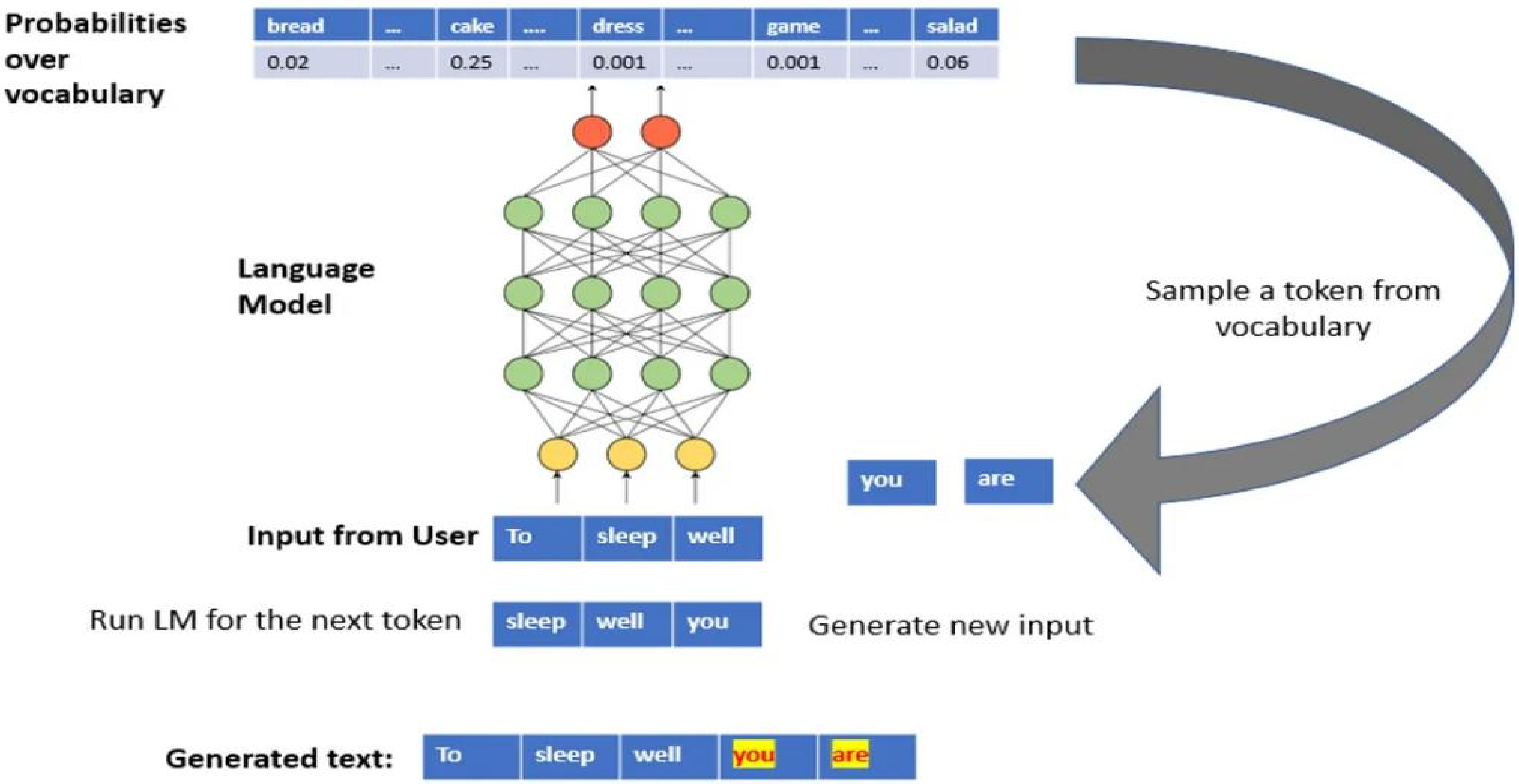
Prompt Engineering (R1UD701T)

Evolution of Language Models

Introduction to Language Models

- A model that predicts the next word in a sequence of words.
- Importance: Used in tasks like translation, summarization, and chatbots.

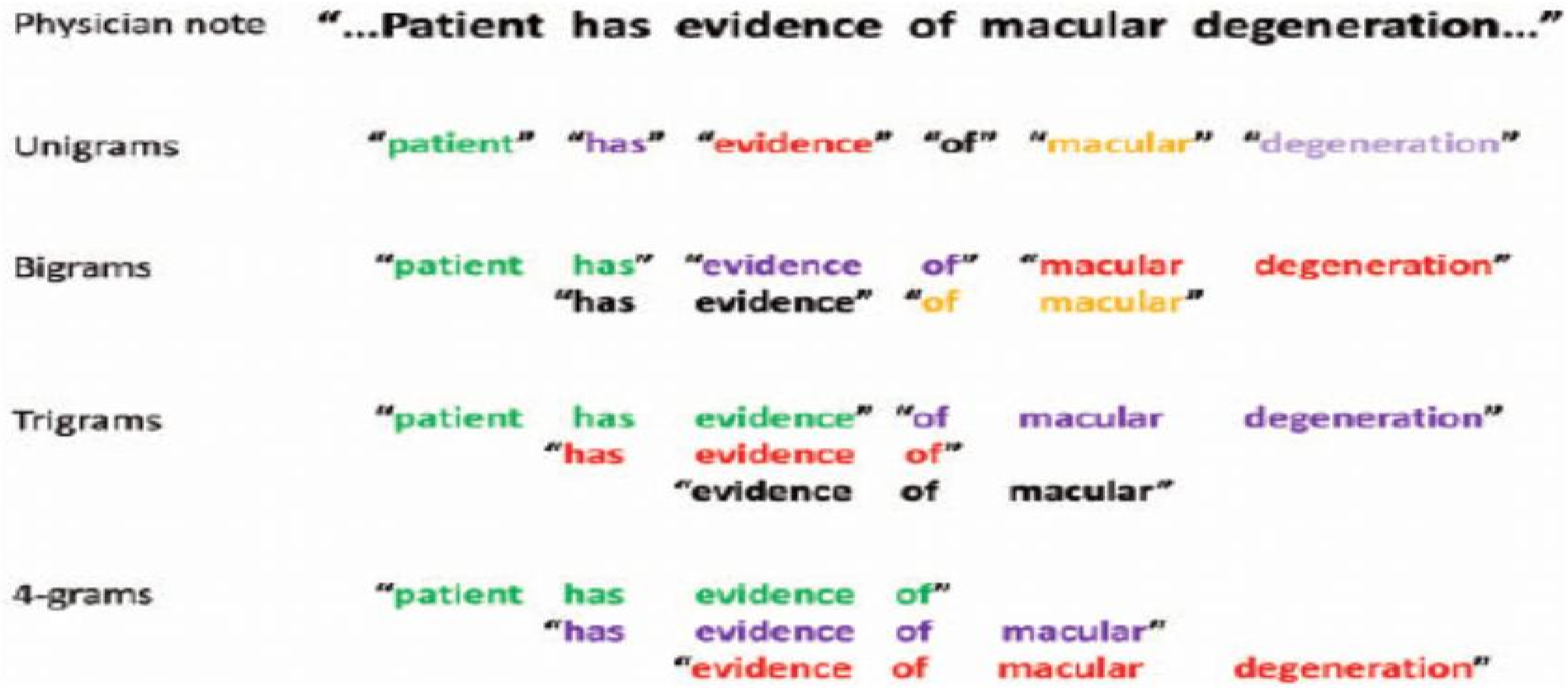


Early Approaches: N-gram Models

N-gram Models:

How they work: Predict the next word based on the previous n−1 words.

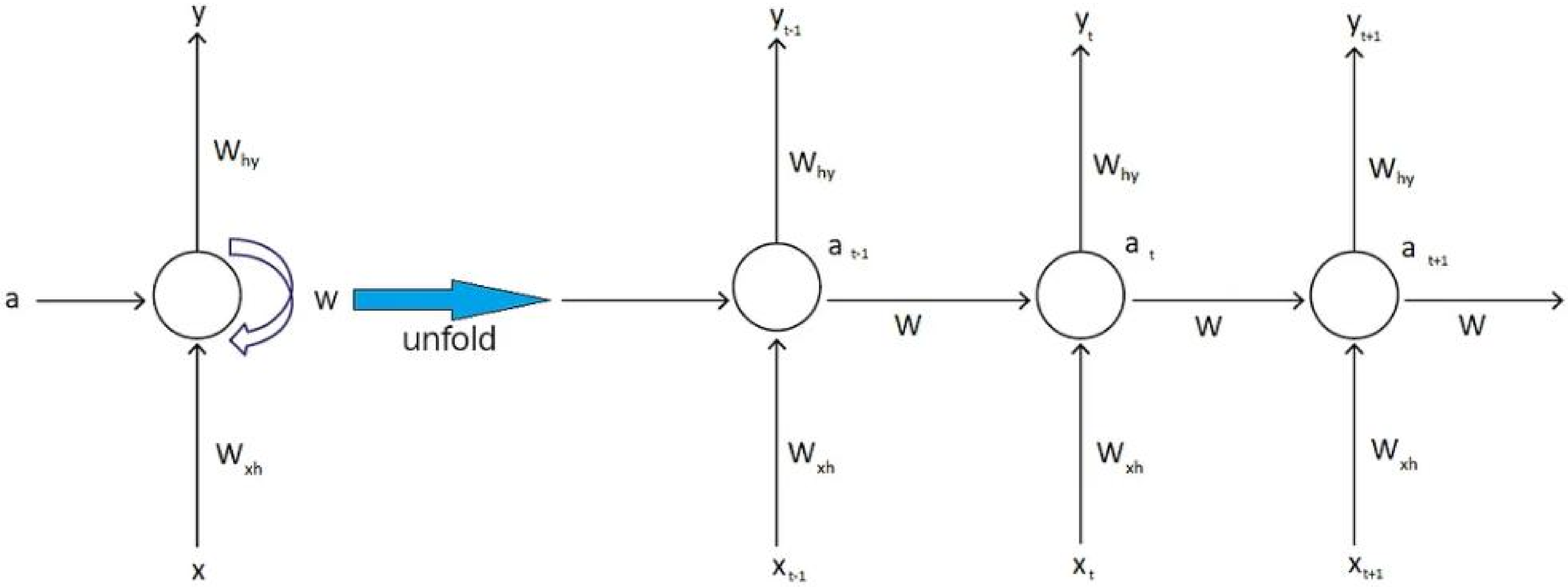
Limitations: Fixed context size, doesn’t handle long dependencies.


Physician note
“...Patient has evidence of macular degeneration...”
Unigrams
“patient” “has” “evidence” “of” “macular” “degeneration”
Bigrams
“patient has” “evidence of” “macular degeneration”
“has evidence” “of macular”
Trigrams
“patient has evidence” “of macular degeneration”
“has evidence of”
“evidence of macular”
4-grams
“patient has evidence of”
“has evidence of macular”
“evidence of macular degeneration”

Neural Network-based Models:

Recurrent Neural Networks (RNNs): Process sequences of data.

Limitations: Struggle with long-term dependencies due to vanishing gradients.



Advancements with LSTMs and GRUs

LSTM (Long Short-Term Memory): Use gates to control information flow.
Advantage: Can capture long-term dependencies.

GRU (Gated Recurrent Unit): Simpler version of LSTMs.

LSTM (Long Short-Term Memory):

Memory Cell: LSTM has a cell state that carries information across the sequence, enabling the network to remember information over long periods.

Gates: LSTM uses three gates:

Input Gate: Controls the extent to which new information flows into the memory cell.
Forget Gate: Controls the extent to which information from the memory cell is erased.
Output Gate: Controls the extent to which information from the memory cell is used to compute the output.
Complexity: The inclusion of multiple gates and a separate cell state makes LSTM more complex compared to GRU.

GRU (Gated Recurrent Unit):

Memory Cell: GRU combines the hidden state and cell state into a single state.
Gates: GRU uses two gates:
Reset Gate: Decides how much of the past information to forget.
Update Gate: Decides how much of the past information needs to be passed along to the future.
Simplicity: GRU is simpler than LSTM because it has fewer gates and lacks a separate cell state.

Attention mechanism

The attention mechanism is a crucial concept in language models, especially in the context of deep learning and natural language processing (NLP). It allows models to focus on specific parts of the input sequence when making predictions, effectively handling the issue of long-range dependencies in sequences. Here's an overview of the attention mechanism:

Key Concepts of Attention Mechanism

Contextual Focus:
Attention mechanisms enable models to assign different weights to different parts of the input sequence. This means the model can focus more on relevant parts of the input while making predictions, rather than treating all parts of the sequence equally.

Alignment:
Attention involves computing a score that measures how well an input element and the current state of the model align. This score is then used to weigh the input elements, creating a context vector that represents a weighted sum of the input elements.

Pre-trained Language Models

Content:

ELMo (Embeddings from Language Models): Creates deep contextualized word embeddings.

OpenAI GPT (Generative Pre-trained Transformer):
GPT-1 to GPT-3: Improvements in text generation capabilities.

BERT (Bidirectional Encoder Representations from Transformers): Uses bidirectional training for better context understanding.

.

ELMo (Embeddings from Language Models)

Architecture:

- ELMo uses a bidirectional Long Short-Term Memory (BiLSTM) network to generate contextualized word embeddings.
- It captures both forward and backward contexts using two separate LSTM networks.

Training Objective:

ELMo is trained with a bidirectional language model objective. It maximizes the likelihood of the current word given the previous words (forward direction) and the likelihood of the current word given the next words (backward direction).

Usage:

ELMo provides word embeddings that are context-sensitive, meaning the representation of a word can change depending on its context within a sentence.

- These embeddings can be used as additional features in downstream tasks like text classification, named entity recognition, etc.

Key Feature:

- ELMo generates deep contextualized word representations, which capture complex characteristics of word usage and how these usages vary across linguistic contexts.

GPT (Generative Pre-trained Transformer)

Architecture:

GPT is based on the Transformer architecture, specifically using a stack of Transformer decoder layers.

It processes text in a unidirectional (left-to-right) manner, meaning it only considers previous tokens when generating or predicting the next token.

Training Objective:

GPT is trained using a language modeling objective, where it maximizes the likelihood of the next word given the previous words in the sequence (causal or autoregressive language modeling).

Usage:

GPT can generate coherent and contextually relevant text based on a given prompt. It can also be fine-tuned for various NLP tasks.

It has been used for tasks such as text generation, translation, summarization, and more.

Key Feature:

GPT is particularly known for its ability to generate high-quality, fluent text. Its unidirectional nature makes it well-suited for generative tasks.

BERT (Bidirectional Encoder Representations from Transformers)

Architecture:

BERT is based on the Transformer architecture, specifically using a stack of Transformer encoder layers.

It processes text bidirectionally, meaning it considers both the left and right context simultaneously.

Training Objective:

BERT uses two main training objectives:

Masked Language Modeling (MLM): Randomly masks some of the tokens in the input and trains the model to predict these masked tokens.

Next Sentence Prediction (NSP): Trains the model to understand sentence relationships by predicting if one sentence follows another.

Usage:

BERT is fine-tuned for specific NLP tasks such as question answering, sentiment analysis, and text classification.

Fine-tuning involves training the pre-trained BERT model on a task-specific dataset with additional layers on top.

Key Feature:

BERT's bidirectional context allows it to achieve state-of-the-art performance on various NLP benchmarks. It effectively captures the full context of a word by looking at both its preceding and succeeding words.

Recent and Advanced Models

T5 (Text-To-Text Transfer Transformer): Unified framework for various NLP tasks.

XLNet: Combines advantages of autoregressive and autoencoding models.

RoBERTa: Enhanced version of BERT.

Conclusion and Future Directions

Future Trends:

Larger models, better efficiency, multimodal models.

Ethical considerations in AI.

1990s - Early 2000s: Statistical Models

- N-gram Models: Use statistical methods to predict the next word based on the previous N-1 words.
- Limitations: Struggle with long-range dependencies and require large amounts of data.

2000s: Introduction of Neural Networks

- Feedforward Neural Networks: Early attempts to use neural networks for language modeling but still limited in capturing long-term dependencies.
- Recurrent Neural Networks (RNNs): Capture sequential data but suffer from vanishing gradient problems.

2014: Long Short-Term Memory (LSTM)

- LSTM (Hochreiter and Schmidhuber): Solves vanishing gradient problems by introducing memory cells and gates.
- Applications: Improved handling of long-range dependencies in sequential data.

2015: Gated Recurrent Unit (GRU)

- GRU (Cho et al.): A simpler alternative to LSTMs with fewer gates and comparable performance.
- Applications: Faster training times while maintaining effective learning capabilities.

2017: Attention Mechanism and Transformer Models

- Attention is All You Need (Vaswani et al.): Introduces the Transformer model with self-attention mechanism, allowing parallelization and better handling of long-range dependencies.
- Impact: Paves the way for state-of-the-art language models.

2018: Contextualized Word Embeddings

ELMo (Peters et al.): Uses BiLSTM to create deep contextualized word representations that change based on context.

Impact: Significantly improves performance on various NLP tasks by providing rich contextual embeddings.

2018: Generative Pre-trained Transformer (GPT)

GPT (Radford et al.): Uses a unidirectional Transformer decoder and is pre-trained on a large corpus with a language modeling objective.

Impact: Demonstrates the power of transfer learning in NLP, excelling at text generation tasks.

2019: Bidirectional Encoder Representations from Transformers (BERT)

BERT (Devlin et al.): Uses a bidirectional Transformer encoder, trained with masked language modeling and next sentence prediction.

Impact: Achieves state-of-the-art results on a wide range of NLP benchmarks, emphasizing the importance of bidirectional context.

2019 - Present: Advancements and Variations

GPT-2 (2019, Radford et al.): An improved version of GPT with significantly more parameters and capabilities.

BERT Variants (e.g., RoBERTa, ALBERT): Enhanced versions of BERT with optimizations in training procedures and architecture.

GPT-3 (2020, Brown et al.): A much larger version of GPT with 175 billion parameters, demonstrating impressive zero-shot and few-shot learning abilities.

T5 (Text-To-Text Transfer Transformer, 2019, Raffel et al.): Treats all NLP tasks as text-to-text transformations, unifying the approach to various NLP problems.

BERT-based Models (e.g., DistilBERT, ELECTRA): Focus on efficiency and training strategies to reduce computational costs while maintaining performance.

Thank You