Best educational resources for deep learning in NLP

This curated guide pairs **one high-quality article** and **one YouTube video** for each of 20 core topics in Deep Learning for NLP. Every resource was selected for clarity, technical accuracy, and accessibility at the graduate level, drawing from educators like Jay Alammar, Andrej Karpathy, 3Blue1Brown, StatQuest, Lilian Weng, and Sebastian Raschka.

---

1. Transformer architecture

📄 **Article — "The Illustrated Transformer"**

- **Author:** Jay Alammar
- **URL:** https://jalammar.github.io/illustrated-transformer/
- **Why:** The gold-standard visual explainer of self-attention, Q/K/V, multi-head attention, and encoder-decoder structure. Referenced in courses at Stanford, Harvard, MIT, and CMU; translated into 12+ languages. (LinkedIn) (Jay Alammar)

📹 **Video — "Attention in transformers, step-by-step | Deep Learning Chapter 6"**

- **Channel:** 3Blue1Brown (Grant Sanderson)
- **URL:** https://www.youtube.com/watch?v=eMlx5fFNoYc
- **Why:** Beautifully animated, mathematically rigorous walkthrough of Q/K/V matrices, softmax, multi-head attention, cross-attention, and parameter counting. (Google Groups) Widely regarded as the single best visual explanation of attention. (Simon Willison)

---

2. Positional encodings

📄 **Article — "Transformer Architecture: The Positional Encoding"**

- **Author:** Amirhossein Kazemnejad
- **URL:** https://kazemnejad.com/blog/transformer_architecture_positional_encoding/
- **Why:** The most widely cited standalone deep-dive on positional encoding. Derives the linear transformation property of sinusoidal encodings, explains design criteria (uniqueness, bounded values, generalization to longer sequences), and provides clear proofs and visualizations. (Kazemnejad)

📹 **Video — "Positional Encoding in Transformers"**

- **Channel:** CodeEmporium
- **URL:** https://www.youtube.com/watch?v=o29P0Kpobz0
- **Why:** Dedicated walkthrough of why order information is needed, how sinusoidal functions encode position, and the intuition behind the sin/cos frequency scheme. (Medium) Accessible and code-oriented.

---

3. Masked self-attention and cross-attention

📄 **Article — "Understanding and Coding Self-Attention, Multi-Head Attention, Cross-Attention, and Causal Self-Attention in LLMs"**

- **Author:** Sebastian Raschka (Ahead of AI / Substack)
- **URL:** https://magazine.sebastianraschka.com/p/understanding-and-coding-self-attention
- **Why:** Comprehensive from-scratch PyTorch walkthrough by a leading ML educator. Covers basic self-attention, multi-head attention, cross-attention (Q from decoder, K/V from encoder), and causal/masked self-attention with triangular masking. (Sebastian Raschka)

📹 **Video — "Attention in transformers, step-by-step | Deep Learning Chapter 6" (timestamps 11:08 and 18:21)**

- **Channel:** 3Blue1Brown (Grant Sanderson)
- **URL:** https://www.youtube.com/watch?v=eMlx5fFNoYc
- **Why:** At **11:08**, explains causal masking (setting future positions to $-\infty$ before softmax). At **18:21**, explains cross-attention (Keys/Values from encoder, Queries from decoder). (Medium) The clearest animated treatment of both concepts.

---

4. Transformer parameter counting

📄 **Article — "Transformer Param Counting"**

- **Author:** Kipply (kipp.ly)
- **URL:** https://kipp.ly/transformer-param-count/
- **Why:** Derives the clean formula **12 × n_layers × d_model²** for dominant parameter counts, breaks down MLP (4× ratio) and attention head relationships, and applies it to real models (GPT-2, GPT-3, Gopher). (kipply's blog) Concise and practical.
- **Alternative:** Michael Wornow's "Counting Model Parameters" at https://michaelwornow.net/2024/01/18/counting-params-in-transformer provides a layer-by-layer GPT-2

breakdown with HuggingFace code verification.

📽️ Video — "Attention in transformers, step-by-step | Deep Learning Chapter 6" (timestamp 15:44)

- **Channel:** 3Blue1Brown (Grant Sanderson)
- **URL:** https://www.youtube.com/watch?v=eMlx5fFNoYc
- **Why:** At **15:44**, walks through parameter counting for a single attention head (~6.3M params: key, query, value, and output matrices) and scales up to GPT-3's 96 heads. (Medium) The clearest animated explanation of transformer parameter counting on YouTube.

---

5. BERT

📄 Article — "The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)"

- **Author:** Jay Alammar
- **URL:** https://jalammar.github.io/illustrated-bert/
- **Why:** The definitive visual explainer of BERT's encoder-only architecture, Masked Language Modeling (MLM), Next Sentence Prediction (NSP), pre-training vs. fine-tuning, and comparisons with ELMo and GPT.

📽️ Video — "Encoder-Only Transformers (like BERT), Clearly Explained!!!"

- **Channel:** StatQuest with Josh Starmer
- **URL:** https://www.youtube.com/watch?v=MN__lSncZBs
- **Why:** StatQuest's step-by-step visual walkthrough of BERT's encoder-only architecture, word embeddings, positional encoding, attention, and practical applications. Classic "Clearly Explained" format ideal for graduate students.

---

6. T5 (Text-to-Text Transfer Transformer)

📄 Article — "Exploring Transfer Learning with T5: the Text-To-Text Transfer Transformer"

- **Author:** Google Research (Adam Roberts, Colin Raffel, et al.)
- **URL:** https://research.google/blog/exploring-transfer-learning-with-t5-the-text-to-text-transfer-transformer/

- **Why:** The official Google Research blog post introducing T5. Clearly explains the unified text-to-text framework with task prefixes, the C4 dataset, the denoising objective, and SOTA benchmark results with interactive demos.

🎥 Video — "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer" (MLST #5)

- **Channel:** Machine Learning Street Talk (Yannic Kilcher, Tim Scarfe, Connor Shorten)
- **URL:** Search YouTube for "Machine Learning Street Talk T5 Exploring the Limits"
- **Why:** In-depth expert discussion of the T5 paper covering the text-to-text framework, architecture choices, pre-training objectives, and compute scaling decisions. Features Yannic Kilcher's detailed analysis. Towards Data Science

---

7. GPT and autoregressive language modeling

📄 **Article — "The Illustrated GPT-2 (Visualizing Transformer Language Models)"**

- **Author:** Jay Alammar
- **URL:** https://jalammar.github.io/illustrated-gpt2/
- **Why:** Superb visual deep-dive into the GPT-2 decoder-only architecture, covering autoregressive next-token prediction, masked self-attention, the contrast with BERT's bidirectional approach, and KV caching. Jay Alammar github

🎥 Video — "Let's build GPT: from scratch, in code, spelled out."

- **Channel:** Andrej Karpathy
- **URL:** https://www.youtube.com/watch?v=kCc8FmEb1nY
- **Why:** A legendary **1h56m** hands-on tutorial building a GPT entirely from scratch in PyTorch. Covers tokenization, self-attention, multi-head attention, residual connections, layer norm, and autoregressive generation. Universally considered one of the best educational videos on transformers ever made.

---

8. Fine-tuning vs prompting vs in-context learning

📄 **Article — "Finetuning Large Language Models"**

- **Author:** Sebastian Raschka (Ahead of AI / Substack)
- **URL:** https://magazine.sebastianraschka.com/p/finetuning-large-language-models

- **Why:** Comprehensive overview covering in-context learning (zero-shot, few-shot), hard/soft prompt tuning, RAG, and all major fine-tuning approaches (feature-based, full fine-tuning, LoRA, adapters). Includes clear diagrams comparing when to use each paradigm.

🎥 Video — "Deep Dive into LLMs like ChatGPT"

- **Channel:** Andrej Karpathy
- **URL:** https://www.youtube.com/watch?v=7xTGNNLPyMI
- **Why:** A **3.5-hour** masterclass covering the full LLM training stack: pre-training, supervised fine-tuning, in-context learning (few-shot prompting), and RLHF. (Medium +2) Karpathy explains the distinction between base models and fine-tuned assistants (Neptune.ai) with practical examples.

---

9. Chain-of-Thought (CoT) prompting

📄 Article — "Prompt Engineering" (Chain-of-Thought sections)

- **Author:** Lilian Weng (Lil'Log)
- **URL:** https://lilianweng.github.io/posts/2023-03-15-prompt-engineering/
- **Why:** Lilian Weng's authoritative blog post (Lil'Log) covers few-shot CoT, zero-shot CoT ("Let's think step by step"), self-consistency sampling, and Auto-CoT — all with references to the original Wei et al. (2022) paper.

🎥 Video — "Deep Dive into LLMs like ChatGPT" (reasoning / prompting sections)

- **Channel:** Andrej Karpathy
- **URL:** https://www.youtube.com/watch?v=7xTGNNLPyMI
- **Why:** Karpathy covers how reasoning emerges in LLMs, the role of step-by-step reasoning in prompts, and how RLHF and training for reasoning (including DeepSeek-R1) improve chain-of-thought capabilities. (X) See also the DAIR.AI Prompt Engineering Guide at https://www.promptingguide.ai/techniques/cot for a supplementary written+video resource.

---

10. Seq2Seq models and attention mechanism

📄 Article — "Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention)"

- **Author:** Jay Alammar

- **URL:** https://jalammar.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/
- **Why:** The gold-standard visual explainer of seq2seq with attention. Uses animations to show the encoder-decoder architecture, the context vector bottleneck, and how Bahdanau attention solves it by letting the decoder attend to all encoder hidden states. (Jay Alammar)

🎥 Video — "Attention for Neural Networks, Clearly Explained!!!"

- **Channel:** StatQuest with Josh Starmer
- **URL:** https://www.youtube.com/watch?v=PSs6nxngL6k
- **Why:** StatQuest's signature clear, step-by-step visual explanation of how attention mechanisms solve the information bottleneck in seq2seq models. Builds from basic encoder-decoder to attention-enhanced architectures.

---

11. Decoding strategies

📄 Article — "How to generate text: using different decoding methods for language generation with Transformers"

- **Author:** Patrick von Platen (Hugging Face)
- **URL:** https://huggingface.co/blog/how-to-generate
- **Why:** The definitive practical guide covering greedy search, beam search, top-k sampling, and nucleus (top-p) sampling (Medium) with runnable HuggingFace code examples. (Medium) Updated July 2023. (Hugging Face)
- **Companion:** Maxime Labonne's "Decoding Strategies in Large Language Models" at https://huggingface.co/blog/mlabonne/decoding-strategies provides excellent interactive visualizations and decision trees.

🎥 Video — "Deep Dive into LLMs like ChatGPT" (generation/sampling section)

- **Channel:** Andrej Karpathy
- **URL:** https://www.youtube.com/watch?v=7xTGNNLPyMI
- **Why:** Karpathy explains temperature, top-k, and sampling during the pre-training and inference sections. For a more dedicated written+visual walkthrough, the AssemblyAI blog "Decoding Strategies: How LLMs Choose The Next Word" at https://www.assemblyai.com/blog/decoding-strategies-how-llms-choose-the-next-word is also excellent.

12. Tokenization

📄 Article — "Byte-Pair Encoding tokenization" (Hugging Face NLP Course, Chapter 6)

- **Author:** Hugging Face Team
- **URL:** https://huggingface.co/learn/llm-course/en/chapter6/5
- **Why:** Hands-on, interactive course chapter walking through BPE step-by-step with code. Part of the broader Chapter 6 covering WordPiece and Unigram tokenization as well. (Hugging Face) The companion summary at https://huggingface.co/docs/transformers/en/tokenizer_summary covers BPE, WordPiece, SentencePiece, and Unigram. (Hugging Face)

🎥 Video — "Let's build the GPT Tokenizer"

- **Channel:** Andrej Karpathy
- **URL:** https://www.youtube.com/watch?v=zduSFxRajkE
- **Why:** A **2h13m** deep dive building a BPE tokenizer from scratch. (fast.ai) (X) Covers Unicode/UTF-8, the BPE algorithm, regex-based pre-tokenization, tiktoken, (LinkedIn) and explains why many LLM quirks (bad arithmetic, poor non-English performance) trace back to tokenization. (Francescopochetti) (Simon Willison) Released February 2024.

---

13. Word embeddings

📄 Article — "The Illustrated Word2vec"

- **Author:** Jay Alammar
- **URL:** https://jalammar.github.io/illustrated-word2vec/
- **Why:** Beautiful visual walkthrough of word embeddings from first principles — covers CBOW, Skip-gram, negative sampling, and even word2vec applications to recommendation systems (Jay Alammar) (Airbnb, Spotify). 347 points on Hacker News. (Jay Alammar) (Jay Alammar)

🎥 Video — "Word Embedding and Word2Vec, Clearly Explained!!!"

- **Channel:** StatQuest with Josh Starmer
- **URL:** https://youtu.be/viZrOnJclY0
- **Why:** StatQuest's clear visual explanation of word2vec including CBOW and Skip-gram architectures and negative sampling. (Glasp) Accompanied by a hands-on PyTorch coding tutorial on the StatQuest

GitHub. GitHub

---

14. Hallucinations in LLMs

📄 **Article — "Extrinsic Hallucinations in LLMs"**

- **Author:** Lilian Weng (Lil'Log / OpenAI)
- **URL:** https://lilianweng.github.io/posts/2024-07-07-hallucination/
- **Why:** Definitive technical blog post distinguishing in-context from extrinsic hallucinations. Lil'Log Covers causes (training data issues, SFT introducing new knowledge, RLHF worsening calibration), detection methods (self-evaluation, semantic entropy), and mitigation strategies. Published July 2024.

🎥 **Video — "Deep Dive into LLMs like ChatGPT" (hallucination section ~1:30:00)**

- **Channel:** Andrej Karpathy
- **URL:** https://www.youtube.com/watch?v=7xTGNNLPyMI
- **Why:** Karpathy provides an intuitive explanation of why LLMs hallucinate: SFT data always contains confident answers (never "I don't know"), so models learn to sound confident even when uncertain. Substack Covers Meta's factuality probing, tool-use mitigation, and knowledge probing. Medium Anfalmushtaq

---

15. RLHF / InstructGPT

📄 **Article — "Illustrating Reinforcement Learning from Human Feedback (RLHF)"**

- **Author:** Nathan Lambert, Louis Castricato, Leandro von Werra, Alex Havrilla (Hugging Face Blog)
- **URL:** https://huggingface.co/blog/rlhf
- **Why:** The canonical RLHF explainer. Breaks down the 3-step process: (1) pretrain a language model, (2) train a reward model from human preference rankings, (3) fine-tune the LM with PPO. Hugging Face Covers InstructGPT, Anthropic's work, and links to open-source implementations (TRL, TRLX). Hugging Face
- **Honorable mention:** Chip Huyen's RLHF explainer at https://huyenchip.com/2023/05/02/rlhf.html provides excellent intuition for *why* RLHF works.

🎥 **Video — "Deep Dive into LLMs like ChatGPT" (RLHF section ~2:07:28)**

- **Channel:** Andrej Karpathy

- **URL:** https://www.youtube.com/watch?v=7xTGNNLPyMI
- **Why:** Karpathy explains the full SFT → reward model → RL pipeline, the AlphaGo "Move 37" analogy for RL discovery, the discriminator-generator gap (ranking is easier than generating), and reward hacking risks. (CodingScape) The most accessible single-video explanation from a top-tier educator.

---

16. Scaling laws for language models

📄 **Article — "Chinchilla's Wild Implications"**

- **Author:** nostalgebraist (LessWrong)
- **URL:** https://www.lesswrong.com/posts/6Fpvch8RR29qLEWNH/chinchilla-s-wild-implications
- **Why:** The most thorough and widely-read analysis of the Chinchilla scaling laws. Derives the loss equation $L(N,D) = A/N^{\alpha} + B/D^{\beta} + E$, explains why prior models (GPT-3, Gopher) were vastly undertrained, and discusses implications for data scaling. (LessWrong) (Epoch AI) Reviewed and referenced by the original Chinchilla team.
- **Complementary:** "Chinchilla data-optimal scaling laws: In plain English" at https://lifearchitect.ai/chinchilla/ by Alan D. Thompson offers an excellent visual summary. (LifeArchitect.ai)

🎥 **Video — "Deep Dive into LLMs like ChatGPT" (pre-training section)**

- **Channel:** Andrej Karpathy
- **URL:** https://www.youtube.com/watch?v=7xTGNNLPyMI
- **Why:** Karpathy contextualizes scaling laws within the practical LLM development pipeline, discussing training token counts, model sizes, and the Chinchilla-optimal ratio. For a dedicated paper breakdown, search for Yannic Kilcher's review of the Chinchilla paper on his YouTube channel. (Towards Data Science)

---

17. PyTorch basics for NLP

📄 **Article — "The Annotated Transformer"**

- **Author:** Alexander Rush (Harvard NLP), with Vincent Nguyen and Guillaume Klein (Harvard)
- **URL:** https://nlp.seas.harvard.edu/2018/04/03/attention.html
- **Why:** The gold-standard line-by-line PyTorch implementation of "Attention Is All You Need." Walks through every component — MultiHeadAttention, positional encoding, encoder/decoder, training loop — in ~400 lines of runnable code. (harvard)

📹 **Video — "Let's build GPT: from scratch, in code, spelled out."**

- **Channel:** Andrej Karpathy
- **URL:** https://www.youtube.com/watch?v=kCc8FmEb1nY
- **Why:** A ~2-hour masterclass building a GPT from scratch in PyTorch covering nn.Module, training loops, self-attention, multi-head attention, residual connections, and layer normalization step by step. The single best resource for understanding transformer implementation in PyTorch.

---

18. HuggingFace Transformers

📄 **Article — "Fine-tuning a model with the Trainer API" (HuggingFace Course, Chapter 3)**

- **Author:** Hugging Face (Sylvain Gugger, Lysandre Debut, et al.)
- **URL:** https://huggingface.co/learn/llm-course/en/chapter3/3
- **Why:** The official HuggingFace course chapter on fine-tuning with the Trainer API. Covers TrainingArguments (learning rate, batch size, eval strategy), compute_metrics, gradient accumulation, mixed precision (fp16), and distributed training. Hugging Face Maintained by the HuggingFace team.

📹 **Video — "Fine-tune a pretrained model – Hugging Face Course"**

- **Channel:** Hugging Face (official)
- **URL:** https://www.youtube.com/watch?v=nvBXf7s7vTI
- **Why:** Official companion video walking through fine-tuning with the Trainer API, including TrainingArguments setup, data collators, evaluation metrics, and training loops. Embedded directly in the HuggingFace documentation. GitHub

---

19. Distributed training

📄 **Article — "How to Train Really Large Models on Many GPUs?"**

- **Author:** Lilian Weng (OpenAI / Lil'Log)
- **URL:** https://lilianweng.github.io/posts/2021-09-25-train-large/
- **Why:** The definitive reference on training parallelism. Comprehensively covers data parallelism (sync/async, DDP), model parallelism, pipeline parallelism (GPipe, PipeDream), tensor parallelism (Megatron-LM), ZeRO optimizer, mixed precision training, activation recomputation, and CPU offloading. Lil'Log github

🎥 Video — "Intro to Distributed Data Parallel (DDP)" — PyTorch DDP Tutorial Series

- **Channel:** PyTorch (official)
- **URL:** https://www.youtube.com/watch?v=TibQO_xv1zc
- **Why:** Official PyTorch tutorial explaining data parallelism concepts, the ring all-reduce algorithm, DistributedDataParallel vs DataParallel, and practical multi-GPU training. PyTorch Part of PyTorch's official DDP video series referenced on their docs. RC Learning Portal

---

20. Contrastive search decoding

📄 Article — "Generating Human-level Text with Contrastive Search in Transformers 🤗"

- **Author:** Yixuan Su & Tian Lan (Hugging Face Blog)
- **URL:** https://huggingface.co/blog/introducing-csearch
- **Why:** The official HuggingFace blog post introducing contrastive search decoding. Explains the NeurIPS 2022 paper, GitHub the degeneration penalty formula — score(v) = (1 − α) × p(v|context) − α × max_cosine_similarity Hugging Face — and provides side-by-side comparisons with greedy, beam, top-k, and nucleus sampling GitHub with runnable code.

🎥 Video — No dedicated video from well-known educators currently exists

- **Note:** Contrastive search is a relatively new and specialized decoding method Hugging Face (NeurIPS 2022). No dedicated explainer video from preferred educators (3Blue1Brown, StatQuest, Karpathy, etc.) is available yet. The HuggingFace blog post with its interactive Colab notebook is currently the best educational resource. The original paper is at https://arxiv.org/abs/2202.06417.

How these resources fit together

The resources above form a natural curriculum arc. Start with **foundational representations** (topics 13, 12) to understand how text becomes numbers. Move to **sequence modeling** (topic 10) and the **attention revolution** (topics 1–4). Then explore the **major architectures** — BERT (5), T5 (6), GPT (7) — and the **learning paradigms** that use them (8, 9). The **generation and reliability** topics (11, 14, 20) address practical deployment. **Training at scale** (15, 16, 19) and **engineering tooling** (17, 18) round out the practical skills needed to train and deploy models.

A few educators appear repeatedly because their work is genuinely best-in-class: **Jay Alammar's** Jay Alammar illustrated series covers foundational architectures with unmatched visual clarity, **Andrej Karpathy's**

Andrej Karpathy "Deep Dive into LLMs" video is a single 3.5-hour resource that touches nearly half these topics at expert level, Medium X **3Blue1Brown's** attention video covers topics 1, 3, and 4 in one beautifully animated presentation, Google Groups and **Lilian Weng's Lil'Log** remains the most authoritative blog for advanced topics like RLHF, hallucinations, distributed training, and prompt engineering.