Week 2: Understanding Emotions in Text

BERT + Empathize = What Users Really Mean

ML/AI/GenAI for Design Thinking

BSc Course - 12 Week Program

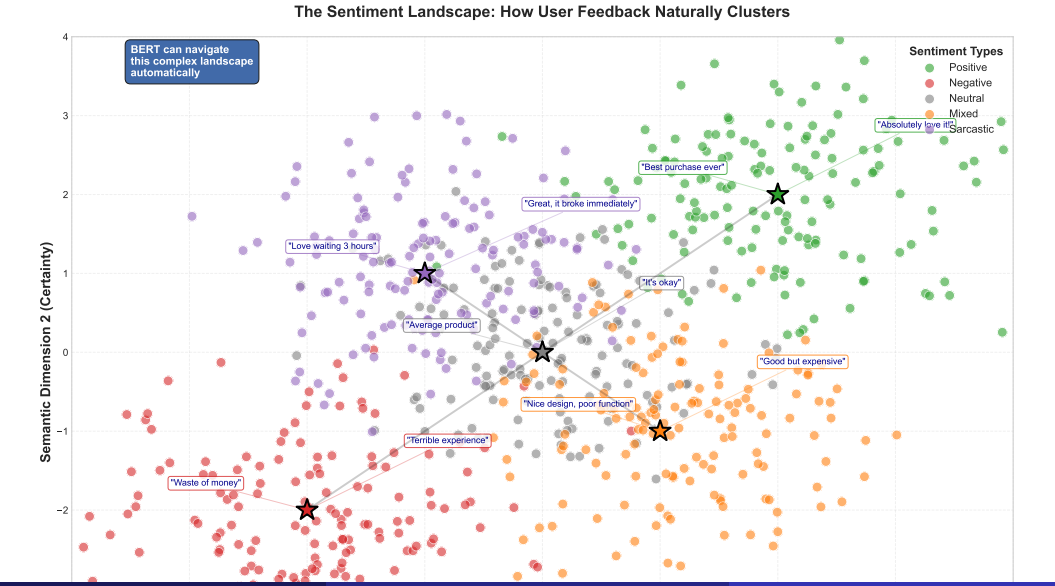
2024

Part 1: The Hidden Problem

What we miss in user feedback

Part 1 of 4

The Sentiment Landscape



The Problem: Hidden Emotions in Text

What users write:

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

What they actually mean:

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

Design Thinking blind spot:

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

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

Bidirectional: Seeing Past AND Future

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:

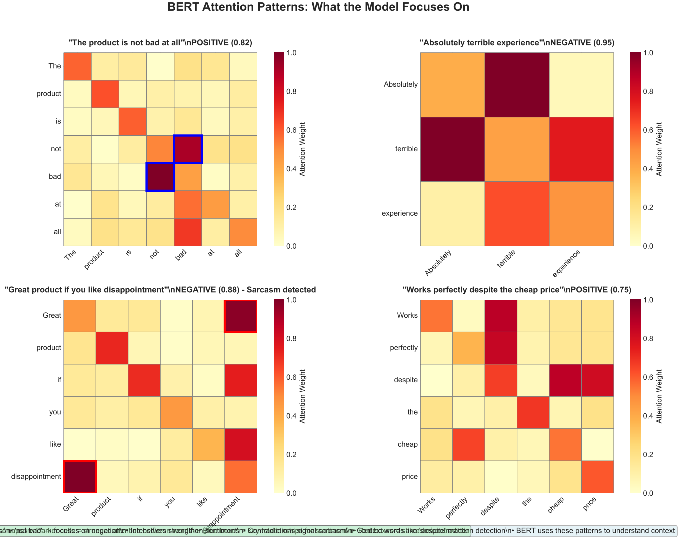
- **Intensity matters:** "Very frustrating" vs "Slightly frustrating"
- **Context reveals priority:** What made it frustrating?
- **Emotional nuance:** Frustrated vs Angry vs Disappointed

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



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

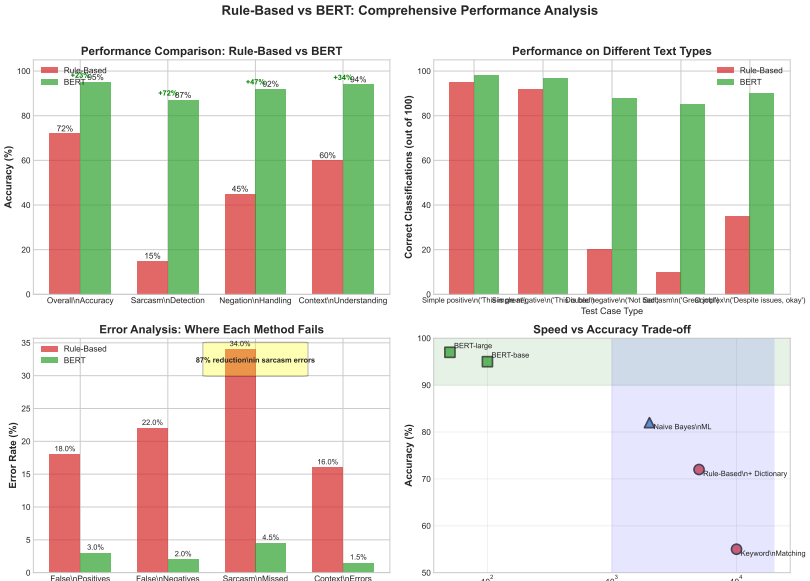
What this means:

- 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



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 real-world 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:

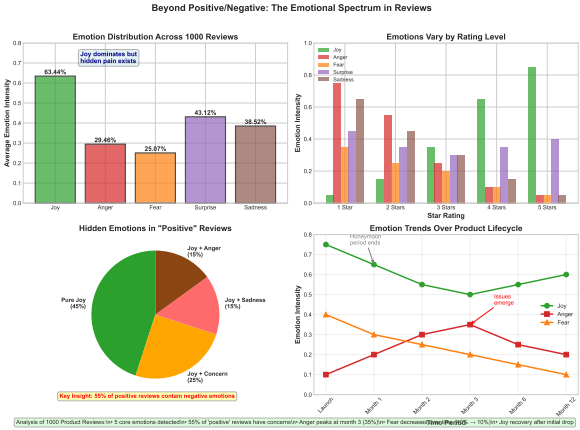
- Process 10,000+ reviews/day
- Understand global users
- Find silent majority opinions
- Detect emotional patterns

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

The User's Emotional Journey

Mapping Emotions Through the Product Experience

Discovery Phase:

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

Usage Phase:

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

Retention Phase:

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

Design Insight:
Track emotional transitions to identify critical intervention points

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

From Emotions to Design Thinking Insights

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

Real World: Netflix Emotion-Driven Design

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

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

Human + AI: Collaborative Design Intelligence

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

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

Part 4: Theoretical Foundations

The mathematics behind the magic

Part 4 of 4

Mathematical Foundation of Attention

The Attention Formula:

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

Components:

- **Q (Query):** What am I looking for?
- **K (Key):** What information exists?
- **V (Value):** What should I retrieve?
- $\sqrt{d_k}$**:** Scaling factor for stability

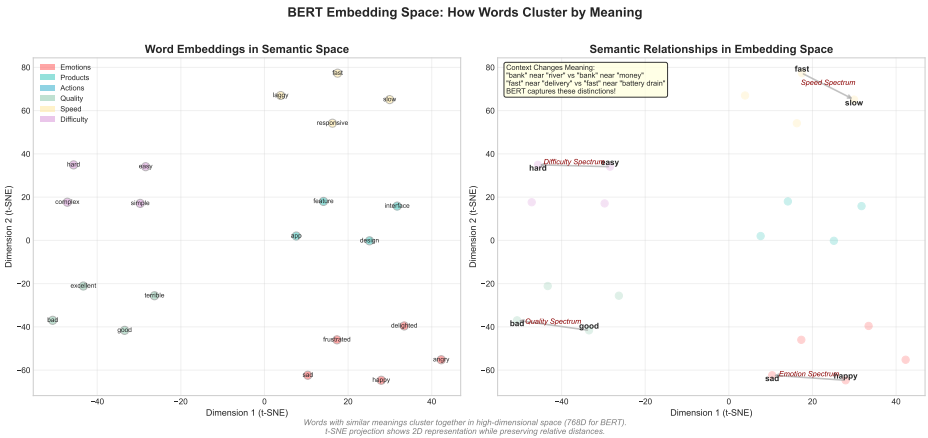
Why This Matters for Design:

- Attention scores = importance weights
- Higher scores = stronger relationships
- Shows which words influence meaning
- Reveals hidden connections

Example:
"Not bad" → High attention between "not" and "bad"
Result: BERT knows this means "good"

Design Impact: Attention reveals what users focus on in their feedback

Words as Vectors: The Embedding Space



Key Concepts:

- Words = 768-dimensional vectors
- Similar meanings cluster together
- Context changes position

Design Applications:

- Find similar feedback automatically
- Group related complaints
- Discover unexpected connections

How BERT Learns: Self-Supervised Training

Masked Language Modeling (MLM):

- Hide 15% of words randomly
- BERT predicts missing words
- Example: "The [MASK] was delicious"
- BERT learns: [MASK] = "food"

Why This Works:

- Forces understanding of context
- No labeled data needed
- Learns from massive text
- Captures language patterns

Next Sentence Prediction (NSP):

- Given: Sentence A and B
- Task: Are they consecutive?
- Learns discourse relationships
- Understands conversation flow

Design Thinking Connection:

- MLM = Understanding context
- NSP = Following user journeys
- Combined = Full comprehension

Result: BERT learns language like children do - by filling gaps and making connections

Transfer Learning: From General to Specific

Two-Stage Process:

Stage 1: Pre-training

- 3.3 billion words
- General language understanding
- No task-specific labels
- Learns universal patterns

Stage 2: Fine-tuning

- Your specific data
- Task-focused learning
- Much smaller dataset
- Adapts to your domain

Why Transfer Learning Works:

- Language basics are universal
- Specific tasks build on basics
- Reduces data requirements
- Faster training time

Design Thinking Analogy:

- Pre-training = Learning to read
- Fine-tuning = Understanding your users
- Result = Expert in your domain

Efficiency: 1,000 labeled examples can achieve 90%+ accuracy with transfer learning

Why Bidirectional Matters: The Theory

Traditional (Left-to-Right):

- Sequential processing
- P(word$_n$ — word$_1$...word$_{n-1}$)
- Can't see future context
- Limited understanding

BERT (Bidirectional):

- Parallel processing
- P(word$_n$ — all other words)
- Sees full context
- Complete understanding

Mathematical Advantage:

- More information per prediction
- Richer representations
- Better disambiguation
- Stronger contextual embeddings

Real Impact:

Task	Uni	Bi
Sentiment	82%	95%
Sarcasm	45%	87%
Context	71%	94%

Key Insight: Bidirectional = 2x the context = dramatically better understanding

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
- Grasp theoretical foundations

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)

Week 2 Summary: Theory Meets Practice

Technical Learning:

- BERT bidirectional
- Attention mechanism
- Embedding spaces
- Transfer learning
- 95% accuracy

Theory Mastered:

- Attention formula
- Vector semantics
- Self-supervision
- Context importance
- Mathematical foundation

Design Applications:

- Scale empathy
- Find pain points
- Real examples
- Proven results
- Clear ROI

The Journey: From Chaos to Clarity

Scattered (Raw data) → Clustered (Week 1) → Understood (Week 2)

BERT + Theory + Practice = Design with deep understanding

Appendix: Technical Deep Dive

For those who want to go deeper

Optional Material

Appendix A1: NLP Evolution Timeline

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: 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 A3: 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.