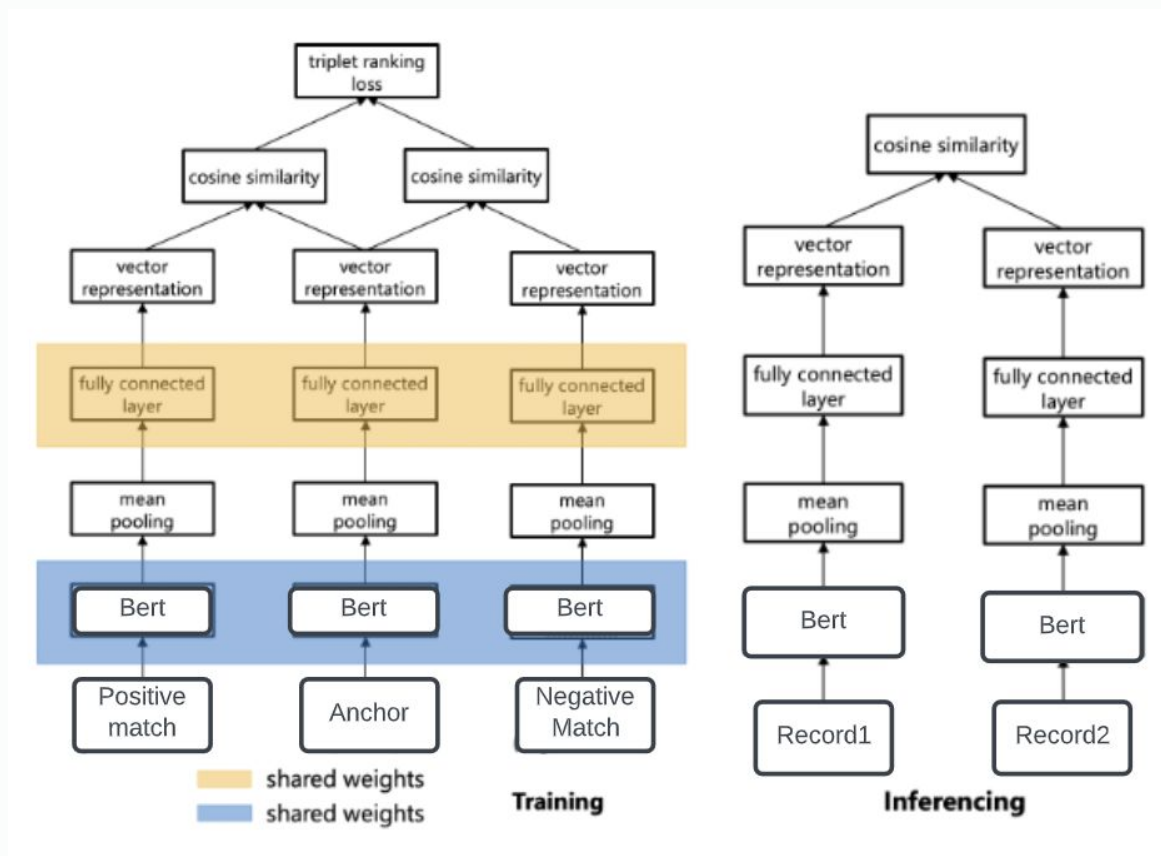
**Loss of a triplet (a, p, n)**

When using an Euclidean distance:

$$L = \frac{1}{Z} \sum_i^N [\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha]_+$$

$[\cdot]_+ = \max(0, \cdot)$   
Hinge loss function

# Deep Siamese Network



Triplet Loss Architecture

# Deep Siamese Network

Similarity function:

$$s(x, y) = \frac{e(x) \cdot e(y)}{\|e(x)\| \|e(y)\|}$$

where  $e()$  is the function that projects the raw input to an embedding vector (the sub-network in the Siamese network) and  $\|e(x)\|$  denotes the norm of the vector  $e(x)$ .

Cost function:

$$L_{triplet} = \frac{1}{|X|} \sum_{(a,p,n) \in X} \max(|s(a,p) - s(a,n) + \alpha|, 0)$$

where  $X$  is the set of  $(a, n, p)$  triplets and  $\alpha$  is a margin between positive and negative pairs. The triplet loss pushes  $s(a, p)$  towards 1 and pushes  $s(a, n)$  below than  $s(a, p)$  by at least  $\alpha$ . We set  $\alpha = 0.3$  for the output shown in this work.

# Deep Siamese Network

[Demo](#)

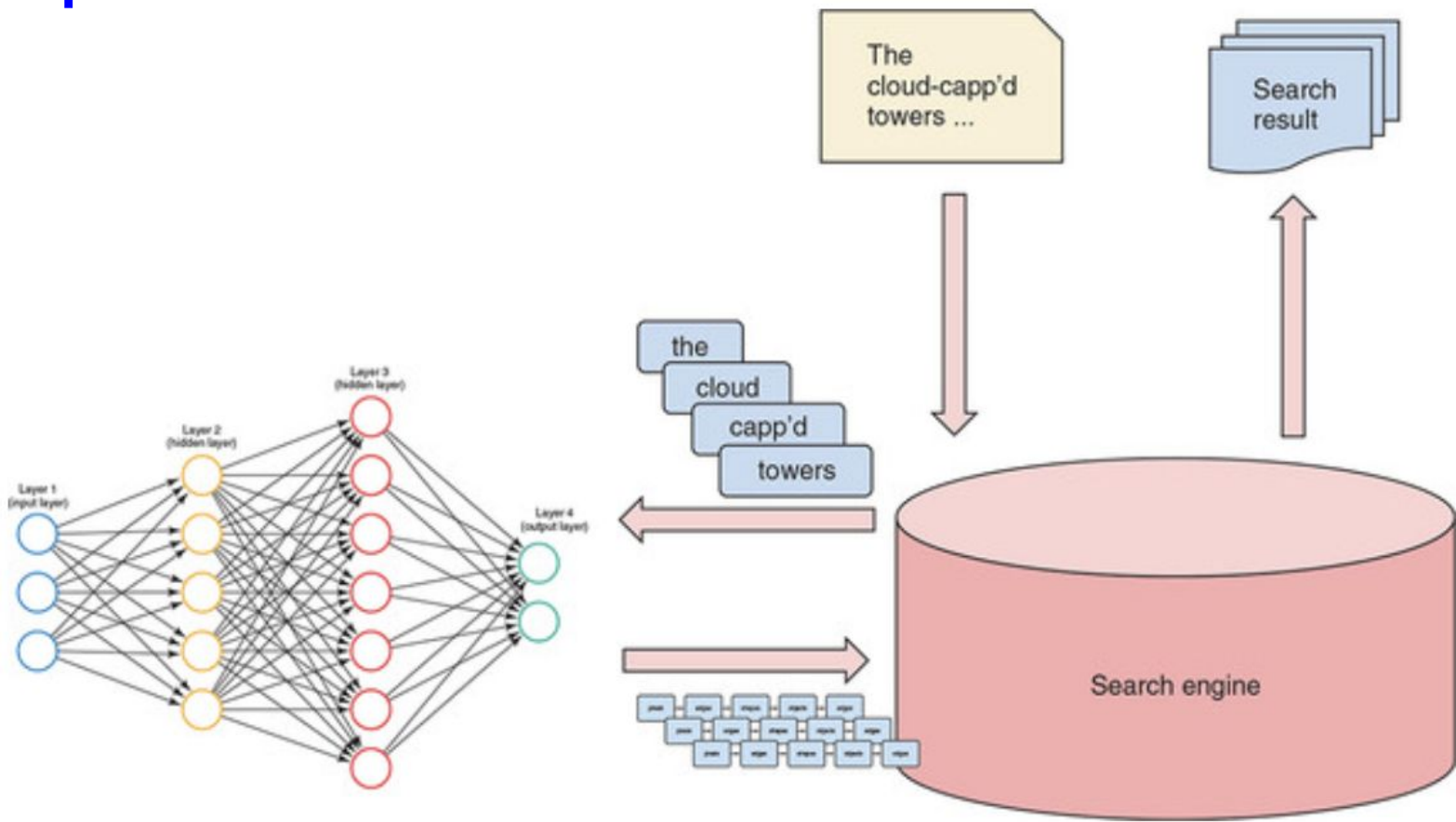


Let's work on the hands on examples now!!

## Neural IR

- Providing a representation of textual data that captures word and document semantics, allowing a machine to say which words and documents are semantically similar
- Under certain assumptions, approximating any function. There's theoretically no limit to the kinds of tasks deep neural networks can achieve

# Deep Classifier



Ref: ([link](#))

# Types of Attentions (Bahdanau vs Luong)

