DATA SCIENCE INTERVIEW PREPARATION

(30 Days of Interview Preparation)

DAY 17

Q1. What is ERM (Empirical Risk Minimization)?

Answer:

Empirical risk minimization (ERM): It is a principle in statistical learning theory which defines a family of learning algorithms and is used to give theoretical bounds on their performance. The idea is that we don't know exactly how well an algorithm will work in practice (the true "risk") because we don't know the true distribution of data that the algorithm will work on, but as an alternative we can measure its performance on a known set of training data.

We assumed that our samples come from this distribution and use our dataset as an approximation. If we compute the loss using the data points in our dataset, it's called empirical risk. It is "empirical" and not "true" because we are using a dataset that's a subset of the whole population.

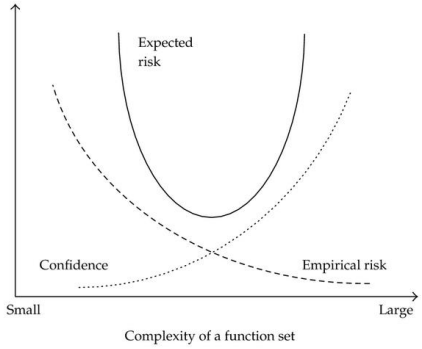
When our learning model is built, we have to pick a function that minimizes the empirical risk that is the delta between predicted output and actual output for data points in the dataset. This process of finding this function is called empirical risk minimization (ERM). We want to minimize the true risk. We don't have information that allows us to achieve that, so we hope that this empirical risk will almost be the same as the true empirical risk.

Let's get a better understanding by Example

We would want to build a model that can differentiate between a male and a female based on specific features. If we select 150 random people where women are really short, and men are really tall, then the model might incorrectly assume that height is the differentiating feature. For building a truly accurate model, we have to gather all the women and men in the world to extract differentiating features. Unfortunately, that is not possible! So we select a small number of people and hope that this sample is representative of the whole population.

$$R_{emp}[f] = \sum_{x \in X} \sum_{j=1}^{2} c(x, y_j, f(x)) p_{emp}(y_j, x)$$

$$= \frac{1}{n} \sum_{i=1}^{n} c(x_i, y_i, f(x_i))$$



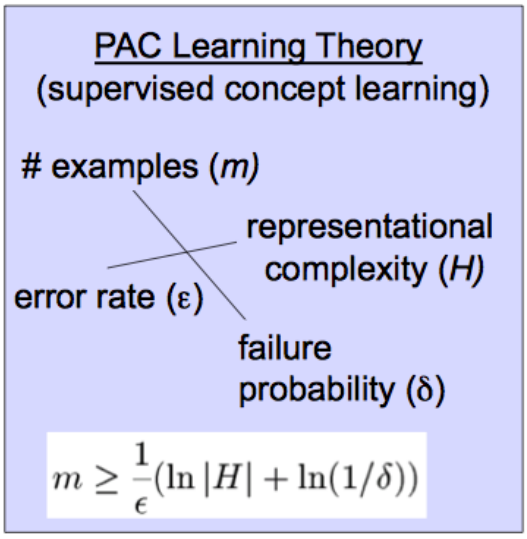
Q2. What is PAC (Probably Approximately Correct)?

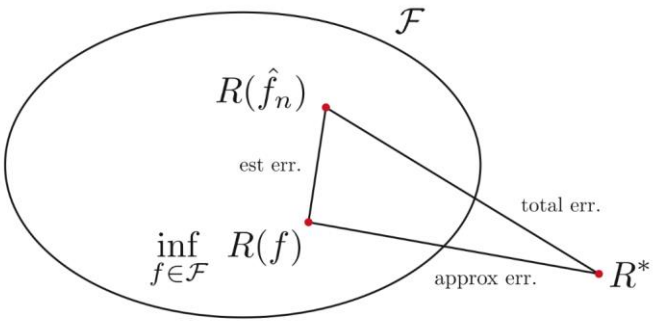
Answer:

PAC: In computational learning theory, probably approximately correct (PAC) learning is a framework for mathematical analysis of machine learning.

The learner receives samples and must have to pick a generalization function (called the *hypothesis*) from a specific class of possible functions. Our goal is that, with high probability, the selected function will have low generalization error. The learner must be able to learn the concept given any arbitrary approximation ratio, probability of success, or distribution of the samples.

Hypothesis class is PAC(Probably Approximately Correct) learnable if there exists a function **m_H** and algorithm that for any labeling function **f,** distribution **D** over the domain of inputs **X,** **delta** and **epsilon** that with **m ≥ m_H** produces a hypothesis **h** like that with probability 1-**delta** it returns a **true error** lower than **epsilon.** Labeling function is nothing other than saying that we have a specific function **f** that labels the data in the domain.





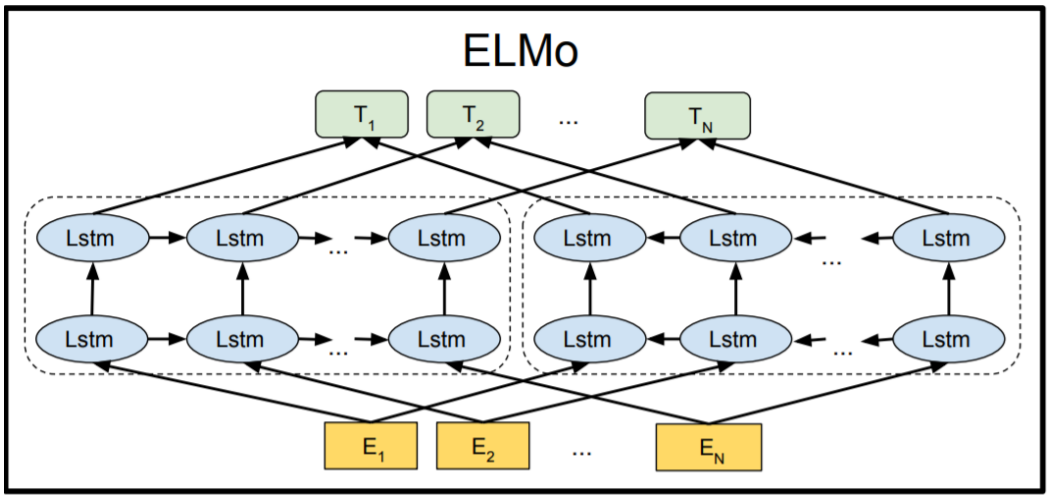
Q3. What is ELMo?

Answer:

ELMo is a novel way to represent words in vectors or embeddings. These word embeddings help achieve state-of-the-art (SOTA) results in several NLP tasks:

Task	Previous SOTA		ELMo + Baseline
SQuAD	SAN	84.4	85.8
SNLI	Chen et al (2017)	88.6	88.7 +/- 0.17
SRL	He et al (2017)	81.7	84.6
Coref	Lee et al (2017)	67.2	70.4
NER	Peters et al (2017)	91.93 +/- 0.19	92.22 +/- 0.10
Sentiment (5-class)	McCann et al (2017)	53.7	54.7 +/- 0.5

It is a deep contextualized word representation that models both complex characteristics of word use (e.g., syntax and semantics), and how these uses vary across linguistic contexts. These word vectors are learned functions of internal states of a deep biLM(bidirectional language model), which is pre-trained on large text corpus. They could be easily added to existing models and significantly improve state of the art across a broad range of challenging NLP problems, including question answering, textual entailment and sentiment analysis.



Q4. What is Pragmatic Analysis in NLP?

Answer:

Pragmatic Analysis(PA): It deals with outside word knowledge, which means understanding i.e external to documents and queries. PA that focuses on what was described is reinterpreted by what it actually meant, deriving the various aspects of language that require real-world knowledge.

It deals with overall communicative and social content and its effect on interpretation. It means abstracting the meaningful use of language in situations. In this analysis, the main focus always on what was said in reinterpreted on what is intended.

It helps users to discover this intended effect by applying a set of rules that characterize cooperative dialogues.

E.g., "close the window?" should be interpreted as a request instead of an order.

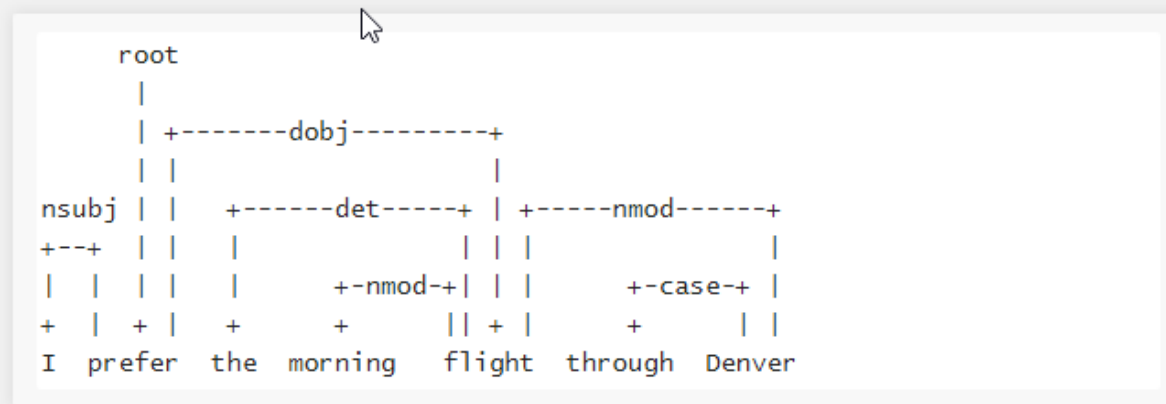
Personal	Organizational devices (Calls/Vocatives)
Request	Questions
Action	Answers/Responses
Permission	Repetition/Imitation
Offering/showing	Elicited identification
Descriptions	Routines
Statements (Internal reports)	Exclamations
Acknowledgements	Unclassified
Performatives	Double coded

Q5. What is Syntactic Parsing?

Answer:

Syntactic Parsing or **Dependency Parsing:** It is a task of recognizing a sentence and assigning a syntactic structure to it. Most Widely we used syntactic structure is the parse tree which can be generated using some parsing algorithms. These parse trees are useful in various applications like grammar checking or more importantly, it plays a critical role in the semantic analysis stage. For example to answer the question "*Who is the point guard for the LA Laker in the next game ?*" we need to figure out its subject, objects, attributes to help us figure out that the user wants the point guard of the LA Lakers specifically for the next game.

Example:



Q6. What is ULMFit?

Answer:

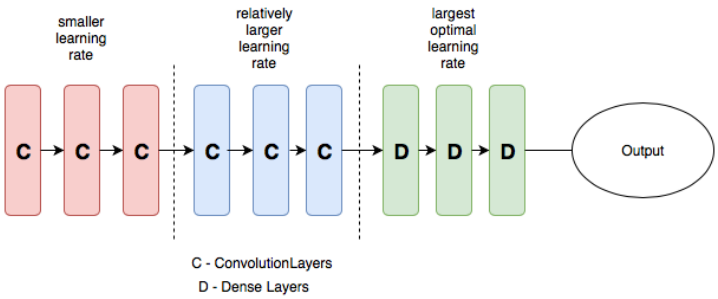
Transfer Learning in **NLP(Natural language Processing)** is an area that had not been explored with great success. But, in May 2018, **Jeremy Howard** and **Sebastian Ruder** came up with the paper – **Universal Language Model Fine-tuning for Text Classification(ULMFit)** which explores the benefits of using a pre trained model on text classification. It proposes **ULMFiT(Universal Language Model Fine-tuning for Text Classification),** a transfer learning method that could be applied to any task in NLP. In this method outperforms the state-of-the-art on six text classification tasks.

ULMFiT uses a **regular LSTM**which is the state-of-the-art language model architecture **(AWD-LSTM)**. The LSTM network has three layers. Single architecture is used throughout – for pre-training as well as for fine-tuning.

ULMFiT achieves the state-of-the-art result using novel techniques like:

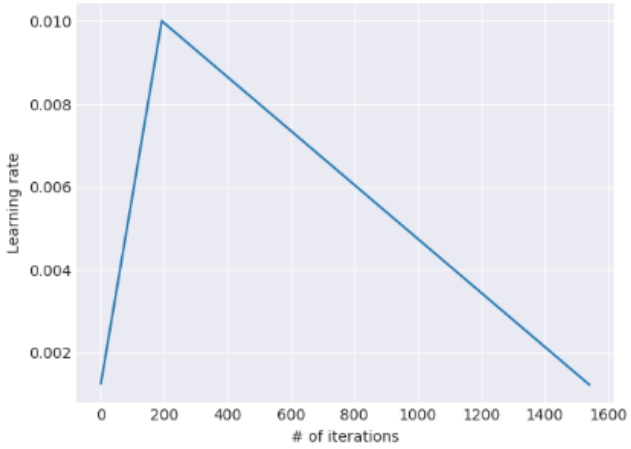
- Discriminative fine-tuning
- Slanted triangular learning rates
- Gradual unfreezing

Discriminative Fine-Tuning



Different layers of a neural network capture different types of information so they should be fine-tuned to varying extents. Instead of using the same learning rates for all layers of the model, discriminative fine-tuning allows us to tune each layer with different learning rates.

Slanted triangular learning



The model should quickly converge to a suitable region of the parameter space in the beginning of training and then later refine its parameters. Using a constant learning rate throughout training is not the best way to achieve this behaviour. Instead Slanted Triangular Learning Rates (STLR) linearly increases the learning rate at first and then linearly decays it.

Gradual Unfreezing

Gradual unfreezing is the concept of unfreezing the layers gradually, which avoids the catastrophic loss of knowledge possessed by the model. It first unfreezes the top layer and fine-tunes all the unfrozen layers for 1 epoch. It then unfreezes the next lower frozen layer and repeats until all the layers have been fine-tuned until convergence at the last iteration.

Q7. What is BERT?

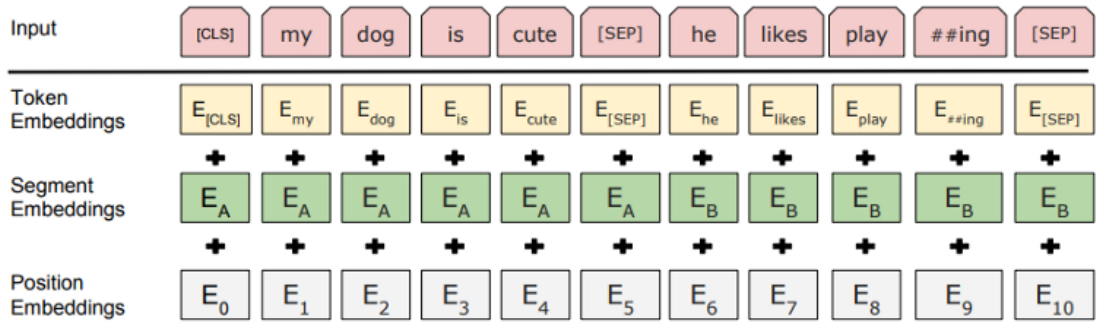
Answer:

BERT (Bidirectional Encoder Representations from Transformers) is an open-sourced NLP pre-training model developed by researchers at Google in 2018. A direct descendant to GPT (Generalized Language Models), BERT has outperformed several models in NLP and provided top results in Question Answering, Natural Language Inference (MNLI), and other frameworks.

What makes it's unique from the rest of the model is that it is the first deeply bidirectional, unsupervised language representation, pre-trained using only a plain text corpus. Since it's open-sourced, anyone with machine learning knowledge can easily build an NLP model without the need for sourcing massive datasets for training the model, thus saving time, energy, knowledge and resources.

How does it work?

Traditional context-free models (like word2vec or GloVe) generate a single word embedding representation for each word in the vocabulary which means the word **"right"** would have the same context-free representation in "I'm sure I'm right" and "Take a right turn." However, BERT would represent based on both previous and next context, making it bidirectional. While the concept of bidirectional was around for a long time, BERT was first on its kind to successfully pre-train bidirectional in a deep neural network.

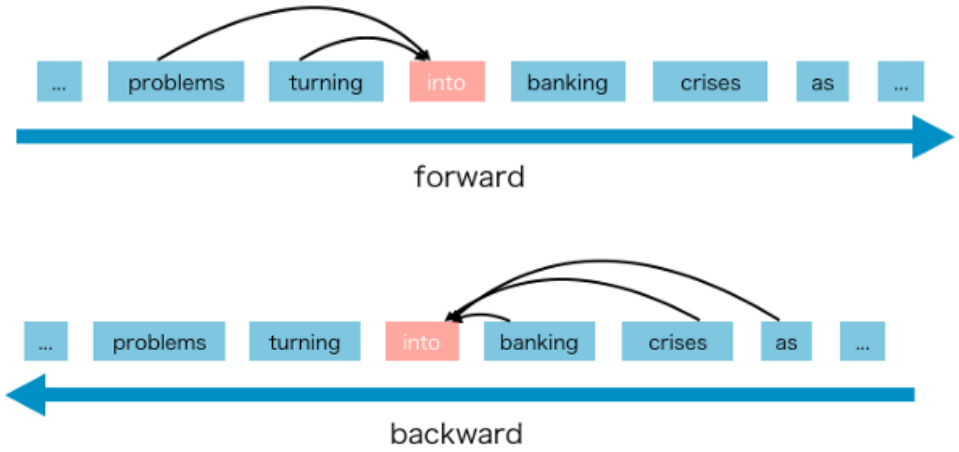


Q8.What is XLNet?

Answer:

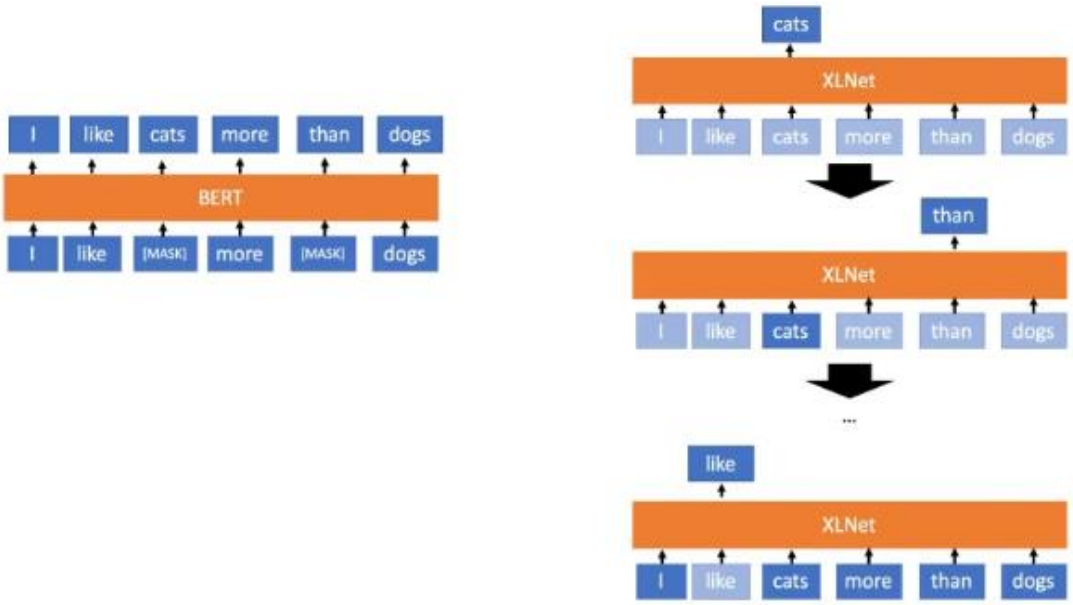
XLNet is a BERT-like model instead of a totally different one. But it is an auspicious and potential one. In one word, **XLNet is a generalized autoregressive pretraining method.**

Autoregressive (AR) language model: It is a kind of model that using the context word to predict the next word. But here the context word is constrained to two directions, either forward or backwards.



The advantages of AR language model are good at generative Natural language Process(NLP) tasks. Because when generating context, usually is the forward direction. AR language model naturally works well on such NLP tasks.

But Autoregressive language model has some disadvantages, and it only can use forward context or backward context, which means it can't use forward and backward context at the same time.



The conceptual difference between BERT and XLNet. Transparent words are masked out so the model cannot rely on them. XLNet learns to predict the words in an arbitrary order but in an autoregressive, sequential manner (not necessarily left-to-right). BERT predicts all masked words simultaneously.

Q9. What is the transformer?

Answer:

Transformer: It is a deep machine learning model introduced in 2017, used primarily in the field of natural language processing (NLP). Like recurrent neural networks(RNN), It is designed to handle ordered sequences of data, such as natural language, for various tasks like machine translation and text summarization. However, Unlike recurrent neural networks(RNN), Transformers do not require that the sequence be processed in the order. So, if the data in question is a natural language, the Transformer does not need to process the beginning of a sentence before it processes the end. Due to this feature, the Transformer allows for much more parallelization than RNNs during training.

Transformers are developed to ~~solve the problem of sequence transduction current neural networks~~. It means any task that transforms an input sequence to an output sequence. This includes speech recognition, text-to-speech transformation, etc.

For models to perform a sequence transduction, it is necessary to have some sort of memory. example, let us say that we are translating the following sentence to another language (French):

"The Transformers" is a Japanese band. That band was formed in 1968, during the height of the Japanese music history."

In the above example, the word "the band" in the second sentence refers to the band "The Transformers" introduced in the first sentence. When you read about the band in the second sentence, you know that it is referencing to the "The Transformers" band. That may be important for translation.

For translating other sentences like that, a model needs to figure out these sort of dependencies and connections. Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) have been used to deal with this problem because of their properties.

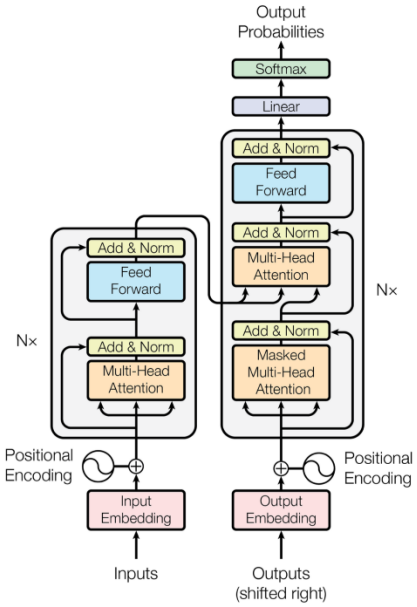


Figure 1: The Transformer - model architecture.

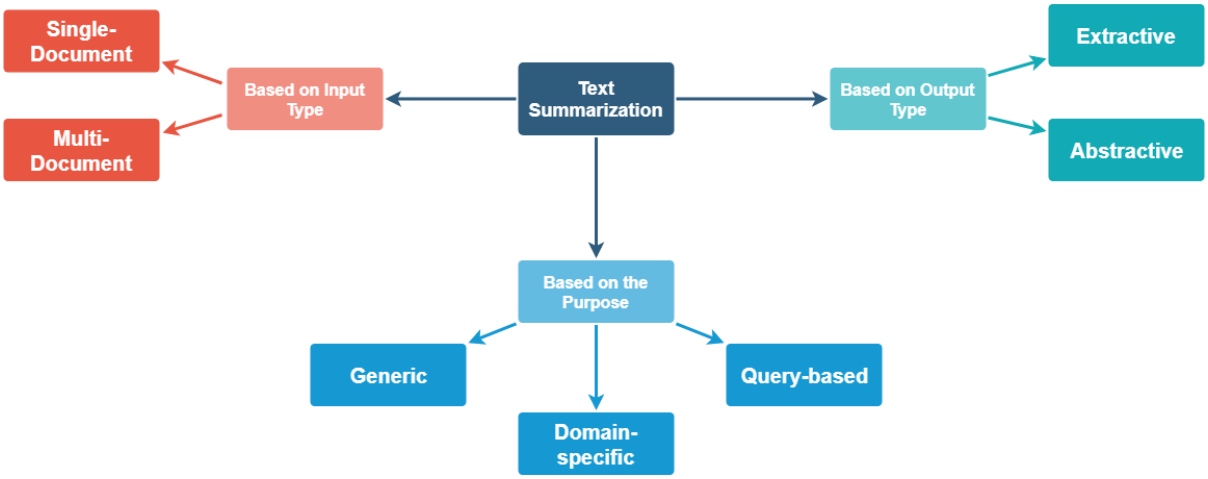
Q10. What is Text summarization?

Answer:

Text summarization: It is the process of shortening a text document, to create a summary of the significant points of the original document.

Types of Text Summarization Methods :

Text summarization methods can be classified into different types.



Example:

Source Text: Peter and Elizabeth took a taxi to attend the night party in the city.

While in the party, Elizabeth collapsed and was rushed to the hospital.

Summary: Elizabeth was hospitalized after attending a party with Peter.
