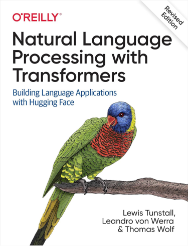
Introduction to Large Language Models

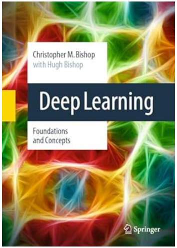
Dr. Aijun Zhang

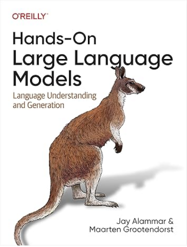
October 2024

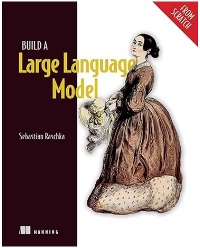
Recommended Texts

- Tunstall, L., von Werra, L. and Wolf, T. (2022).
- Bishop, C. and Bishop, H. (2023)
- Alammar, J. and Grootendorst, M. (2024)
- Raschka, S. (2024)





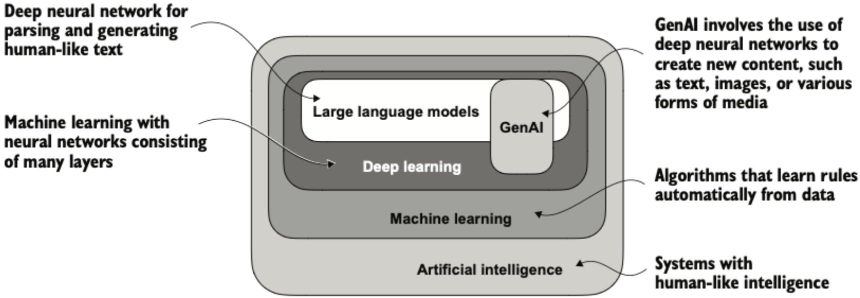




Outline

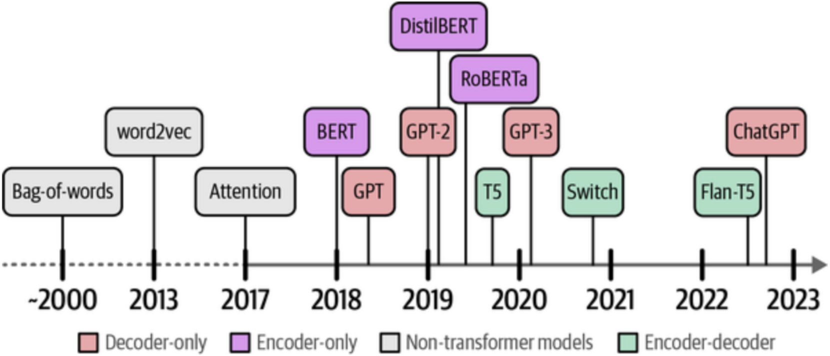
1. LLMs - A Quick Overview
2. Attention is All You Need (2017)
   - Transformer Architecture
   - Attention Mechanism
   - Specialized Transformer LLMs
3. Interpreting Contextual Embeddings
4. Text Classification and Fine-Tuning

Landscape of Artificial Intelligence



Source: Raschka (2024)

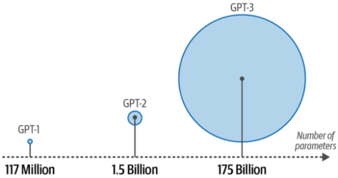
Evolution of Language Models

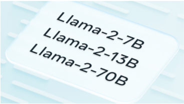


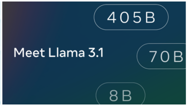
Source: Alammar and Grootendorst (2024)

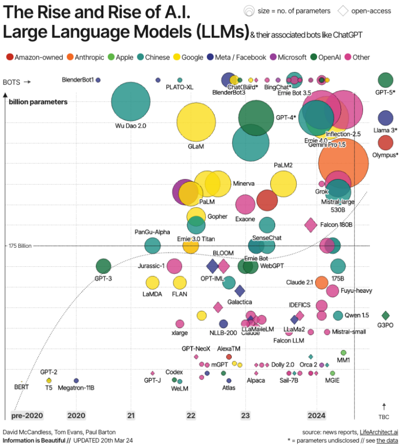
Growing Scales of Language Models

Scaling laws: increasing parameters, data quality, and compute resources generally improves LLM performance.









Click into VizSweet

Key Tasks of Large Language Models

- **Text Classification**: Sentiment analysis, Spam detection, Named Entity Recognition (NER), Natural Language Inference (NLI)
- **Text Generation**: Creative writing, Chatbot, Translation, Coding
- **Summarization**: News article summaries, Research paper abstracts, Document condensation
- **Question Answering**: Customer support queries, Educational FAQs
- **Knowledge Integration**: Retrieval-Augmented Generation (RAG) for up-to-date responses, evidence-based information generation
- **Reasoning**: Logical deduction, Mathematics problem solving, Chain-of-Thought analysis

Outline

1. LLMs - A Quick Overview
2. Attention is All You Need (2017)
   - Transformer Architecture
   - Attention Mechanism
   - Specialized Transformer LLMs
3. Interpreting Contextual Embeddings
4. Text Classification and Fine-Tuning

Attention is All You Need (2017)

- By Google Brain: Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L. and Gomez, A. N., Kaiser, L. and Polosukhin, I. (NIPS 2017)
- It revolutionized neural networks by introducing the Transformer, a highly flexible architecture enabling LLMs like BERT, GPT, and T5.
- The core innovation of self-attention allows each token to capture global context efficiently, eliminating the need for recurrence.
- It sparked groundbreaking models across NLP and other domains, with applications extending to vision, audio, and multimodal tasks.

> **Andrej Karpathy**
> @karpathy
>
> The Transformer is a magnificent neural network architecture because it is a general-purpose differentiable computer. It is simultaneously:
> 1) expressive (in the forward pass)
> 2) optimizable (via backpropagation+gradient descent)
> 3) efficient (high parallelism compute graph)
>
> 2:54 PM · Oct 19, 2022

Transformer Architecture

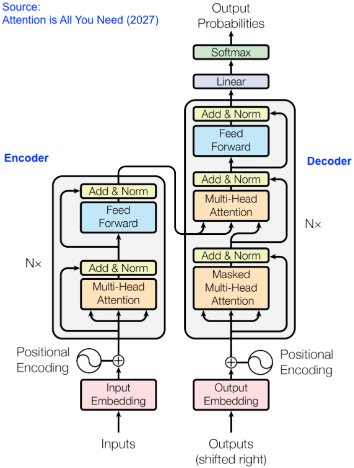
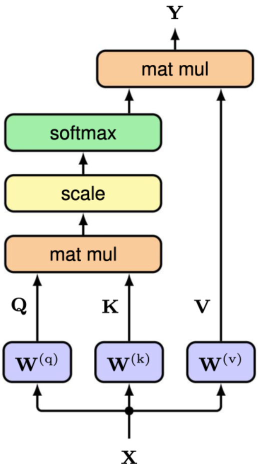


Figure 1: The Transformer - model architecture.

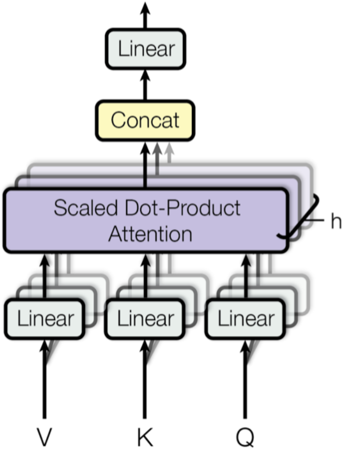
- **Encoder** processes input
- **Decoder** generates output, predicting the next token auto-regressively
- Feed forward network (deep learning)
- Multi-head self-attention
- Masked attention for decoder
- Positional encoding
- Check PyTorch function:

  Docs>torch.nn>Transformer

Attention Mechanism

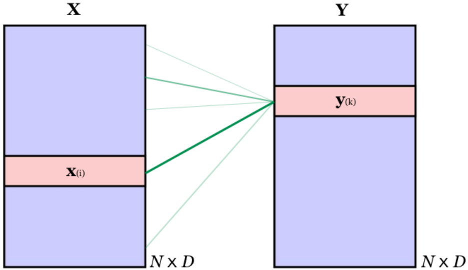


Single-head Attention



Multi-head Attention

Scaled Dot-Product Self-Attention



$$\mathbf{y}_{(k)} \leftarrow \sum_{i=1}^{N} \alpha_{ki} \mathbf{x}_{(i)}$$

$$\alpha_{ki} = \frac{\exp(\mathbf{x}_{(k)} \mathbf{x}_{(i)}^T)}{\sum_{j=1}^{N} \exp(\mathbf{x}_{(k)} \mathbf{x}_{(j)}^T)}$$

Expressed in matrix form:

$$\mathbf{Y} = \text{Softmax}[\mathbf{X}\mathbf{X}^T]\mathbf{X}$$

(Q,K,V)-parameterization: $\mathbf{W}^{(q)}, \mathbf{W}^{(k)}, \mathbf{W}^{(v)} \in \mathbb{R}^{D \times D}$

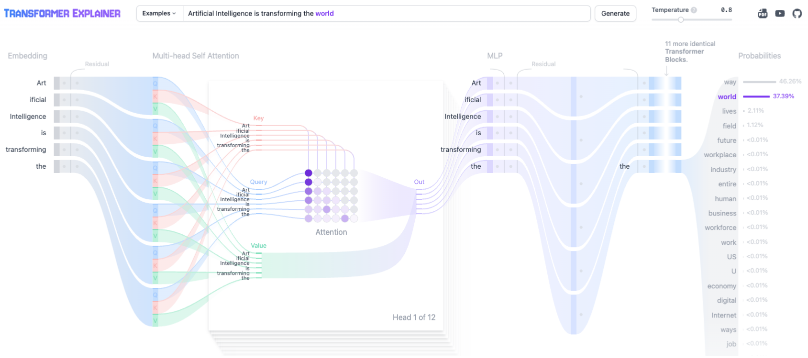
$$\mathbf{Y} = \text{Softmax}\left[\mathbf{X}\mathbf{W}^{(q)}(\mathbf{X}\mathbf{W}^{(k)})^T\right]\mathbf{X}\mathbf{W}^{(v)} \equiv \text{Softmax}[\mathbf{Q}\mathbf{K}^T]\mathbf{V}$$

Scaled self-attention: $\mathbf{Y} = \text{Softmax}\left[\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D}}\right]\mathbf{V} \equiv \text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$

Specialized Transformer LLMs

- **Encoder-Only Models: BERT** (Bidirenctional Encoder Representations from Transformers), DistilBERT, RoBERTa, DeBERTa, etc. Ideal for *understanding tasks* such as text classification, sentiment analysis, and named entity recognition, with bidirectional attention capturing full context.
- **Decoder-Only Models: GPT** (Generative Pretrained Transformer), GPT-2, GPT-3, and beyond. Optimized for *generation tasks*, with unidirectional attention. Live example: ChatGPT (GPT-3.5+)
- **Encoder-Decoder Models: T5** (Text-to-Text Transfer Transformer), BART. Balance input understanding with output generation, suitable for tasks like translation, summarization, and paraphrasing.

Highlight: Encoder-only models (BERT) are for understanding tasks, while decoder-only models (GPT) are for generative tasks.

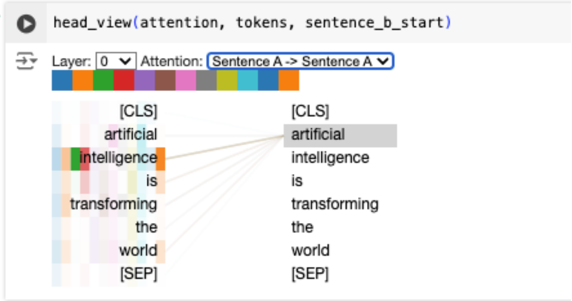


Transformer Explainer: Interactive Learning of GPT-2 by Cho et al.(2024)

Live Demo: BertViz

BertViz: Visualize Attention in NLP Models (BERT, GPT2, BART, etc.)

https://github.com/jessevig/bertviz



Try it on Google Colab: BertViz Interactive Tutorial

Outline

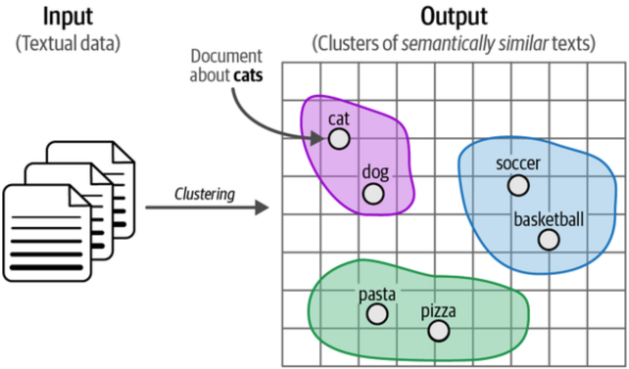
Static and Contextual Embeddings

- **Embeddings** provide a way to represent textual data as dense, continuous vectors in a high-dimensional space, to capture the semantic meaning of words, phrases, and documents.
- **Traditional Static Embeddings:** Word2Vec, GloVe, etc. Easy to compute, able to capture basic semantic relationships. However, each word has a single, fixed embedding regardless of context.
- **Contextual Embeddings:** BERT, GTP, etc. Generate dynamic, context-dependent embeddings for each token. BERT is most popular as it captures both the left and right context of a word in a sentense.

Interpreting Contextual Embeddings

- **Interpretability matters** as it is crucial to understand how LLMs work and make decisions, fostering transparency, trust and reliability.
- **Challenges in Interpreting LLMs**:
  - Complexity of contextual embeddings, high-dimensionality
  - Black-box nature, millions/billions of parameters, highly nonlinear
  - Semantic ambiguity, polysemy (words with multiple meanings) in diverse contexts
- **Here's a structured process** to interpret contextual embeddings:
  1. Dimensionality reduction for effective clustering
  2. Extract topics for each cluster
  3. Further dimensionality reduction for 2D or 3D visualization

Interpreting Contextual Embeddings



Source: Alammar and Grootendorst (2024)

Dimensionality Reduction Techniques

Technique	Type	Strength	Weakness	Best Use
PCA	Linear	Captures maximum variance	Assumes linearity, less flexible	Medium data with linear relationships
t-SNE	Nonlinear	Preserves local structure	Computationally expensive	Small data, ideal for 2D/3D visualization
UMAP	Nonlinear	Balances local & global structure, scalable	Requires tuning	Clustering in large, complex datasets
Random Projection	Linear (Random)	Fast, scalable, preserves distances	Lower interpretability	High-dim data with simple structure
AE (Auto-Encoders)	Nonlinear	Learns nonlinear relationships, customizable	Requires tuning & significant data	Complex datasets with non-linear patterns
VAE (Variational AE)	Nonlinear (Probabilistic)	Captures data variability, generates new data	Complex training, requires tuning	When data variability and generation are important
MDS	Nonlinear	Flexible metrics, preserves distances	Computationally intensive	Semantic similarity in embeddings

Clustering Techniques

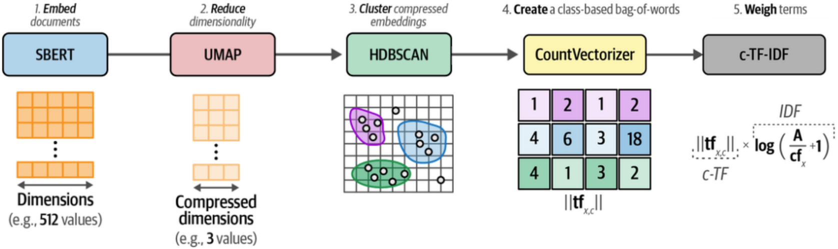
Technique	How It Works	Advantages	Limitations	Best Use
k-Means Clustering	Minimizing distances to centroids	Simple, efficient, scalable	Requires k in advance; assumes spherical clusters	When clusters are approximately spherical and equally sized
Agglomerative /Hierarchical Clustering	Merges closest clusters iteratively, forming a hierarchy	Captures hierarchical structure, no need for k	Computationally intensive, sensitive to noise	Data with inherent hierarchical structure
DBSCAN	Groups densely packed points; labels sparse points as outliers	Handles non-convex shapes, detects outliers	Sensitive to parameters, no good for varying densities	Arbitrary shapes, outlier detection, unknown clusters
Spectral Clustering	Uses similarity matrix and eigenvalues for clustering	Good for non-convex clusters, adaptable similarity measures	Computationally expensive, requires k	Small data with complex relationships

Topic Extraction Techniques

Technique	Description	Use in Clusters
TF-IDF (Term Frequency – Inverse Document Frequency)	Identifies important terms by comparing term frequency in a document to its frequency in the corpus. High scores indicate terms important to the document but uncommon in the corpus.	Applied to each cluster to find keywords that best represent its content.
KeyBERT (Keyword Extraction with BERT)	A transformer-based method using BERT embeddings to extract semantically relevant keywords.	Identifies representative keywords or phrases, providing rich, contextually accurate topic descriptions for each cluster.
LDA (Latent Dirichlet Allocation)	A topic modeling technique that assumes documents are mixtures of topics, each with a unique word distribution.	Extracts coherent topics from clusters, revealing specific themes within broader topics.

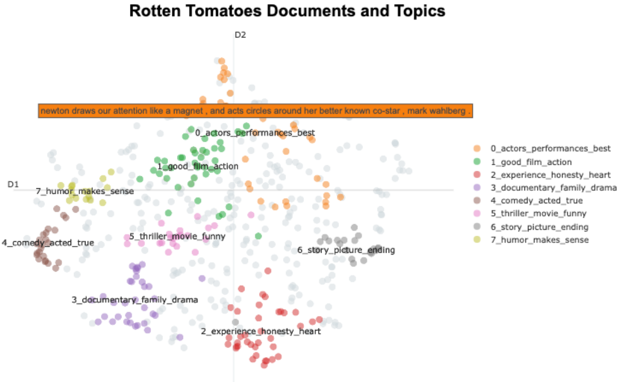
BERTopic Pipeline for Interpretable Topic Modeling

https://github.com/MaartenGr/BERTopic



Source: Alammar and Grootendorst (2024)

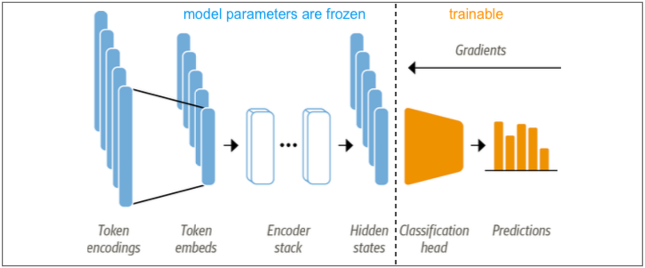
Live Demo: BERTopic



Try it on Google Colab: BERTopic Demo with Rotten Tomatoes Dataset

Outline

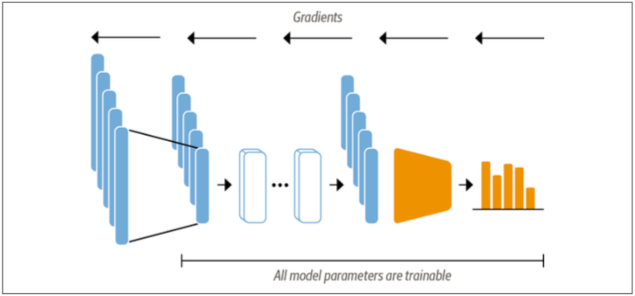
Text Classification with Pretrained Transformers



Source: Tunstall et al. (2022)

1. Select a pretrained transformer model, e.g. SBERT or DistilBERT;
2. Prepare the labeled text data, split into training and validation sets;
3. Add a classifier layer with sigmoid/softmax activation;
4. Freeze the transformer model parameters, train only the classifier;
5. Evaluate performance and perform outcome analysis.

Fine-Tuning Transformers for Classification



Source: Tunstall et al. (2022)

- Pros: Fine-tuning the entire model adapts fully to the task, yielding higher accuracy and flexibility.
- Cons: Increased computational demands, potential risk of overfitting.

Live Demo: DistilBERT

- DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base.
- Check the hugging face transformers page: DistilBERT

Table: DistilBERT Classification on Rotten Tomatoes Data

	DistilBERT-raw	**DistilBERT-finetuned**
train-AUC	0.918097	0.980089
test-AUC	0.882378	0.911010

Try it on Google Colab: Text Classification using DistiBERT

Thank you!

https://www.linkedin.com/in/ajzhang/