Welcome to NLP 2025

From N-grams to Transformers: Your Journey Begins Today

NLP Course 2025

First Day of Class

Natural Language Processing: 12 Weeks to Understanding ChatGPT

What You'll Build This Semester

From Zero to Transformer

Week 2:
- Word embeddings from scratch
- Train on real text
- Visualize word similarities

Week 3:
- LSTM language model
- Generate Shakespeare
- Understanding memory

Week 5:
- **Complete transformer**
- Self-attention mechanism
- The architecture behind ChatGPT

Week 6:
- Fine-tune BERT
- Transfer learning
- State-of-the-art results

Week 9:
- Text generation strategies
- Control creativity
- Beam search and sampling

Week 12:
- Deploy responsibly
- Measure and mitigate bias
- **Ethical AI**

Why This Course Matters

NLP is Transforming Every Industry

Current Applications:

- **ChatGPT & Claude** - Conversational AI
- **GitHub Copilot** - Code generation
- **Google Translate** - 100+ languages
- **Grammarly** - Writing assistance
- **Alexa & Siri** - Voice assistants
- **Gmail** - Smart compose

Career Impact:

- NLP engineers: Top 5 percent salaries
- ML researchers: High demand
- Data scientists: Essential skill
- Product managers: Understanding capability
- All roles: AI literacy critical

Market Growth:

- NLP market: USD 20B (2024)
- Projected: USD 100B+ (2030)
- AI investment: USD 500B+ annually

Understanding transformers is understanding the future of computing

Our Learning Philosophy

Learn by Building, Not Just Watching

Discovery-Based Learning:

- Start with concrete problems
- Discover solutions yourself
- THEN learn the formal theory
- Build intuition first

Progressive Complexity:

- Week 1: Count words
- Week 5: Build transformer
- Week 12: Deploy ethically
- Each week builds on previous

Hands-On Practice:

- 12 lab notebooks
- Real code, real data
- Run in your browser
- No magic black boxes

Real Implementation:

- PyTorch from scratch
- Every line explained
- No hidden abstractions
- Understanding not memorization

By the end, you'll have IMPLEMENTED a transformer, not just understood it

Your 12-Week Journey

Three Phases of Mastery

Phase 1: Foundations
Weeks 1-4

Week 1: Statistical LM
N-grams, probability
Week 2: Word Embeddings
Word2Vec, GloVe
Week 3: RNN/LSTM
Sequential models
Week 4: Seq2Seq
Attention mechanism

Master: Neural basics

Phase 2: Revolution
Weeks 5-8

Week 5: Transformers
Self-attention
Week 6: Pre-trained
BERT, GPT
Week 7: Advanced
T5, GPT-3, scaling
Week 8: Tokenization
BPE, WordPiece

Master: Modern architectures

Phase 3: Application
Weeks 9-12

Week 9: Decoding
Generation strategies
Week 10: Fine-tuning
Adaptation methods
Week 11: Efficiency
Optimization
Week 12: Ethics
Responsible deployment

Master: Production deployment

Each phase unlocks new capabilities - by Week 12, you're deployment-ready

Prerequisites: Do You Have What It Takes?

Self-Assessment

Required (Must Have):

- Python programming experience
- Basic linear algebra (vectors, matrices)
- Basic probability theory
- Comfortable with loops and functions
- Can read and write code

Helpful But Not Required:

- PyTorch or TensorFlow
- Neural networks basics
- Machine learning concepts
- Backpropagation understanding
- GPU programming

If You Don't Have These:

- Python: Complete Python tutorial first
- Math: Review Khan Academy linear algebra
- Probability: Review basic probability

We Provide:

- Neural Network Primer (if needed)
- LSTM Primer (deep dive)
- Progressive difficulty
- Office hours support

New to neural networks? Start with our Neural Network Primer before Week 2

Course Materials: Everything You Need

Complete Open-Source Course

Presentations:

- 60+ slide decks
- Optimal readability design
- Multiple versions (BSc, enhanced)
- PDF and LaTeX source

Lab Notebooks:

- 12 interactive Jupyter notebooks
- Hands-on implementation
- Real data and models
- Run in your browser

Handouts:

- Pre-class discovery exercises
- Post-class technical practice
- Student and instructor versions

Supplementary:

- Neural Network Primer
- LSTM Primer (32 slides)
- Word Embeddings Module
- 8 visualization notebooks

Documentation:

- Complete installation guide
- Week-by-week course index
- Troubleshooting support
- Project templates

All Materials:

- Open source (MIT license)
- GitHub repository
- Always accessible

Getting Started TODAY

Your First Assignment (Due Next Class)

Step 1: Clone Repository

- `git clone github.com/josterri/2025_NLP_Lectures.git`
- `cd 2025_NLP_Lectures`

Step 2: Install Dependencies

- `pip install -r requirements.txt`

Step 3: Verify Installation

- `python verify_installation.py`

Expected Time:

- Clone: 2 minutes
- Install: 10-15 minutes
- Verify: 30 seconds

Step 4: Test First Notebook

- Launch: `jupyter lab`
- Open: Week 2 word embeddings lab
- Run first 3 cells
- Verify imports work

GPU Optional:

- Weeks 1-4: CPU sufficient
- Weeks 5+: GPU recommended
- All labs work on CPU
- GPU speeds up training

Get Help:

- Read INSTALLATION.md
- Check GitHub Issues
- Office hours: TBD

Week-by-Week Schedule

What Happens Each Week

Before Class (Monday):

- Read handout (pre-class section)
- Complete discovery exercises
- Build intuition
- Identify questions

During Lecture (Wednesday):

- Review key concepts
- Live coding demonstrations
- Q&A on pre-class material
- Preview lab notebook

Lab Session (Friday):

- Implement concepts
- Hands-on coding
- Get help from TAs

After Lab (Weekend):

- Complete lab notebook
- Finish post-class handout
- Experiment and explore
- Submit by Monday

Office Hours (TBD):

- Individual help
- Concept clarification
- Project discussion
- Career advice

Weekly Commitment:

- Lecture: 2 hours
- Lab: 2 hours
- Independent work: 4-6 hours

How You'll Be Assessed

Grading Breakdown

Lab Notebooks (40 percent):

- 12 notebooks, 11 graded (drop lowest)
- Due Monday after lab
- Implementation correctness
- Code quality and comments
- Experimentation and insights

Midterm Project (25 percent):

- After Week 6
- Implement and evaluate model
- Written report
- Code submission

Final Project (30 percent):

- Weeks 11-12
- Original NLP application

Participation (5 percent):

- Class attendance
- Lab participation
- Office hours engagement
- Helping classmates

Grading Philosophy:

- Focus on learning not perfection
- Partial credit for attempts
- Bonus for creativity
- Collaboration encouraged (cite sources)

Late Policy:

- 48-hour grace period (no penalty)
- After that: 10 percent per day
- Max 5 days late

Example Final Projects

What Past Students Built

Text Generation:

- **Poetry generator** - Style transfer
- **Code completion** - GitHub-trained
- **Story writer** - Character-consistent
- **Email composer** - Professional tone

Information Extraction:

- **Resume parser** - Extract skills
- **News summarizer** - Multi-document
- **Question answering** - Domain-specific
- **Fact checker** - Claim verification

Classification:

- **Sentiment analysis** - Product reviews
- **Spam detection** - Email filtering
- **Intent classification** - Chatbot

Translation & Multilingual:

- **Domain translator** - Legal/medical
- **Code-switching** - Multilingual text
- **Style transfer** - Formal to casual
- **Dialect conversion** - Regional

Creative Applications:

- **Music from text** - Lyrics to melody
- **Debate partner** - Argument generation
- **Language learning** - Adaptive tutor
- **Game dialogue** - NPC conversations

Research Projects:

- **Bias measurement** - Gender/race
- **Interpretability** - Attention analysis
- **Efficiency** - Model compression

Milestone: What You'll Know By Week 5

From Zero to Transformer in 5 Weeks

Technical Understanding:

- How attention mechanism works
- Query, Key, Value matrices
- Multi-head attention computation
- Positional encoding mathematics
- Layer normalization
- Feed-forward networks
- Residual connections

Implementation Skills:

- Scaled dot-product attention
- Multi-head splitting and merging
- Complete transformer block
- Positional encoding
- Training loop

Practical Abilities:

- Debug attention weights
- Visualize attention patterns
- Optimize hyperparameters
- Compare with RNN/LSTM
- Understand GPT architecture
- Read research papers

Real-World Knowledge:

- Why ChatGPT is so powerful
- How BERT differs from GPT
- Computational complexity trade-offs
- When to use transformers
- Current limitations
- Future directions

Common Questions

Frequently Asked Questions

Q: Do I need a GPU?
A: No, all labs work on CPU. GPU speeds things up for Weeks 5+ but isn't required.

Q: How much Python?
A: Comfortable writing functions, loops, and classes. PyTorch will be taught.

Q: Mathematical background?
A: Linear algebra (vectors, matrices, dot products) and basic probability. We'll review as needed.

Q: Can I audit?
A: Yes! All materials are open source. You won't get grades but can follow along.

Q: Group projects?
A: Labs are individual. Final project can be pairs with instructor approval.

Q: How hard is this?
A: Challenging but doable. Budget 10 hours/week. We support you every step.

Q: What if I fall behind?
A: Office hours, catch-up sessions, and grace periods. Communication is key.

Q: Industry vs research?
A: Both! Course covers practical deployment and research methods.

Q: After this course?
A: You'll be ready for NLP engineering roles, ML research, or advanced courses.

Q: Most important week?
A: Week 5 (transformers) is the foundation. Don't miss it!

Strategies for Success

How Top Students Excel

Before Each Week:

- Read pre-class handout carefully
- Try exercises before looking at solutions
- Write down questions
- Review previous week's concepts

During Lab:

- Start early (don't wait until deadline)
- Experiment beyond requirements
- Ask questions immediately
- Help classmates (best way to learn)
- Save your work frequently

Study Groups:

- Form groups of 3-4 students
- Meet weekly to discuss concepts

Debugging Mindset:

- Read error messages carefully
- Print intermediate outputs
- Test with small examples first
- Check dimensions and shapes
- Use debugger or print statements

Going Deeper:

- Read suggested papers
- Try bonus challenges
- Implement variants
- Share findings with class

Stay Organized:

- Keep notes for each week
- Maintain code repository

Where to Get Help

Course Materials:

- **GitHub:** github.com/josterri/2025_NLP_Lectures
- **Documentation:** README.md, COURSE_INDEX.md
- **Installation:** INSTALLATION.md
- **Syllabus:** SYLLABUS.md

Getting Help:

- **Office Hours:** TBD
- **Email:** TBD
- **Discussion Forum:** TBD
- **Lab TAs:** Available during lab sessions

External Resources:

- **PyTorch Tutorials:** pytorch.org/tutorials
- **Papers:** arxiv.org
- **Blogs:** Jay Alammar, Lil'Log

Community:

- **GitHub Issues:** Report bugs
- **Pull Requests:** Contribute improvements
- **Discussions:** Share insights

Reference Books:

- Speech and Language Processing (Jurafsky & Martin)
- Neural Network Methods for NLP (Goldberg)
- Dive into Deep Learning (Zhang et al)
- All available free online

Tools:

- **Google Colab:** Free GPU notebooks
- **Hugging Face:** Pre-trained models
- **Weights & Biases:** Experiment tracking

Your Immediate Action Items

What to Do Right Now

Today (Next 30 Minutes):

1. Clone the repository
2. Read the README.md
3. Check system requirements
4. Identify if you need GPU

Tonight (1-2 Hours):

1. Install dependencies (requirements.txt)
2. Run verify_installation.py
3. Fix any installation issues
4. Read INSTALLATION.md if stuck

Before Next Class:

1. Complete Neural Network Primer (if needed)
2. Read Week 2 pre-class handout
3. Try the discovery exercises
4. Prepare questions

Optional (For Eager Students):

1. Browse Week 2 lab notebook
2. Read Week 1 presentation
3. Check out supplementary modules
4. Join discussion forum

Most important: Get your environment working BEFORE Week 2

Why Transformers Changed Everything

Before Transformers (Pre-2017):

- Sequential processing (RNNs)
- Slow training (hours to days)
- Limited context (hundreds of words)
- Vanishing gradient problems
- Hard to parallelize
- Specialized architectures per task

Limitations We Hit:

- Couldn't scale to large models
- Couldn't use massive datasets
- Couldn't capture long-range dependencies
- Transfer learning was limited

After Transformers (2017+):

- Parallel processing (attention)
- Fast training (hours on GPUs)
- Unlimited context (thousands of tokens)
- Stable gradients
- Massively parallelizable
- Universal architecture

What Became Possible:

- GPT-3: 175 billion parameters
- GPT-4: Multimodal understanding
- Claude: 100k+ token context
- ChatGPT: Conversational AI
- All modern LLMs use transformers

You'll learn WHY this architecture unlocked the AI revolution

What You'll Be Able to Do

By the End of This Course

Technical Skills:

- Implement transformers from scratch
- Fine-tune BERT and GPT models
- Debug attention mechanisms
- Optimize model efficiency
- Deploy models responsibly
- Measure and mitigate bias

Conceptual Understanding:

- How ChatGPT works internally
- Why attention beats RNNs
- When to use which architecture
- Trade-offs in model design
- Current research frontiers

Practical Abilities:

- Read and implement research papers
- Design NLP systems
- Evaluate model performance
- Choose appropriate methods
- Debug complex models
- Communicate technical concepts

Career Readiness:

- NLP engineer interviews
- ML research positions
- Data science roles
- PhD program preparation
- Startup technical co-founder
- Technical leadership

Next Week Preview: Word Embeddings

Your First Hands-On Lab

What You'll Learn:

- Words as vectors
- Semantic similarity
- Word2Vec algorithm (CBOW & Skip-gram)
- Training on real text
- Visualizing embeddings
- Analogies: king - man + woman = queen

What You'll Build:

- Train word embeddings from scratch
- Implement Skip-gram model
- Visualize word relationships
- Discover semantic patterns
- Test word analogies

Prepare By:

- Reading pre-class handout
- Understanding distributional hypothesis
- Reviewing basic neural networks
- (Optional) Neural Network Primer

Lab Highlights:

- Work with real text corpus
- See embeddings evolve during training
- Interactive visualizations
- Explore semantic relationships
- Bonus challenges available

Welcome to NLP 2025

Key Takeaways:

- **Build transformers** - Understand how ChatGPT works
- **12-week journey** - From foundations to deployment
- **Hands-on learning** - Implement everything yourself
- **Real applications** - Deploy models responsibly
- **Complete support** - Extensive materials and help

First Assignment:

Install environment by next class
Run verify_installation.py successfully
Read Week 2 pre-class handout