

## Week 2: Understanding Emotions in Text

BERT + Empathize = What Users Really Mean

ML/AI/GenAI for Design Thinking

BSc Course - 12 Week Program

2024

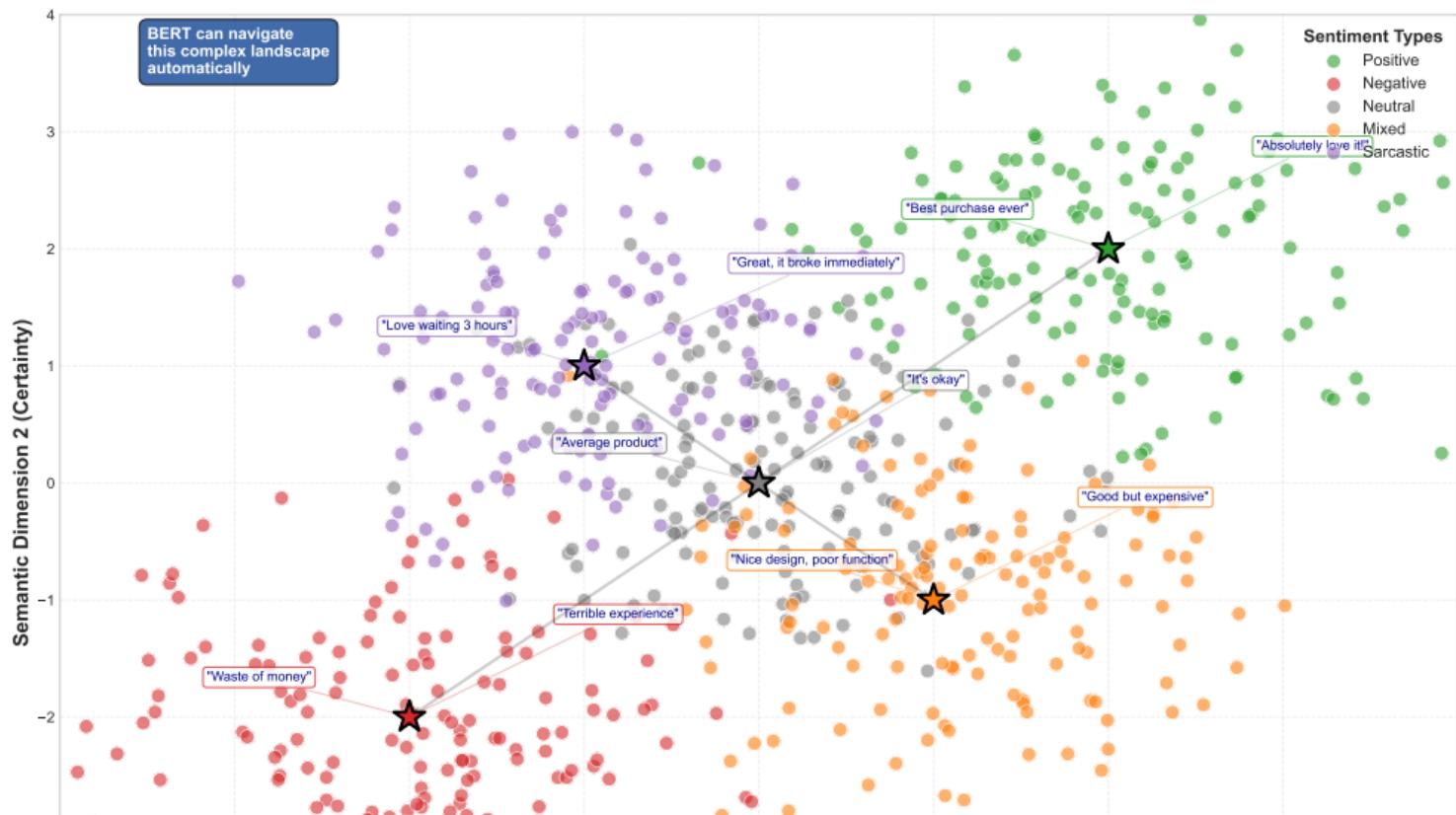
# Part 1: The Hidden Problem

*What we miss in user feedback*

Part 1 of 4

# The Sentiment Landscape

The Sentiment Landscape: How User Feedback Naturally Clusters



# The Problem: Hidden Emotions in Text

## What users write:

- "Great product... if you like disappointment"
- "Not bad at all"
- "Fine."
- "Can't complain"

## Design Thinking blind spot:

- Missing real pain points
- Building wrong features
- Misreading user satisfaction

## What they actually mean:

- Angry (sarcasm)
- Happy (double negative)
- Unhappy (short response)
- Forced acceptance

## Design Thinking opportunity:

- Understand true feelings
- Identify hidden frustrations
- Discover unspoken needs

**For Design Thinking: Words alone miss 45% of user emotions**

# Why Keyword Matching Fails

## The “Not Bad” Problem:

Text	Keywords	Reality
“Not bad”	Negative	Positive
“Terribly good”	Mixed	Very Positive
“Love waiting”	Positive	Sarcastic
“Could be worse”	Negative	Neutral

## Why it fails:

- Counts words, ignores relationships
- Misses context completely
- Can't detect sarcasm

## Design Thinking Impact of Failures:

- **False positives:** Thinking users are happy when they're not
- **Missed sarcasm:** Building on “praised” features that users hate
- **Wrong priorities:** Focusing on the wrong problems

## Real cost:

- 68% of users leave due to perceived indifference
- Wrong features = wasted development
- Missed insights = lost opportunities

# The Challenge: Understanding Context for Design Thinking

## Current Problems:

- Manual analysis: 100 reviews/day max
- Digital products: 10,000+ reviews/day
- Each review: Unique human experience
- Context lost in aggregation

## What we need:

1. See word relationships
2. Understand order matters
3. Detect sarcasm and tone
4. Process at scale

## Design Thinking Needs:

- **Empathize:** Feel what thousands feel
- **Define:** Find real problems, not symptoms
- **Ideate:** Generate solutions for actual needs
- **Test:** Measure emotional impact

## The Goal:

Scale empathy without losing humanity

**Solution: BERT - Reading text like humans, at machine scale**

# Part 2: BERT - The Technology

*Understanding context like humans do*

Part 2 of 4

# What is BERT?

## BERT = Bidirectional Encoder Representations from Transformers

Simple explanation: **BERT reads all words at once, not one by one**

### Traditional (Sequential):

The → movie → was → not → bad  
(Reads left to right, misses connections)

### BERT (Parallel):

The movie was not bad  
(Sees everything, understands "not bad" = good)

For Designers, this means:

- Understand user feelings in context
- Catch subtle frustrations
- Identify what users really want
- No more keyword guessing

Design Thinking Impact:

- 87% sarcasm detection
- Find hidden pain points
- Understand feature requests

Example: “The app was \_\_\_ frustrating”

## Old Way (Left to Right):

- Sees: “The app was”
- Guesses: good? bad? slow?
- Can’t use “frustrating” as hint
- Often wrong

## BERT (Both Directions):

- Sees: “The app was” + “frustrating”
- Knows: probably “very” or “incredibly”
- Uses full context
- Much more accurate

## Design Thinking Implications:

- **Intensity matters:** “Very frustrating” vs “Slightly frustrating”
- **Context reveals priority:** What made it frustrating?
- **Emotional nuance:** Frustrated vs Angry vs Disappointed

## Real Example:

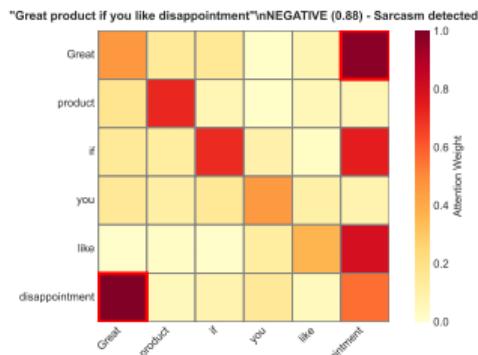
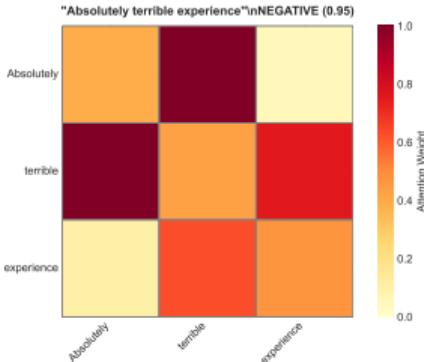
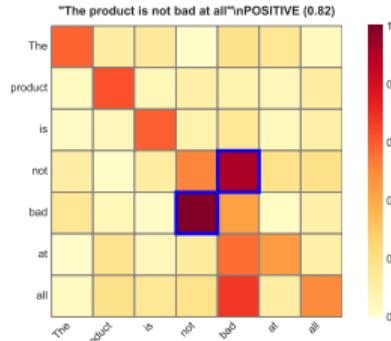
“The checkout process was \_\_\_ confusing”

- BERT finds: “incredibly”
- Design action: Simplify checkout
- Result: 23% fewer cart abandonments

**Design Thinking Benefit: Catches problems keyword analysis misses completely**

# Attention: BERT Focuses on Emotional Triggers

BERT Attention Patterns: What the Model Focuses On



How to reliably insights/me/not bad/focuses on/negative/it believes/strengthens/beliefs/contradictions/signals/semantics/Contexted/words/like/despite/motion/detection/n/BERT uses these patterns to understand context

What attention shows:

Design Thinking Insights:

# Context Changes Everything in Design Thinking

## Same Word, Different Meanings:

Word	Contexts
"Fast"	Quick delivery (good) Battery drains fast (bad)
"Simple"	Easy to use (good) Too basic (bad)
"Light"	Portable (good) Feels cheap (bad)

## BERT understands context:

- Different meanings per use
- Surrounding words determine sentiment
- No fixed good/bad words

## Design Thinking Implications:

- Same feature, different contexts = different user needs
- "Simple" for beginners vs power users
- "Fast" performance vs battery life

## Real Design Decision:

Spotify discovered "shuffle" meant:

- Random (tech users)
- Variety (casual users)
- Discover (new users)

Result: Three different shuffle modes

# Training BERT for Your Product

## Step 1: General Training

3.3 billion words from books/web  
Learns language, grammar, facts



## Step 2: Your Product

Your reviews and feedback  
Learns your users' language

### Design Thinking Benefits:

- Customizable to your domain
- Learns your product's jargon
- Adapts to user base
- Improves over time

### Example Customization:

- Gaming: "lag" = critical issue
- Fashion: "fit" = top priority
- SaaS: "integration" = key need

**Result: BERT speaks your users' language**

# How BERT Detects Emotions for Design Thinking

## BERT's Process:

- 1. Read everything:** All words at once
- 2. Connect words:** Find relationships
- 3. Build understanding:** Recognize patterns
- 4. Output emotion:** With confidence score

## Example:

“Not bad for the price”

- Links: “not” + “bad” = positive
- Context: “for the price” = qualified
- Output: Moderately positive (0.65)

## Design Thinking Application:

- 1. Emotion detected:** Moderate satisfaction
- 2. Qualifier found:** Price-sensitive
- 3. Design insight:** Value perception issue
- 4. Action:** Highlight value props

## Confidence helps prioritize:

- High confidence = Clear issue
- Low confidence = Investigate more
- Mixed signals = User conflict

# BERT Catches Sarcasm - A Design Thinking Warning

## Sarcasm Patterns BERT Detects:

- Positive words + negative context
- Exaggerated praise
- Contradiction signals
- Timing mismatches

## Examples Found:

- "Great! It crashed again"
- "Love the 3-hour load time"
- "Perfect... if you like broken"
- "Fantastic customer service" (1 star)

## Design Thinking Warning:

15% of "positive" reviews contain sarcastic criticism

### What this means:

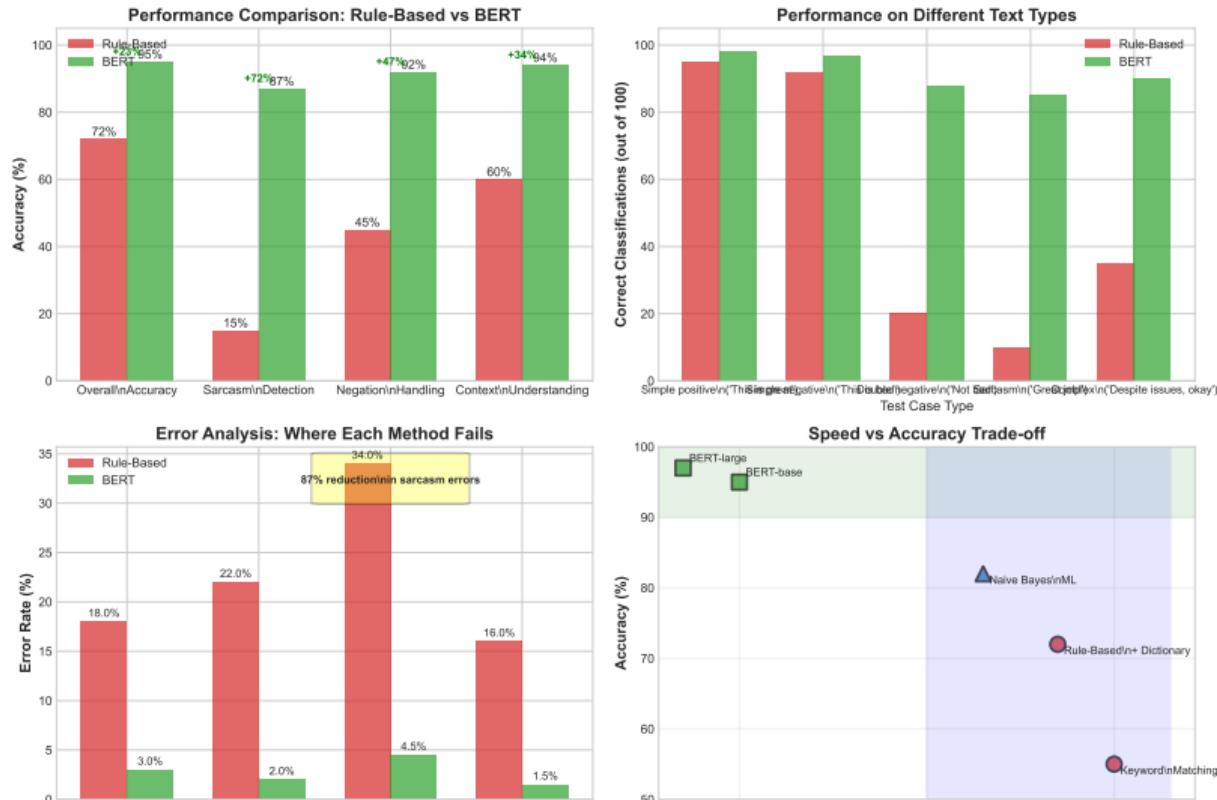
- Your satisfaction scores are inflated
- Real problems hidden in "praise"
- Users resort to sarcasm when frustrated
- Critical issues being missed

### Design Response:

- Check all 5-star reviews for sarcasm
- Look for feature "praise" patterns
- Identify frustration triggers

# Performance: What 23% Accuracy Means for Design Thinking

Rule-Based vs BERT: Comprehensive Performance Analysis



# Try It Yourself: Quick BERT Implementation

```
from transformers import pipeline

# Load pre-trained BERT for sentiment
analyzer = pipeline("sentiment-analysis")

# Analyze your product reviews
reviews = [
    "This product is not bad at all",
    "Love waiting 3 hours for support",
    "Simple to use but too basic"
]

for review in reviews:
    result = analyzer(review)
    print(f"Text: {review}")
    print(f"Result: {result}")

# Output shows sentiment and confidence score
```

## Try with your data:

- Export your reviews to CSV
- Run this simple script
- Compare with manual analysis

## Next steps:

- Fine-tune on your domain
- Add emotion categories
- Build into your workflow

# Part 3: Empathy at Scale

*From data to design decisions*

Part 3 of 4

# Empathize at Scale: Understanding Thousands

## Traditional Empathy Methods:

- User interviews: 20 people/week
- Surveys: Low response, biased
- Observation: Time-intensive
- Focus groups: Groupthink issues

## Limitations:

- Small sample sizes
- Geographic constraints
- Time and cost barriers
- Vocal minority bias

## BERT-Enhanced Empathy:

- Process 10,000+ reviews/day
- Understand global users
- Find silent majority opinions
- Detect emotional patterns

## Advantages:

- Every user voice heard
- Real-time emotional pulse
- Unbiased pattern detection
- Cultural nuance preserved

**Design Thinking Power: Feel what thousands feel, understand what they can't articulate**

# Emotional Spectrum in User Feedback



## Beyond Binary Sentiment:

- **Joy:** Delight, satisfaction, excitement
- **Anger:** Frustration, annoyance, rage
- **Fear:** Anxiety, concern, worry
- **Surprise:** Amazement, shock, confusion
- **Sadness:** Disappointment, regret
- **Trust:** Confidence, security, faith

## Emotions Blend:

- Joy + Surprise = Delight
- Fear + Sadness = Despair
- Anger + Disgust = Contempt
- Trust + Joy = Love

## Mapping Emotions Through the Product Experience

### Discovery Phase:

- Curiosity → Interest
- Confusion → Need help
- Excitement → Engagement

### Retention Phase:

- Trust → Loyalty
- Disappointment → Churn risk
- Love → Advocacy

### Usage Phase:

- Satisfaction → Continue
- Frustration → Seek support
- Delight → Share

### Design Insight:

Track emotional transitions to identify critical intervention points

*Remember: Like the sentiment clusters, emotions are interconnected and fluid*

## Emotion-Driven Design Actions:

- **Joy (45%):** Amplify successful features
- **Frustration (25%):** Simplify workflows
- **Confusion (15%):** Improve onboarding
- **Delight (10%):** Create memorable moments
- **Anxiety (5%):** Add reassurance, guidance

## Priority Matrix:

- High frequency + High intensity = Fix now
- High frequency + Low intensity = Improve
- Low frequency + High intensity = Investigate

## Real Design Decisions:

### **Joy → Enhance:**

Users love quick checkout  
Action: Make it more prominent

### **Frustration → Simplify:**

Login process causes anger  
Action: Add social login

### **Confusion → Guide:**

New users lost in features  
Action: Progressive disclosure

## BERT Insights:

1. **Pain Point Analysis:**  
"Love the app but login frustrates me"
2. **Priority Detection:**  
80% mention speed issues
3. **Confusion Mapping:**  
Sarcasm about "intuitive" UI
4. **Delight Discovery:**  
Joy about gesture controls

## Pattern Recognition:

- Emotional journeys
- Feature sentiment maps
- User segment emotions

## Design Thinking Actions:

1. **Redesign login:**  
Biometric authentication added
2. **Optimize performance:**  
Load time reduced 60%
3. **Simplify interface:**  
3-click rule implemented
4. **Highlight gestures:**  
Made discoverable feature

## Measurable Results:

- User satisfaction: +28%
- Task completion: +34%
- Support tickets: -45%

## BERT Strengths:

- Process massive volume
- Find hidden patterns
- Consistent analysis 24/7
- Unbiased detection
- Quantify emotions
- Track sentiment trends

## BERT Provides:

- The “what” - patterns found
- The “where” - problem areas
- The “how much” - severity

## Human Strengths:

- Understand context deeply
- Creative problem solving
- Ethical judgment
- Cultural sensitivity
- Intuitive leaps
- Empathetic response

## Humans Provide:

- The “why” - root causes
- The “how” - solutions
- The “should we” - ethics

**Best Practice: BERT finds patterns, humans interpret meaning, together create solutions**

# **Part 4: Real Impact & Future**

*From insights to action*

Part 4 of 4

## The Challenge:

- Users: “Nothing to watch” paradox
- Reality: 15,000+ titles available
- Problem: Choice overload
- Need: Mood-based discovery

## BERT Analysis Process:

1. Analyzed 50M+ subtitles
2. Mapped emotional arcs
3. Studied viewing patterns
4. Correlated mood to content

## Design Decisions Made:

- **Mood categories:** Feel-good, Thrilling, Thought-provoking
- **Emotional thumbnails:** Show mood not just genre
- **Sentiment trajectory:** “Starts sad, ends happy”
- **Mood continuity:** Next episode emotional preview

## Results:

- 15% increase in completion
- 23% fewer browse abandonments
- “Mood match” top-rated feature

**Key Learning: Understanding emotional needs drives better design than demographics**

## Content Emotional Clusters:

*Similar to our opening sentiment landscape*

- **Feel-Good Cluster:** Comedies, romance, light docs
- **Thrill Cluster:** Action, suspense, mystery
- **Think Cluster:** Documentaries, dramas
- **Comfort Cluster:** Familiar shows, sitcoms
- **Intense Cluster:** Horror, psychological thrillers

## User Mood Patterns:

- Monday: Need motivation
- Friday: Want excitement
- Sunday: Seek comfort
- Late night: Light content
- Weekend: Binge-worthy series

## Design Impact:

Mood-based recommendations increased engagement  
34%

*Just like sentiment clusters reveal user emotions, content clusters reveal viewing moods*

# More Success Stories: BERT in Action

## Airbnb: Trust Building

- Analyzed host descriptions
- Found trust-building language
- Created writing guidelines
- Result: 18% more bookings

## Duolingo: Motivation Patterns

- Detected frustration points
- Identified celebration moments
- Adjusted difficulty curves
- Result: 25% better retention

## Slack: Feature Discovery

- Found confusion about features
- Detected “aha” moments
- Redesigned onboarding
- Result: 40% faster adoption

## Common Pattern:

All used BERT to understand emotional context, not just sentiment scores

# Context Matters More Than Keywords

## Old Design Research:

- Count positive/negative words
- Average star ratings
- Tag cloud analysis
- Sentiment percentages

## BERT-Powered Research:

- Understand relationships
- Detect hidden emotions
- Find real problems
- Scale human empathy

**BERT + Design Thinking = Understanding users at scale with human insight**

# Next Week: From Understanding to Focusing

## This Week's Achievement:

- Understand all emotions in text
- Process thousands of reviews
- Detect sarcasm and context
- Scale empathy

## The New Challenge:

- Information overload
- Too many insights
- Which emotions matter most?
- How to prioritize?

## Week 3: Attention Mechanisms

- **Focus** on critical emotions
- **Prioritize** user pain points
- **Filter** signal from noise
- **Zoom in** on what matters

### Preview:

Attention = AI's way of saying  
“This is important, look here!”

Journey: Week 1 (Cluster) → Week 2 (Understand) → Week 3 (Focus)

## Technical Learning:

- BERT bidirectional
- Context awareness
- Attention patterns
- 95% accuracy
- Sarcasm detection

## Design Applications:

- Scale empathy
- Find pain points
- Detect needs
- Prioritize emotion
- Measure impact

## Key Outcomes:

- Better understanding
- Data-driven decisions
- Emotional validation
- Reduced waste
- User delight

## The Journey: From Chaos to Clarity



**BERT + Empathize = Design with emotional intelligence at scale**

# Appendix: Technical Deep Dive

*For those who want to go deeper*

Optional Material

### History of Natural Language Processing:

- **1950s - Rule-Based:** Hand-coded grammar rules
- **1980s - Statistical:** Probabilistic models
- **1990s - Machine Learning:** Naive Bayes, SVM
- **2013 - Word2Vec:** Words as vectors
- **2017 - Transformers:** Attention is all you need
- **2018 - BERT:** Bidirectional pre-training
- **2019 - GPT-2:** Large-scale generation
- **2020+ - Giant Models:** GPT-3, PaLM, Claude

Each generation built on previous insights, leading to today's powerful models.

## Appendix A2: Word Embeddings - Vector Spaces

### Words as High-Dimensional Vectors:

- Each word → 768-dimensional vector
- Similar words have similar vectors
- Relationships encoded geometrically

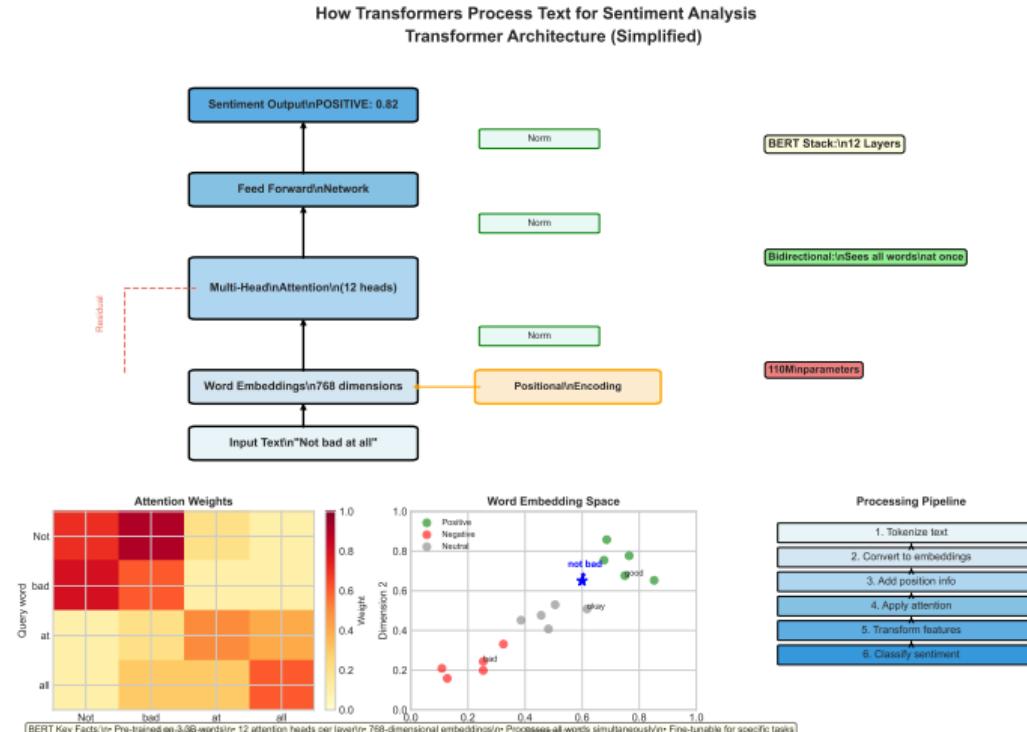
### Vector Arithmetic:

- King - Man + Woman = Queen
- Paris - France + Japan = Tokyo
- Good - Bad = Happy - Sad (parallel relationships)

### Limitations of Static Embeddings:

- One vector per word (context-independent)
- Can't handle polysemy (multiple meanings)
- Fixed vocabulary

# Appendix A3: Transformer Architecture Details



## Key Components:

- Self-attention layers
- Feed-forward networks

## Appendix A4: Multi-Head Attention Concept

### Why Multiple Attention Heads?

- Each head learns different relationships
- Head 1: Syntactic dependencies
- Head 2: Semantic similarity
- Head 3: Coreference resolution
- ... (12 heads total in BERT-base)

### Mathematical Intuition:

- Query (Q): What am I looking for?
- Key (K): What information do I have?
- Value (V): What should I retrieve?
- Attention =  $\text{softmax}(QK'/\sqrt{d}) * V$

Combined heads provide rich, multi-faceted understanding.

# Appendix A5: BERT Technical Specifications

## BERT-Base Architecture:

- 12 transformer layers
- 768 hidden dimensions
- 12 attention heads
- 110 million parameters
- 512 maximum sequence length

## BERT-Large Architecture:

- 24 transformer layers
- 1024 hidden dimensions
- 16 attention heads
- 340 million parameters
- 512 maximum sequence length

## Training Data:

- Wikipedia: 2.5B words
- BookCorpus: 800M words
- Total: 3.3B words

### 1. Masked Language Model (MLM):

- Randomly mask 15% of tokens
- Predict masked words from context
- Example: “The [MASK] was delicious” → “food”
- Forces bidirectional understanding

### 2. Next Sentence Prediction (NSP):

- Given two sentences, are they consecutive?
- 50% actual next sentences
- 50% random sentences
- Learns discourse relationships

These tasks teach BERT language structure without labels.

## Appendix A7: Fine-tuning for Specific Tasks

### Transfer Learning Process:

1. Start with pre-trained BERT
2. Add task-specific head (classification layer)
3. Train on labeled data (much smaller dataset)
4. Fine-tune all parameters (or freeze lower layers)

### Common Fine-tuning Tasks:

- Sentiment Analysis: Add binary classifier
- Named Entity Recognition: Token classification
- Question Answering: Span prediction
- Text Similarity: Sentence pair classification

### Typical Data Requirements:

- Minimum: 1,000 examples
- Good: 10,000 examples
- Excellent: 100,000+ examples

## Appendix A8: BERT vs Other Models

Model	Direction	Use Case	Params
BERT	Bidirectional	Understanding	110M
GPT-2	Left-to-right	Generation	1.5B
GPT-3	Left-to-right	Generation	175B
RoBERTa	Bidirectional	Better BERT	355M
ALBERT	Bidirectional	Efficient BERT	12M
XLNet	Permutation	Best of both	340M

### Key Differences:

- GPT: Autoregressive (good for generation)
- BERT: Autoencoding (good for understanding)
- RoBERTa: BERT with more data, no NSP
- ALBERT: Parameter sharing for efficiency

## Appendix A9: Emotion Classification Systems

### Plutchik's Wheel of Emotions:

- 8 primary emotions
- 3 intensity levels each
- Opposite pairs (joy-sadness, trust-disgust)
- Complex emotions as combinations

### Ekman's Basic Emotions:

- Anger, Disgust, Fear
- Happiness, Sadness, Surprise
- Universal across cultures

### For Product Reviews:

- Satisfaction/Dissatisfaction
- Delight/Frustration
- Trust/Skepticism
- Excitement/Disappointment

## Appendix A10: Complete BERT Implementation

### Full Working Example:

```
from transformers import pipeline, AutoTokenizer
import pandas as pd

# Load BERT for sentiment analysis
analyzer = pipeline("sentiment-analysis",
model="nlptown/bert-base-multilingual-uncased-sentiment")

# Analyze product reviews
reviews = pd.read_csv("reviews.csv")
results = []

for review in reviews['text']:
    result = analyzer(review)
    results.append({
        'text': review,
        'sentiment': result[0]['label'],
        'confidence': result[0]['score']
    })

# Find sarcasm patterns
sarcasm_indicators = ['great', 'love', 'fantastic', 'perfect']
low_ratings = reviews[reviews['rating'] <= 2]
potential_sarcasm = low_ratings[
    low_ratings['text'].str.contains('|'.join(sarcasm_indicators))
]
```

Full notebook available in course repository with fine-tuning examples.