Week 2: Understanding Emotions in Text

BERT + Empathize = What Users Really Mean

ML/AI/GenAI for Design Thinking

BSc Course - 12 Week Program

2024

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

Words alone don't tell the whole story

Why Keyword Matching Fails

The "Not Bad" Problem:

Text	Keyword Method	Reality
"Not bad"	Negative	Positive
"Terribly good"	Mixed	Very Positive
"I love waiting 3 hours"	Positive	Negative (sarcasm)
"Could be worse"	Negative	Neutral/Positive

Why it fails:

- Counts words, ignores relationships
- Misses context completely
- Can't detect sarcasm or tone
- Treats all "not" as negative

The Challenge: Understanding Context

How can we teach computers to understand not just words, but what people really mean?

What we need:

1. See relationships between words
2. Understand that order matters
3. Detect sarcasm and tone
4. Work with thousands of reviews

Solution: BERT - A new way of reading text

What is BERT?

BERT = Bidirectional Encoder Representations from Transformers

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

Traditional: The $\rightarrow$ movie $\rightarrow$ was $\rightarrow$ not $\rightarrow$ bad $\rightarrow$ at $\rightarrow$ all
(Reads left to right, like humans)

BERT: [The movie was not bad at all]
(Sees everything simultaneously)

Why this matters:

- "Not" can look ahead to "bad"
- "At all" can modify "not bad"
- Context flows in both directions

Bidirectional: Seeing Past AND Future

Example: "The movie was ____ boring"

Old Way (Left to Right):

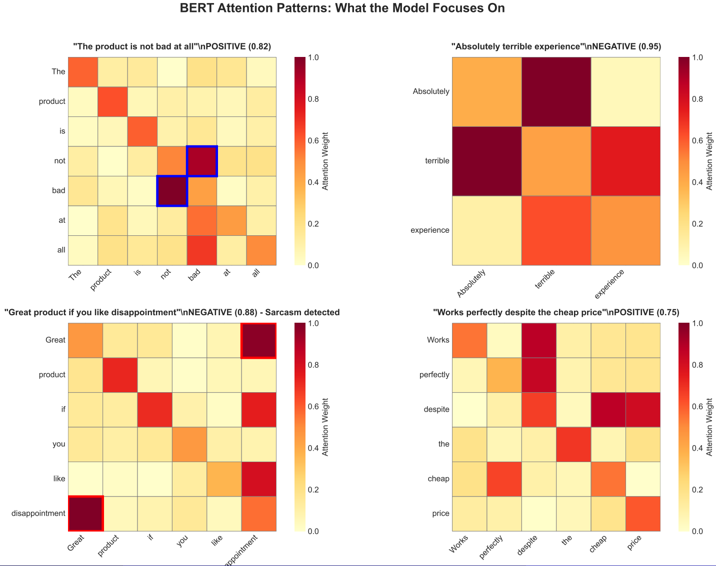
- Sees: "The movie was"
- Guesses: good? bad? long?
- Can't use "boring" as hint
- Often wrong

BERT (Both Directions):

- Sees: "The movie was" + "boring"
- Knows: probably "very" or "so"
- Uses full context
- Much more accurate

BERT sees the whole sentence like humans do

Attention: BERT Focuses on Important Words



Same Word, Different Meanings

Context completely changes word meaning:

Word	Context 1	Context 2
"Sick"	"I feel sick" (negative)	"That's sick!" (positive slang)
"Fire"	"Fire hazard" (danger)	"This song is fire" (excellent)
"Bank"	"River bank" (geography)	"Bank account" (finance)
"Apple"	"Apple pie" (food)	"Apple iPhone" (tech)

How BERT handles this:

- Creates different representations for each use
- Uses surrounding words to determine meaning
- No fixed dictionary - meaning emerges from context

How BERT Learns: Two-Step Process

Step 1: Pre-training (Learning Language)
Read millions of books and articles
Learn grammar, facts, and patterns
Like going to "language school"

↓

Step 2: Fine-tuning (Learning Your Task)
Read labeled reviews (positive/negative)
Learn what makes reviews positive or negative
Like specialized job training

Result: BERT understands language AND your specific problem

How BERT Detects Emotions

BERT's Process for "This product is not bad at all":

1. **Read everything:** See all 7 words simultaneously
2. **Connect words:** Link "not" with "bad", "at all" with phrase
3. **Build understanding:** Recognize double negative pattern
4. **Output emotion:** Positive (0.82 confidence)

> **Key insight:** BERT doesn't count words,
> it understands relationships

BERT Catches Sarcasm

Example: “Great product if you like disappointment”

How BERT detects sarcasm:

1. Sees contradiction: “Great” vs “disappointment”
2. High attention between conflicting words
3. Pattern matches with training examples
4. Outputs: Negative (0.88 confidence) + Sarcasm flag

Other sarcasm patterns BERT learns:

- “Perfect! It broke on day one”
- “Wonderful 3-hour wait”
- “Exactly what I wanted... not”

BERT vs Traditional Methods

Empathize: Understanding 10,000 Users at Once

The Design Challenge:

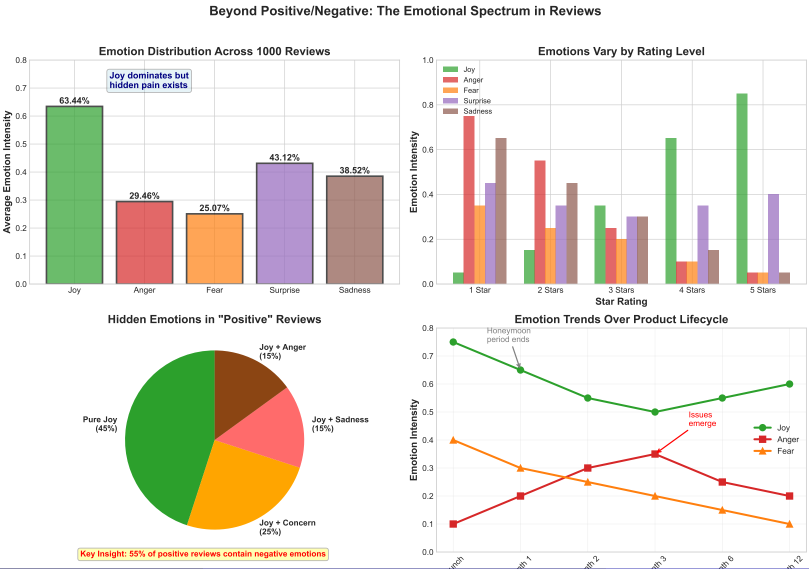
- Manual reading: 100 reviews/day maximum
- Digital products: 10,000+ reviews/day
- Each review: Unique human experience

BERT enables mass empathy:

- Process thousands of reviews in minutes
- Understand subtle emotions and frustrations
- Detect patterns humans might miss
- Maintain consistency across all data

BERT helps designers feel what thousands of users feel

Beyond Positive/Negative: Emotional Spectrum



Using Sentiment for Design Decisions

BERT insights → Design actions:

1. **Identify pain points:**
   "Love the app but login is frustrating" → Redesign login
2. **Understand priorities:**
   80% mention speed, 20% mention features → Focus on performance
3. **Detect confusion:**
   Sarcasm about "intuitive" interface → Simplify design
4. **Find delight moments:**
   Joy about specific feature → Enhance and highlight

Result: Data-driven empathy guides design choices

Combining BERT with Human Intuition

BERT strengths:

- Process volume
- Find patterns
- Consistent analysis
- Never tired

Human strengths:

- Understand context
- Creative solutions
- Ethical judgment
- Cultural nuance

Best results: BERT finds patterns, humans interpret meaning

Real World: Netflix Using BERT

Netflix Subtitle Emotion Analysis:

Problem: How to recommend shows based on mood?
Solution: BERT analyzes subtitle emotions
Result: Mood-based recommendations

Process:

1. BERT reads 50M+ subtitle files
2. Identifies emotional patterns in shows
3. Maps user viewing to emotional preferences
4. Recommends shows matching desired mood

Outcome: 15% increase in viewing completion

Context Matters More Than Keywords

- **Old way:** Count positive and negative words
- **BERT way:** Understand relationships and context
- **Result:** Real understanding of human emotion

BERT reads like a human, at machine scale

Next Week: From Understanding to Focusing

This week: BERT understands everything in text

The problem: Too much information!

- Long reviews with key points buried
- Important feedback hidden in noise
- Can't process everything equally

Next week: Attention Mechanisms

- How to focus on what matters most
- Finding needles in haystacks
- Extracting key insights automatically

From understanding all to focusing on what matters

BERT + Empathize = Understanding Emotions

What we learned:

1. Keywords fail because they ignore context
2. BERT reads bidirectionally (all words at once)
3. Context completely changes word meaning
4. BERT catches sarcasm through contradiction patterns
5. 95% accuracy vs 72% for traditional methods

For Design Thinking:

- Scale empathy to thousands of users
- Understand complex emotions beyond positive/negative
- Combine BERT insights with human creativity

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: Word Embeddings - Vector Spaces

Words as High-Dimensional Vectors:

- Each word $\rightarrow$ 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

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 = 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

Appendix A6: BERT Pre-training Tasks

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
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: Simple BERT Implementation

Python Code Example:

```
from transformers import pipeline

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

# Analyze text
text = "This product is not bad at all"
result = analyzer(text)

# Output: [{'label': 'POSITIVE', 'score': 0.82}]

# Fine-tuning example
from transformers import BertForSequenceClassification
model = BertForSequenceClassification.from_pretrained(
"bert-base-uncased", num_labels=2)
```

Full implementation available in course repository.