

Master Template Showcase

Optimal Readability Design System

Natural Language Processing Course 2025

Department of Computer Science

September 20, 2025

Presentation Overview

1. Optimal Readability Color System
2. Standard Slide Layouts
3. Chart and Visualization Integration
4. Code and Algorithm Display
5. Tables and Data Presentation
6. Mathematical Notation and Formulas
7. Educational Components
8. Comparison and Analysis
9. Template Best Practices
10. Summary and Resources

Color Palette - WCAG AAA Compliant

Primary Colors:

- Pure Black (21:1 contrast)
- Deep Blue (12.6:1 contrast)
- Dark Gray (9.7:1 contrast)

Chart Colors (Colorblind-Safe):

- Chart Blue - Primary data
- Chart Orange - Secondary data
- Chart Teal - Tertiary data
- Chart Purple - Quaternary data

Semantic Colors:

- Dark Green - Success/Positive
- Dark Red - Warning/Negative
- Light Gray - Borders/Grids

All colors meet WCAG AAA standards for accessibility

Two-Column Layout

Left Column Content:

- Primary information
- Key concepts
- Main arguments
- Core definitions

This layout is ideal for:

1. Comparisons
2. Before/after scenarios
3. Theory vs practice
4. Problem/solution pairs

Right Column Content:

- Supporting details
- Examples
- Visualizations
- Code snippets

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Secondary text appears in gray for visual hierarchy

Three-Column Layout

Column 1

- Concept A
- Detail 1
- Detail 2

Advantages:

- Pro 1
- Pro 2

Column 2

- Concept B
- Detail 1
- Detail 2

Limitations:

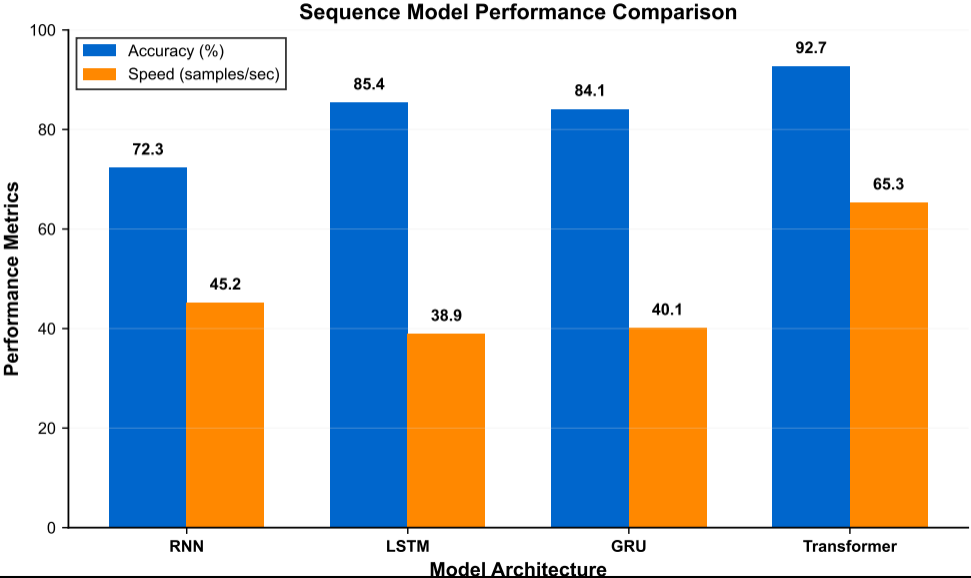
- Con 1
- Con 2

Column 3

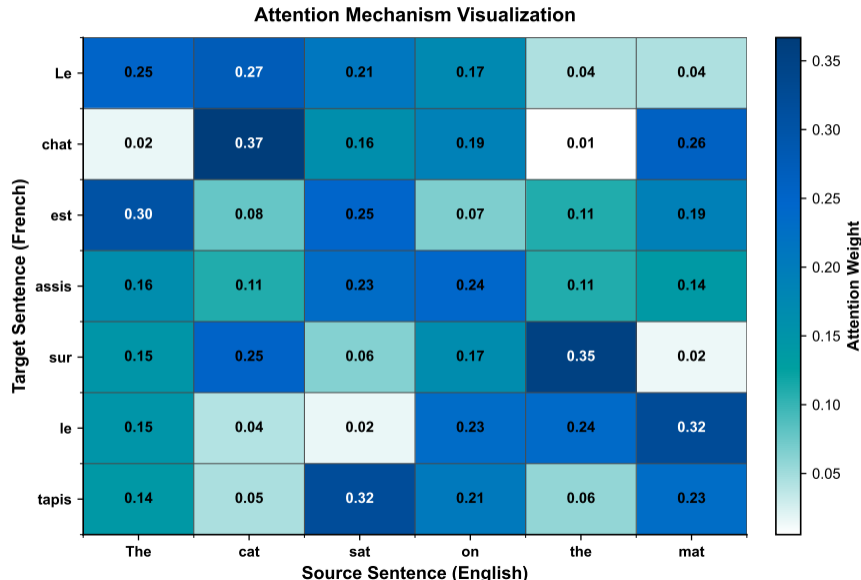
- Concept C
- Detail 1
- Detail 2

Results:

- Finding 1
- Finding 2



Scaled Chart with Caption



Concept Slide with Visualization

Encoder-Decoder Architecture

Key components:

- **Encoder:** Processes input sequence
- **Context Vector:** Fixed-size representation
- **Decoder:** Generates output sequence

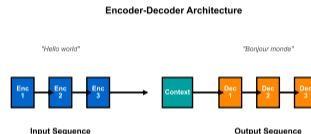
Mathematical formulation:

$$h_t = f(x_t, h_{t-1})$$

$$c = q(h_1, h_2, \dots, h_T)$$

$$s_t = g(y_{t-1}, s_{t-1}, c)$$

Note: Variable length input/output capability



Multi-Panel Dashboard

Training Dashboard - Seq2Seq Model



Code Display with Syntax Highlighting

```
1 import torch
2 import torch.nn as nn
3
4 class Seq2Seq(nn.Module):
5     def __init__(self, encoder, decoder, device):
6         super().__init__()
7         self.encoder = encoder
8         self.decoder = decoder
9         self.device = device
10
11     def forward(self, src, trg, teacher_forcing_ratio=0.5):
12         # Encoder processes entire source sequence
13         encoder_outputs, hidden = self.encoder(src)
14
15         # Decoder generates target sequence
16         outputs = []
17         input = trg[0] # <SOS> token
18
19         for t in range(1, trg.shape[0]):
20             output, hidden = self.decoder(input, hidden, encoder_outputs)
21             outputs.append(output)
22
23             # Teacher forcing decision
24             use_teacher_force = random.random() < teacher_forcing_ratio
25             input = trg[t] if use_teacher_force else output.argmax(1)
26
27         return torch.stack(outputs)
```

Performance Comparison Table

Model	BLEU	Speed	Memory
RNN	23.5	Fast	Low
LSTM	31.2	Medium	Medium
GRU	30.8	Medium	Medium
Transformer	41.3	Slow	High

Transformer achieves best quality at computational cost

Complex Table with Multiple Sections

Architecture	Training			Inference		
	Time	GPU	Batch	Speed	Memory	Latency
Sequential Models						
RNN	2h	4GB	32	1000/s	1GB	5ms
LSTM	3h	8GB	32	800/s	2GB	6ms
GRU	2.5h	6GB	32	850/s	1.5GB	5.5ms
Attention Models						
Seq2Seq+Attn	4h	12GB	16	400/s	3GB	12ms
Transformer	8h	16GB	8	200/s	4GB	20ms

Performance metrics across different model architectures and operational phases

Mathematical Expressions

Probability Notation:

- Joint: $P(A, B)$
- Conditional: $P(A \mid B)$
- Marginal: $P(A) = \sum_b P(A, B = b)$

Optimization:

$$\theta^* = \operatorname{argmax}_{\theta} \mathcal{L}(\theta)$$

$$\hat{y} = \operatorname{argmax}_y P(y \mid x)$$

Neural Network Operations:

Attention mechanism:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

Context vector:

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

Output distribution:

$$P(y_t) = \operatorname{softmax}(W_o \cdot s_t)$$

All mathematical notation follows standard ML conventions

Text Highlighting and Emphasis

Highlighting Commands:

- **Primary highlight** - key concepts
- Secondary text - supporting info
- **Success message** - positive outcomes
- **Warning text** - important cautions
- **Data reference** - metrics/values
- **Alternative data** - comparisons

Usage Examples:

The **attention mechanism** solved the **bottleneck problem** in sequence models.

Performance improved from **23.5 BLEU** to **41.3 BLEU**.

Key achievement: Variable-length sequences

Note: Results may vary with dataset size

Consistent color coding enhances comprehension and retention

Side-by-Side Comparison

Traditional RNN

- Sequential processing
- Fixed context window
- Gradient vanishing
- Fast inference
- Low memory usage

$$h_t = \tanh(W_h h_{t-1} + W_x x_t)$$

Strengths:

- Simple implementation
- Efficient for short sequences

Transformer

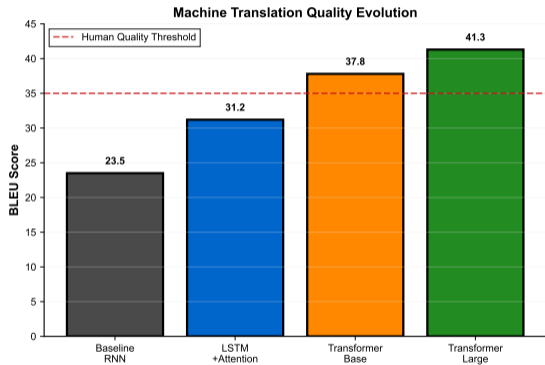
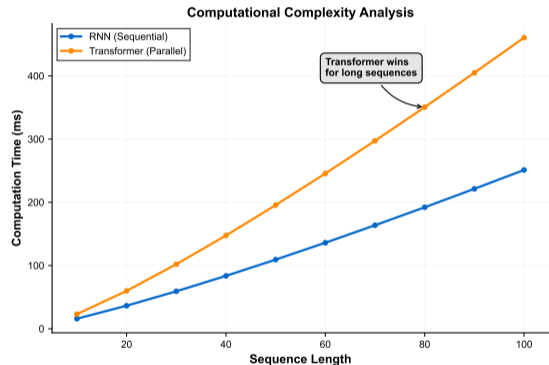
- Parallel processing
- Global context access
- Stable gradients
- Slower inference
- High memory usage

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Strengths:

- Superior for long sequences
- Better context modeling

Performance Analysis Charts



Left: Computational complexity — Right: Translation quality evolution

Design Principles

Typography Guidelines:

- 8pt base font size for readability
- Bold for emphasis, not color alone
- Consistent heading hierarchy
- Maximum 3 font sizes per slide

Color Usage:

- Black text on white background
- Accents for structure, not decoration
- Semantic colors for meaning
- Colorblind-safe chart palettes

Layout Principles:

- Maximum 3 key points per slide
- Consistent spacing and alignment
- Visual hierarchy through size/weight
- White space for clarity

Content Organization:

- Clear section divisions
- Logical flow and progression
- Summary/takeaway boxes
- Minimal text, maximum impact

Simplicity and consistency enhance learning outcomes

Template Features Summary

- **WCAG AAA compliant** color system
- **Colorblind-safe** visualization palette
- **Multiple layout** templates
- **Integrated chart** generation system
- **Mathematical notation** support
- **Code highlighting** with Python/LaTeX
- **Semantic color** coding
- **Accessibility-first** design

Key Takeaway

Resources and Files

Core Template Files:

- `master_template.tex`
- `chart_utils.py`
- `generate_charts.py`

Documentation:

- Layout command reference
- Color palette guide
- Chart generation examples
- Best practices checklist

Example Usage:

- Week 4: Seq2Seq models
- Attention mechanisms
- Training dashboards
- Performance comparisons

File Organization:

- `figures/` - Generated charts
- `scripts/` - Python utilities
- `previous/` - Version archive

This template ensures consistent, accessible presentations across the course

Thank You

Questions?

Template Version 1.0 - Optimal Readability

Natural Language Processing Course 2025